

The Official CompTIA® Linux+® Powered by LPI Study Guide (Exams LX0-103 and LX0-104)

Course Edition: 1.0

Acknowledgements

PROJECT TEAM



Jaron Rubenstein, Author

Alex Tong, Media Designer

Michelle Farney , Content Editor

Peter Bauer, Content Editor

Thomas Reilly, Vice President Learning

Katie Hoenicke, Director of Product Management

James Chesterfield, Manager, Learning Content and Design

Becky Mann, Senior Manager, Product Development

James Pengelly, Courseware Manager

Rob Winchester, Senior Manager, Technical Operations

Notices

DISCLAIMER

While CompTIA, Inc. takes care to ensure the accuracy and quality of these materials, we cannot guarantee their accuracy, and all materials are provided without any warranty whatsoever, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. The use of screenshots, photographs of another entity's products, or another entity's product name or service in this book is for editorial purposes only. No such use should be construed to imply sponsorship or endorsement of the book by nor any affiliation of such entity with CompTIA. This courseware may contain links to sites on the Internet that are owned and operated by third parties (the "External Sites"). CompTIA is not responsible for the availability of, or the content located on or through, any External Site. Please contact CompTIA if you have any concerns regarding such links or External Sites.

TRADEMARK NOTICES

CompTIA®, Linux+® Powered by LPI, and the CompTIA logo are registered trademarks of CompTIA, Inc., in the U.S. and other countries. All other product and service names used may be common law or registered trademarks of their respective proprietors.

Red Hat® Enterprise Linux® is a registered trademark of Red Hat Inc., in the U.S. and other countries; the Red Hat products and services discussed or described may be trademarks of Red Hat Inc. All other product and service names used may be common law or registered trademarks of their respective proprietors.

COPYRIGHT NOTICE

Copyright © 2018 CompTIA, Inc. All rights reserved. Screenshots used for illustrative purposes are the property of the software proprietor.

Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission CompTIA, 3500 Lacey Road, Suite 100, Downers Grove, IL 60515-5439.

This book conveys no rights in the software or other products about which it was written; all use or licensing of such software or other products is the responsibility of the user according to terms and conditions of the owner. If you believe that this book, related materials, or any other CompTIA materials are being reproduced or transmitted without permission, please call 1-866-835-8020 or www.help.comptia.org.

The Official CompTIA® Linux+® Powered by LPI Study Guide (Exams LX0-103 and LX0-104)

Title	The Official CompTIA® Linux+® Powered by LPI Study Guide (Exams LX0-103 and LX0-104)
	About This Guide
	Guide Description
	How to Use This Book
	1 Performing Basic Linux Tasks
	TOPIC A Identify the History and Development of Linux
	TOPIC B Enter Shell Commands
	TOPIC C Get Help Using Linux
	TOPIC D Start and Stop Linux
	Summary
	2 Managing User and Group Accounts
	TOPIC A Create User and Group Accounts
	TOPIC B Configure User Profiles
	TOPIC C Administer User and Group Accounts
	Summary
	3 Managing Partitions and the Linux Filesystem
	TOPIC A Create Partitions
	TOPIC B Navigate Through the Linux Filesystem
	TOPIC C Manage the Filesystem
	TOPIC D Maintain the Filesystem
	Summary
	4 Managing Files in Linux

TOPIC A Create and Edit Text Files

TOPIC B Locate Files

TOPIC C Search Text Using Regular Expressions

TOPIC D Apply Filters to Text Streams

TOPIC E Link Files

TOPIC F Back Up and Restore Files

TOPIC G Manage Databases Using MariaDB

Summary

5 Managing Linux Permissions and Ownership

TOPIC A Modify File and Directory Permissions

TOPIC B Modify Default Permissions

TOPIC C Modify File and Directory Ownership

TOPIC D Set Special Permissions and Attributes

Summary

6 Printing Files

TOPIC A Configure a Local Printer

TOPIC B Print Files

TOPIC C Configure Remote Printing

Summary

7 Managing Packages

TOPIC A Manage Packages Using RPM

TOPIC B Verify Packages

TOPIC C Upgrade Packages

TOPIC D Configure Repositories

TOPIC E Manage Packages Using YUM

TOPIC F Advanced Package and Application Management

Summary

8 Managing Kernel Services

TOPIC A Explore the Linux Kernel

TOPIC B Customize Kernel Modules

TOPIC C Create an initrd Image

TOPIC D Manage Device Drivers and Hardware Devices

TOPIC E Monitor Processes and Resources

Summary

9 Working with the Bash Shell and Shell Scripts

TOPIC A Perform Basic Bash Shell Operations

TOPIC B Write a Bash Shell Script

TOPIC C Customize the Bash Shell

TOPIC D Redirect Standard Input and Output

TOPIC E Use Control Statements in Shell Scripts

Summary

10 Managing Jobs and Processes

TOPIC A Manage Jobs and Background Processes

TOPIC B Manage Processes Using the Process Table

TOPIC C Delay and Detach Jobs

TOPIC D Schedule Jobs

TOPIC E Maintain the System Time

Summary

11 Managing System Services

TOPIC A Configure System Services

TOPIC B Monitor System Logs

TOPIC C Configure Security-Enhanced Linux (SELinux)

Summary

12 Configuring Network Services

TOPIC A Connect to a Network

TOPIC B Configure Routes

TOPIC C Configure Client Network Services

TOPIC D Manage Remote Network Systems

Summary

13 Configuring Basic Internet Services

TOPIC A Configure Email Services

TOPIC B Control Internet Services

Summary

14 Securing Linux

TOPIC A Implement Basic System Security

TOPIC B Secure User Accounts

Summary

15 Managing Hardware

TOPIC A Identify Common Hardware Components and Resources

TOPIC B Configure Removable Hardware

TOPIC C Configure Disk Quotas

Summary

16 Troubleshooting Linux Systems

TOPIC A Troubleshoot System-Based Issues

TOPIC B Troubleshoot Hardware Issues

TOPIC C Troubleshoot Network Connection and Security Issues

Summary

17 Installing Linux

TOPIC A Prepare for Installation

TOPIC B The Linux Boot Process

TOPIC C Configure GRUB

TOPIC D Install the Operating System

Summary

18 Configuring the GUI

TOPIC A Implement X

TOPIC B Customize the Display Manager

TOPIC C Enable Accessibility Settings in Linux

Summary

A Taking the Exams

B Mapping Course Content to the CompTIA® Linux +® Powered by LPI (Exams LX0-103 and LX0-104) Exam Objectives

C Syntax

D Additional Security Topics

TOPIC A Enable Firewall Functionality

TOPIC B Security Auditing

TOPIC C Identify an Intrusion Detection System

Solutions

ACTIVITY 1-1: Performing Basic Linux Tasks Review

ACTIVITY 2-1: Managing User and Group Accounts Review

ACTIVITY 3-1: Managing Partitions and the Linux Filesystem Review

ACTIVITY 4-1: Managing Files in Linux Review

ACTIVITY 5-1: Managing Linux Permissions and Ownership Review

ACTIVITY 6-1: Printing Files Review

ACTIVITY 7-1: Managing Packages Review

ACTIVITY 8-1: Managing Kernel Services Review

ACTIVITY 9-1: Working with the Bash Shell and Shell Scripts Review

ACTIVITY 10-1: Managing Jobs and Processes Review

ACTIVITY 11-1: Managing System Services Review

ACTIVITY 12-1: Configuring Network Services Review

ACTIVITY 13-1: Configuring Basic Internet Services Review

ACTIVITY 14-1: Securing Linux Review

ACTIVITY 15-1: Managing Hardware Review

ACTIVITY 16-1: Troubleshooting Linux Systems Review

ACTIVITY 17-1: Installing Linux Review

ACTIVITY 18-1: Configuring the GUI Review

Glossary

Index

About This Guide

The Official CompTIA® Linux+® Powered by LPI Study Guide (Exams LX0-103 and LX0-104)

builds on your existing user-level knowledge and experience with the Linux operating system to present fundamental skills and concepts that you will use on the job in any type of Linux career.

This guide can benefit you in two ways. If your job duties include Linux troubleshooting, installation, or maintenance, or if you are preparing for any type of Linux-related career, it provides the background knowledge and skills you will require to be successful. In addition, if you intend to pass the CompTIA® Linux+® Powered by LPI certification examinations (LX0-103 and LX0-104) , this guide can be a significant part of your preparation.

Guide Description

Target Student

This guide is intended for entry-level computer support professionals with basic knowledge of computer hardware, software, and operating systems, who wish to increase their knowledge and understanding of Linux concepts and skills to prepare for a career in Linux support or administration, or to prepare for the CompTIA® Linux+® certification examination. A typical student in this guide should have at least 6 to 12 months of Linux experience.

Guide Prerequisites

To ensure your success, *CompTIA® A+®: A Comprehensive Approach (Exams 220-801 and 220-802)* is helpful but not required.

Guide Objectives

In this guide, you will administer Linux.

You will:

- Identify basic Linux concepts and perform basic Linux tasks.
- Manage user and group accounts.
- Manage partitions and the Linux filesystem.
- Manage various files in Linux.
- Work with Linux permissions and ownership.
- Print files.
- Manage packages.
- Manage kernel services.
- Work with the Bash shell and shell scripts.
- Manage jobs and processes.
- Manage system services.
- Configure network services.
- Configure basic Internet services.

- Implement measures to secure a Linux system.
- Manage hardware associated with Linux systems.
- Troubleshoot Linux system issues.
- Install the Linux operating system.
- Configure the GUI.

How to Use This Book

As You Learn

This book is divided into lessons and topics, covering a subject or a set of related subjects. In most cases, lessons are arranged in order of increasing proficiency.

The results-oriented topics include relevant and supporting information you need to master the content. Each topic has various types of information designed to enable you to solidify your understanding of the informational material presented in the guide. Information is also provided for reference and reflection to facilitate understanding and practice.

At the back of the book, you will find a glossary of the definitions of the terms and concepts used throughout the guide. You will also find an index to assist in locating information within the instructional components of the book.



As a Reference

The organization and layout of this book make it an easy-to-use resource for future reference.

Taking advantage of the glossary, index, and table of contents, you can use this book as a first source of definitions, background information, and summaries.

Guide Icons

Watch throughout the material for the following visual cues.

<i>Icon</i>	<i>Description</i>
	A Note provides additional information, guidance, or hints about a topic or task.
	A Caution note makes you aware of places where you need to be particularly careful with your actions, settings, or decisions so that you can be sure to get the desired results of an activity or task.

1 Performing Basic Linux Tasks

Lesson Time: 2 hours, 30 minutes

Lesson Introduction

You may have experience using the Linux® environment, or you may be ready to learn Linux for the first time. In either case, it is good to have an understanding of how Linux was developed and where it is today. In this lesson, you will identify important elements in the history and development of Linux and perform basic Linux tasks.

When Linux was first available, it was not readily accepted by businesses. But today, it is rapidly gaining acceptance in the corporate world, especially for website hosting. Linux is also increasingly used on desktops as a viable alternative to various versions of Microsoft Windows for businesses and individuals alike. By learning the origin of Linux and familiarizing yourself with its basic functions, you will become a more confident Linux user.

Lesson Objectives

In this lesson, you will identify basic Linux concepts and perform basic Linux tasks. You will:

- Identify the key events in the history and development of Linux.
- Enter basic shell commands.
- Access help in Linux.
- Start and stop Linux.

TOPIC A Identify the History and Development of Linux

Operating systems vary greatly from manufacturer to manufacturer. Even if you know very little about the Linux® environment, a basic understanding of its roots will be beneficial to you. In this topic, you will identify key events in the history and development of Linux.

Over the past few years, Linux has rapidly gained ground in the competitive operating system marketplace. For example, Linux is now widely preferred for web servers and Internet systems.

Many individuals and organizations have accepted it as a desktop and server alternative because of its high security, low cost, and ease of licensing. By learning about the basics of Linux and its development cycle, you will understand and appreciate its benefits.

Open Source Software

Open source software enables users to access its source code and gives them the right to modify it.

Open source licensing ensures that free and legal redistribution of the software is possible. Although the software can be modified and improved by individual users, the integrity of the author's code is preserved by ensuring that modifications to the original source code are redistributed only as patches.



Figure 1-1: Linux is an open source operating system.

Need for Open Source

In the early days of computing, many programmers freely shared new software they developed with other users, along with the source code. This community approach enabled knowledgeable users to modify and improve the software. However, with the introduction of restrictive licensing practices by big companies, some operating systems and utility programs could not be legally copied by users, and users no longer had access to the source code, making it impossible for users to create their own customized versions of the software. Some programmers, therefore, disliked the concepts of closed source and proprietary software. Richard Stallman, then working at MIT's Artificial Intelligence labs, was one such programmer who wanted to create an alternative, open source software. Some examples of open source software used today are Linux, Perl, PHP, Python, and OpenOffice.

Free Software vs. Open Source Software

Although most of the free software is also open source, the terms are not interchangeable. Open source is a development methodology in which anyone can access the source code, though it is possible to prevent any modification of the code by means of a special licensing agreement. Free software focuses on ethical issues of protecting a user's freedom, where there are no restrictions on how the user runs a program or how frequently the user is allowed to copy and share the program.

The GNU Project

GNU's Not UNIX (GNU) is a comprehensive computer operating system composed entirely of free software. The [GNU project](#) was started by Richard Stallman in 1984, as an initiative to produce a source for free and open software. Stallman wrote much of the GNU software himself, including the GNU C compiler or gcc and the emacs text editor. Later, several programmers worked together to develop more utilities that are compatible with GNU utilities.



Note: Richard Stallman chose the recursive acronym "GNU's Not UNIX" to show that though GNU was like the free version of [UNIX](#) in its design, it did not contain any code from UNIX. Note that the "G" in GNU is included in the pronunciation of the term "guh-NOO."

The Free Software Foundation (FSF) is a nonprofit organization founded by Richard Stallman in 1984 to promote the development of free software. It advocates the movement against the monopoly of copyrighted, proprietary software by ensuring the availability of all software to users without any restrictions on use, distribution, or modification.

Free software refers to users' rights rather than cost, and so free software may be sold at a price.

Free software must not be confused with freeware, which is software that is available free of cost.

Freeware may sometimes include even proprietary software offered on a demo basis. For an online source, see www.fsf.org.

Copyleft

Copyleft is the method of ensuring that all original works, and their derivative works, are kept free and open. The term copyleft is used to define a concept that is essentially the opposite of copyright.

Richard Stallman proposed this concept to create a licensing arrangement under which software can be freely used, modified, and copied by others. The Free Software Foundation (FSF) recommends that all free software be copylefted and released under General Public License (GPL).



Figure 1-2: The copyleft symbol.

GNU Utilities

GNU utilities released under GPL are copylefted because they cannot be copyrighted by anyone who modifies them.

GPL

General Public License (GPL) is a licensing agreement that effectively enforces public ownership of software released under it. GPL states that a programmer holds the copyright to a specific piece of software. This prevents the software from being placed in the public domain, where anyone can modify it and then copyright the modified version. The software is then subjected to a licensing agreement that allows it to be freely used, modified, and copied. Anyone who modifies the code and distributes it to others must provide the open source code that includes their modifications, making it freely available under the terms of GPL. Copyleft is a principle or standard of which GPL is an implementation. Three versions of GPL are available.

Version	Date of Release
GPLv1	January 1989
GPLv2	June 1991
GPLv3	June 2007

The Linux Operating System

The **Linux** operating system is a complete, open source operating system that combines GNU utilities and the Linux **kernel**. The kernel is the central core of the Linux operating system that manages all the computer's physical

devices. The Linux kernel was developed by Linus Torvalds in 1991, while he was a student at the University of Helsinki. A year later, Torvalds released Linux kernel 1.0 under GPL. The Linux commands closely resemble those found in other UNIX-type operating systems. Many programs written for other operating systems run on Linux.

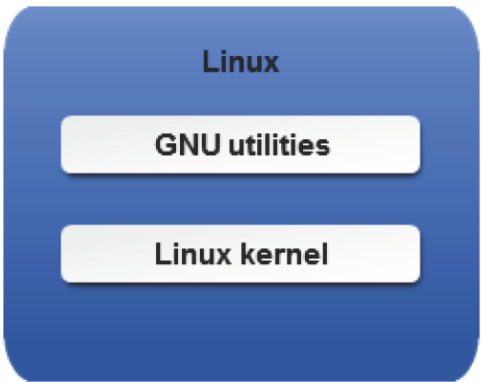


Figure 1-3: Linux is a combination of GNU utilities and the Linux kernel.

Origin of the Linux Kernel

Linus Torvalds, independently developed a UNIX-like operating system kernel in 1991 for his own use, inspired by another system called Minix. He posted his creation on the Internet and asked other programmers to help him further develop it. At that point, Linux could already run UNIX utilities such as **bash**, **gcc**, and **gnu-sed**. Until Torvalds agreed to release Linux under GPL, the GNU project was not a complete operating system, and the kernel itself was incomplete as an operating system without utilities.

Linux Timeline

The following table outlines important dates in the development of Linux.

<i>Year</i>	<i>Linux-Related Events</i>
1984	Richard Stallman launched the GNU project.
1989	GNU and GPL were released.
1991	Linus Torvalds developed a UNIX-like operating system called Linux (version .02).
1992	Linux kernel 1.0 was released under GPL, and SUSE and TurboLinux were founded.
1993	Red Hat was founded, the Debian project began, and Slackware was first released.
1994	Caldera Inc. was founded.
1995	Red Hat Linux 4.0 was released.
1996	The penguin (Tux) was suggested as the mascot for Linux and Linux kernel 2.0 was released.
1997	Red Hat Linux 5.2 and Debian 1.3 were released.
1998	MandrakeSoft was founded and Debian 2.0 was released.
1999	Red Hat Linux 6.0 and SUSE 6.3 were released.
2000	Red Hat Linux 7.0 and Caldera OpenLinux eDesktop 2.4 were released.
2001	Linux kernel 2.4, SUSE 7.2, Debian 2.23r, Slackware 8.0, Caldera OpenLinux Server, and Workstation 3.1 were released.
2002	GNOME 2.0 was released.
2003	Linux kernel 2.6 and Fedora were released.
2004	The GNU project celebrated its 20th anniversary, and GNOME 2.6 and the first official version of Ubuntu Linux were released. The first release (CentOS 3) of CentOS Linux was distributed as a community-supported distribution derived from sources freely provided to the public by Red Hat.
2005	Red Hat Enterprise Linux 4 was released. Mandrake Linux was renamed as Mandriva Linux.

<i>Year</i>	<i>Linux-Related Events</i>
2007	Red Hat Enterprise Linux 5 and Red Hat Enterprise Linux 5 Update 1 (5.1) were released.
2008	Linux kernel 2.6.28 and Red Hat Enterprise Linux 5 Update 2 (5.2) were released.
2009	Red Hat Enterprise Linux 5 Update 3 (5.3) and Linux kernel 2.6.29 were released. Red Hat Enterprise Linux 5 Update 4 (5.4) and Linux kernel 2.6.30 through 2.6.32 were released.
2010	Red Hat Enterprise Linux 5 Update 5 (5.5) and Linux kernel 2.6.33 were released. Red Hat Enterprise Linux 6 and CentOS Linux 6 were released.
2014	Red Hat Enterprise Linux 7, CentOS Linux 7, and Ubuntu 14.04 LTS were released.

Common Areas of Use for Linux

Linux is mainly used on servers, workstations, and desktops.

<i>Use</i>	<i>Description</i>
Server	Used as a web server to host websites and as a file server to provide file access for multiple clients. Also used to control and secure network traffic.
Workstation	Designed for a business environment geared toward programmers.
Desktop	Focused on home users who run office and graphics applications and games.

Other Uses of Linux

The Linux operating system is very versatile. It can be used as a:

- Domain name server.
- Gateway or routing server.
- Web server.
- Database server.
- Software development platform.
- High Performance Supercomputer Cluster.

Linux Distributions

Since its creation, Linux has evolved into hundreds of distributions, also called *distros*, each tailored to their designers' needs. If you are a beginner, you will find it easier to choose one of the mainstream distributions depending on the installations. Some common distributions are:

- CentOS
- Red Hat® Enterprise Linux (RHEL)
- Fedora
- SUSE Linux Enterprise
- openSUSE
- Debian
- Ubuntu

- Mandriva
- Mint

Internet Reference for Common Linux Distributions

You can refer to common Linux distributions in the following Internet sites:

- CentOS Linux: www.centos.org
- Red Hat Enterprise Linux (RHEL): www.redhat.com
- Fedora: <http://fedoraproject.org>
- SUSE Linux Enterprise: www.novell.com/linux
- openSUSE: www.opensuse.org
- Debian: www.debian.org
- Ubuntu: www.ubuntu.com
- Mandriva: www.mandriva.com
- Mint: www.linuxmint.com

The CentOS Linux Distribution

The CentOS Linux distribution is a stable, predictable, manageable, and reproducible platform derived from the sources of Red Hat Enterprise Linux (RHEL). CentOS is maintained by The CentOS Project, a community-driven free software effort that is modelled on the structure of the Apache Foundation and has its own governing board. CentOS benefits from Red Hat's ongoing contributions and investment.

This course uses CentOS Linux because it provides a free enterprise class computing platform that aims to be functionally compatible with the upstream product (RHEL) that it derives from. CentOS Linux does not contain Red Hat, Inc.'s product or certifications, although it is built from the same sources as the upstream enterprise products. More details about this may be found in the CentOS FAQ here: <http://wiki.centos.org/FAQ/General>

For production environments, the licensed and fully supported Red Hat Enterprise Linux product is recommended.

Software Acquisition

There are two ways of obtaining Linux software: purchasing it from a local computer outlet or downloading it from a website. While it is convenient to purchase Linux software from a store, downloading it over a broadband connection is also practical.

Comparing Distributions and Their Packaging Solutions

Linux distributions are similar to each other and each has its own strengths and weaknesses. The software packaged with each distribution can make a huge difference in how Linux works for you.

Although you can always download or purchase missing components, the software package is easier to use if the components have already been tested and compiled together to work with your distribution.

TOPIC B Enter Shell Commands

Now that you understand the origin of Linux, you should learn its basics so that you can use it. The Linux shell prompt is where you enter commands. In this topic, you will describe the shell and enter shell commands.

Learning to enter shell commands will allow you to interact directly with the Linux operating system. You will be able to utilize Linux commands to perform various tasks. In its formative stages, Linux was operated solely through the command line interface using shell commands. With the addition of the GUI, tasks have become easier, but a lot of power and flexibility still reside in knowing the shell commands.

The GUI

The Linux **Graphical User Interface (GUI)** is a collection of icons, windows, and other screen graphical elements that help users interact with the operating system. The desktop menu provides access to the GUI applications available on the Linux desktop. There are different GUI implementations such as K Desktop Environment (KDE) and GNU Object Model Environment (GNOME).

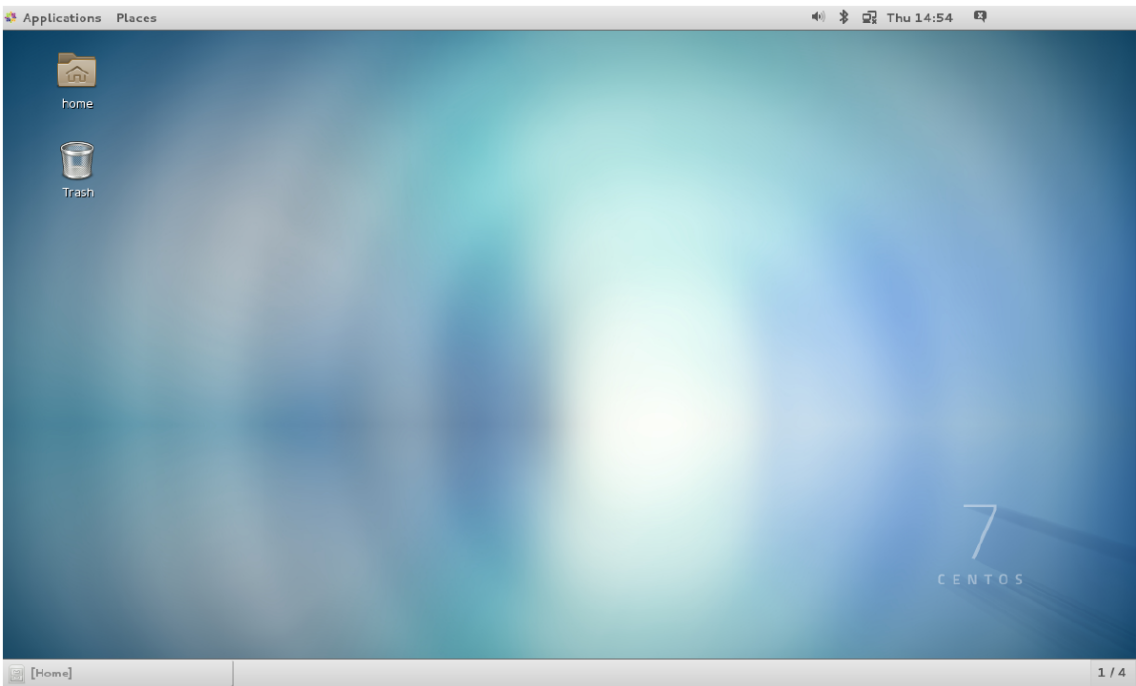


Figure 1-4: A GNOME desktop in CentOS 7.

The following table lists the uses of some common **Applications** menu categories in the GNOME GUI.

<i>Applications Menu Category</i>	<i>Used To</i>
Accessories	Access applications for performing work-related tasks such as creating text documents and presentations or using a calculator.
Internet	Access applications for performing tasks on the Internet such as web browsers, email clients, instant messengers, or web editors.
Sound & Video	Access applications for viewing movies and listening to sound files or CDs.
System Tools	Access options for changing the settings on the Linux system.
Documentation	Access help on Linux.

The CLI

The **Command Line Interface (CLI)** is a text-based interface for the operating system, where a user typically enters commands at the **command prompt** to instruct the computer to perform a specific task. A **command line interpreter**, or command line shell, is a program that implements the commands entered in the text interface. The command line interpreter analyzes the input text provided by the user, interprets the text in the concept given, and then provides the output.

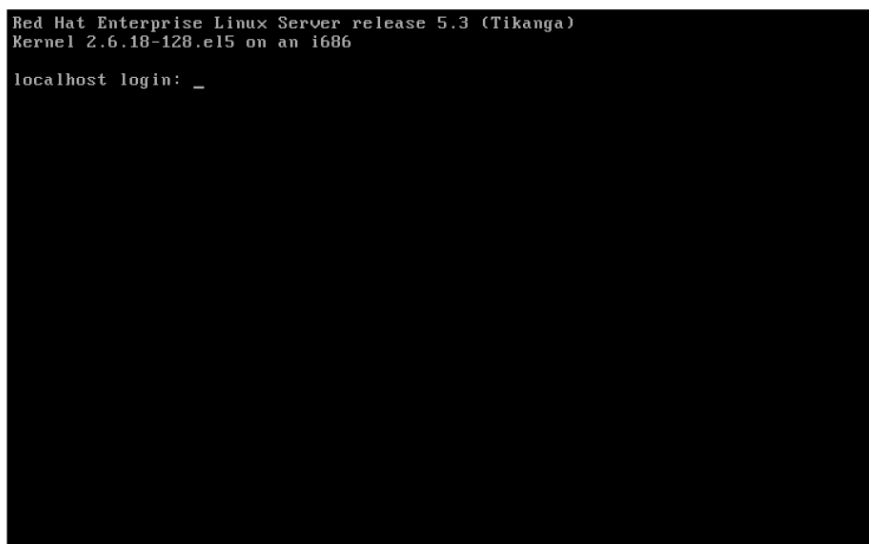


Figure 1-5: A CLI screen.

Shells

A **shell** is a component that interacts directly with users. It also functions as the command interpreter for the Linux system. The shell accepts user commands and ensures that the kernel carries them out. The shell also contains an interpretive programming language.

The various shells available in Linux are described in the following table.

Shell	Description
Bash	This is the default Linux shell. It provides the flexibility of the C shell in a Bourne shell-type environment. Use the command <code>bash</code> to open the Bash shell.
Bourne	This is the original UNIX shell developed by Steve Bourne at Bell Labs and is available on all Linux systems. Use the command <code>sh</code> to open the Bourne shell.
C shell	This was developed by Bill Joy at Berkeley and was designed to support C language development environments. It was also designed for more interactive use, providing several ways to reduce the amount of typing needed to complete a job. Use the command <code>csh</code> to open the C shell.
Korn	This shell is a combination of the C and Bourne shells. It uses the features of the C shell but the syntax of the Bourne shell. Use the command <code>ksh</code> to open the Korn shell.



Figure 1-6: A blank shell prompt.



Figure 1-7: The shell prompt in the GUI terminal window.

Opening Multiple Shells

You can have several shells open at the same time with different processes or programs running in each shell. For example, to open a second Bash shell, enter `bash` at the command prompt. To open a C shell, enter `csh`. To close a shell, either enter `exit` or press **Ctrl+D**.

Determining the Current Shell

The `echo` command enables you to determine the shell that is established at the login. To determine the current shell, enter `echo $SHELL`, where `$SHELL` is the environmental variable name that holds the name of the current shell.

View a File One Page at a Time

To view a file one page at a time, simply type the `more` command in front of the file you want to open. For example, if you want to read the `/etc/passwd` file, type `more /etc/passwd`. You can then view the next full screen of content by pressing the **Spacebar** or one additional line of content at a time by pressing **Enter**. To quit viewing the file, press the **q** key.

To view a file one page at a time, simply type the `less` command in front of the file you want to open. For example, if you want to read the `/etc/passwd` file, type `less /etc/passwd`. As with the `more` command, you can then view the next full screen of content by pressing the **Spacebar** or one additional line of content at a time by pressing **Enter**. To go backwards a page, press the **b** key. To quit viewing the file, press the **q** key.



Note: The `less` and `more` commands are very similar. At one time, the `less` command had additional features that the `more` command did not have, but they are now feature equivalent and can generally be used interchangeably. Feel free to use your preferred command throughout the course.

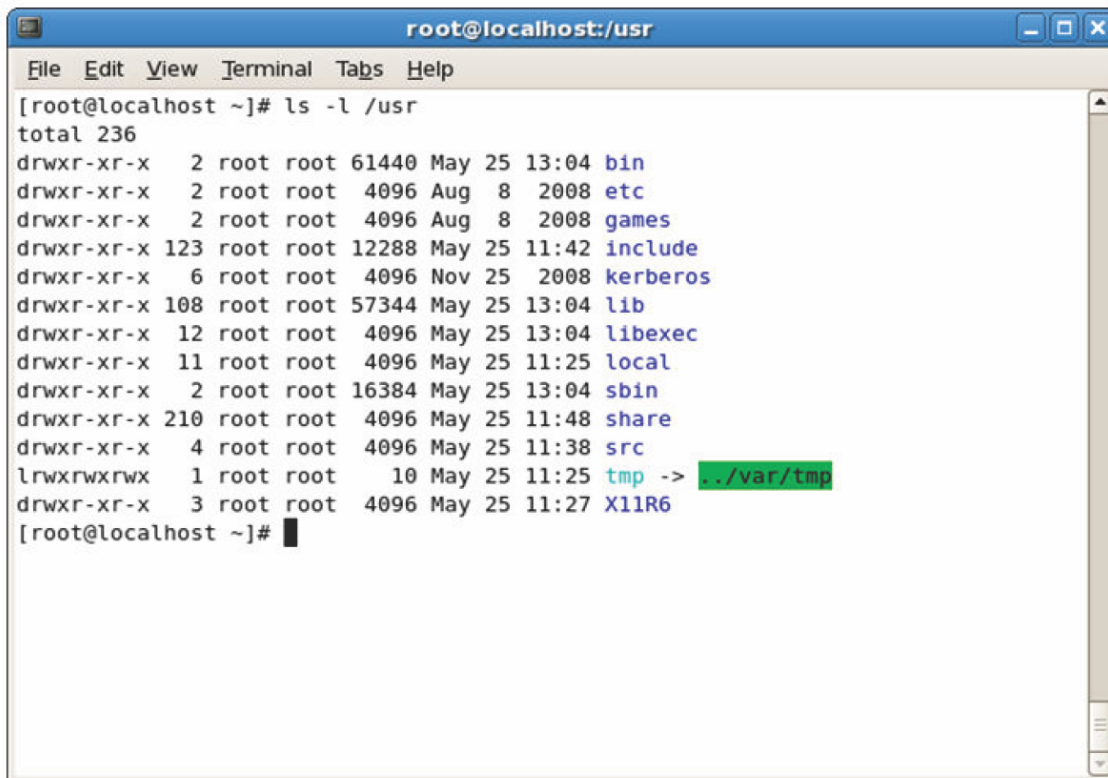
The head and tail Commands

The `head` command displays the first 10 lines of each file. The `tail` command displays the last 10 lines of each file. These commands are useful when you only need to see the beginning or the end of a file. For example, you can check recent log entries by viewing the last 10 lines of a log file.

Shell Commands

The generic format for a shell command is `command -option argument`. After typing your command, the shell responds by performing a specific action that is associated with that command.

Linux is case sensitive, so you must enter commands in the required case.



```
root@localhost: /usr
File Edit View Terminal Tabs Help
[root@localhost ~]# ls -l /usr
total 236
drwxr-xr-x  2 root root 61440 May 25 13:04 bin
drwxr-xr-x  2 root root 4096 Aug  8 2008 etc
drwxr-xr-x  2 root root 4096 Aug  8 2008 games
drwxr-xr-x 123 root root 12288 May 25 11:42 include
drwxr-xr-x  6 root root 4096 Nov 25 2008 kerberos
drwxr-xr-x 108 root root 57344 May 25 13:04 lib
drwxr-xr-x 12 root root 4096 May 25 13:04 libexec
drwxr-xr-x 11 root root 4096 May 25 11:25 local
drwxr-xr-x  2 root root 16384 May 25 13:04 sbin
drwxr-xr-x 210 root root 4096 May 25 11:48 share
drwxr-xr-x  4 root root 4096 May 25 11:38 src
lrwxrwxrwx  1 root root   10 May 25 11:25 tmp -> ../var/tmp
drwxr-xr-x  3 root root 4096 May 25 11:27 X11R6
[root@localhost ~]#
```

Figure 1-8: The `ls` command displays the list of files in the `usr` directory.

Argument

An **argument**, also called command line argument, is usually a file name or directory name that indicates the files on which the command will operate. It is used as an input by some commands in Linux. Arguments can be files, directories, commands, or even a command switch. For example,

`ls {file name}`, `ls {directory name}`, and `ls -l`.

Command History

Sometimes commands can become quite long. You can access previously entered commands that are stored in the History file by using the **Up Arrow** and the **Down Arrow** keys.

Invoking Commands Outside a Path

There are two ways of invoking a command located outside a path. You can specify the path in which the command is located and then invoke the command. For example, assume that a command is located in the **{user-defined directory}** directory. To invoke this command, you need to enter `{user-defined directory}/{command name}`.

You can also navigate to the directory that contains the command and then invoke it. For example, assume that a command is located in the **{user-defined directory}** directory. You need to change to that directory with the `cd {user-defined directory}` command and then enter `{command name}`.

The Tab-Completion Feature

Some commands have long names containing version number information, weird spellings, or capitalizations. This can make it difficult to correctly enter the commands on the first try. In such a case, you can make use of the tab-completion feature. To use this feature, enter the first few characters of the command and then press **Tab**. If there is only one match, the rest of the file name is displayed. If you press the next letter of the file name you want and press **Tab** again, the complete file name should come up. If the system still cannot differentiate between the commands, it will beep again, and you have to enter additional characters or press **Tab** two times to view all available options.

Piping Commands

You can send or redirect the results of one command to another command. Pipes are used to combine Linux tools on a single command line, enabling you to use the output of one command as the input to another. The pipe symbol is a vertical bar (|), which you type between two commands.

For example, `ls|more` enables you to look at a large directory listing one screen at a time.

Issuing More Than One Command

You can issue more than one command before pressing **Enter**. Place a semicolon (;) between the commands and they will be issued one after the other.

The exec Command

If you enter a command, it runs as a child process to Bash, which is the parent process. If you enter `exec {command}`, the `exec` command will kill the parent processes and the bash process, and `{command}` starts to run as the parent process. For example, when a user has a limit applied on the number of process, the user can use the `exec` command to run an additional process by killing the parent process. Once the `exec {command}` is executed, you will be automatically logged out because the bash process has been terminated.

The date Command

The `date` command displays the current date and time set on a system. You can use the hyphen (-) or the colon (:) between the different fields of the date for a clear output.

```
[root@srv02 ~]# date
Wed Apr  1 11:12:35 EDT 2015
[root@srv02 ~]# _
```

Figure 1-9: Viewing the current date using the `date` command.

Syntax

The syntax of the `date` command is `date +[format]`, where *format* is the string of characters that are used to display the different fields of the output.

Characters Used with the `date` Command

The characters that are used to display the different fields of the `date` command are listed in the following table.

Character	Description
%d	Displays the current day of the month (01 to 31).
%D	Displays the date in the format mm/dd/yy, where mm is month, dd is day, and yy is year.

Character	Description
%H	Displays the current hour in the 24-hour format (00 to 23).
%I	Displays the current hour (1 to 12). This option does not display A.M. or P.M. after the hour.
%m	Displays the current month of the year (01 to 12).
%M	Displays the current minute (00 to 59).
%r	Displays the time in the 12-hour format; i.e., hh:mm:ss [A.M. or P.M.].
%R	Displays the time in the 24-hour format; i.e., hh:mm. This option does not display the seconds.
%S	Displays the seconds (00 to 60).
%T	Displays the time in the 24-hour format; i.e., hh:mm:ss. This option displays the seconds also.
%y	Displays the last two digits of the current year.
%Y	Displays the current year in four digits (yyyy).

The cal Command

The cal command displays the calendar for any month or year. If you do not specify the month or year with the cal command, it will display the calendar of the current month. You can display the calendar of a specific month in a year by specifying the month and the year after the cal command.

The year must be specified in the yyyy format. The command cal 10 will display the calendar for the year 10 A.D. and not for the year 2010.

```
[root@srv02 ~]# cal
      April 2015
Su Mo Tu We Th Fr Sa
    1  2  3  4
  5  6  7  8  9 10 11
 12 13 14 15 16 17 18
 19 20 21 22 23 24 25
 26 27 28 29 30
[root@srv02 ~]# _
```

Figure 1-10: Viewing the calendar of the current month using the cal command.

Syntax

The syntax of the cal command is cal {month} {year}.

Options for the cal Command

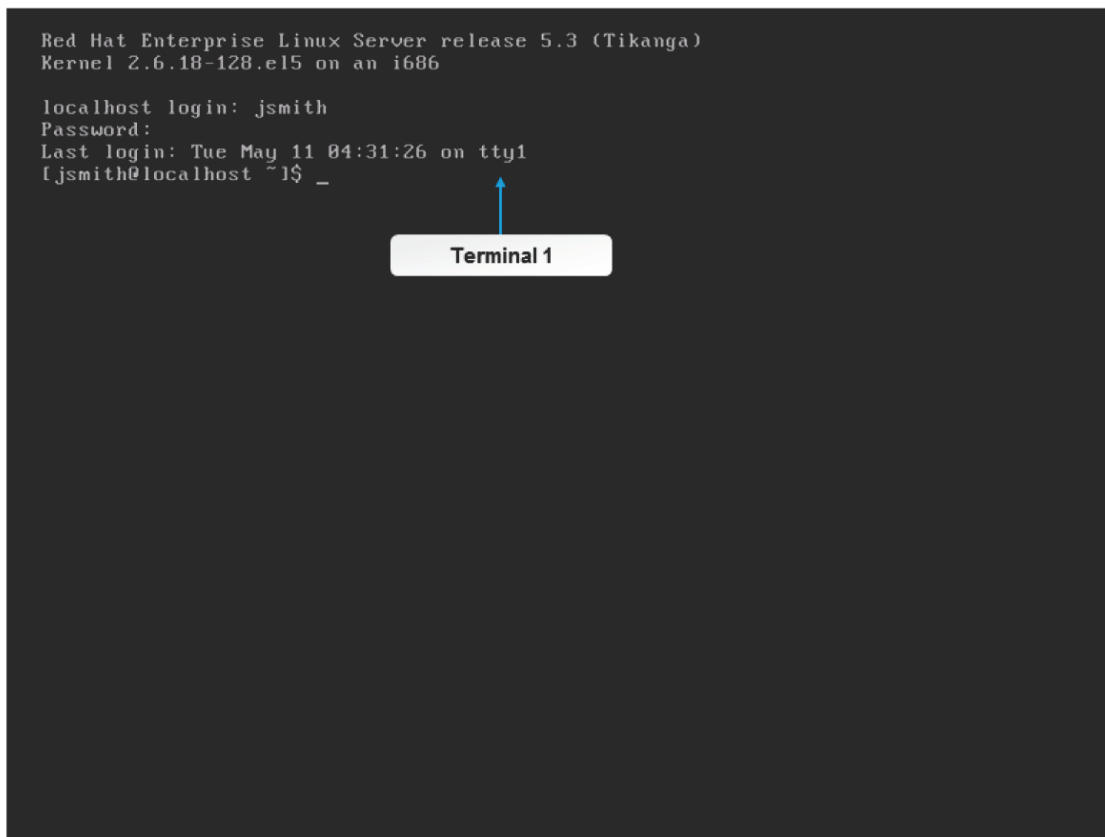
Some options for the cal command are listed in the table.

Option	Description
-m	Displays Monday as the first day of the week.
-j	Displays the Julian dates.
-y	Displays the current year's calendar.

Virtual Terminals

A **terminal** or **console** is a computer interface for text entry and display, where information is displayed as an array of preselected characters. Linux supports six virtual terminals in the CLI mode, which provide a text terminal with a

login prompt to the shell. You can choose from among these six terminals by using the key combination of **Ctrl+Alt+F1–F6**. You can be logged in to multiple virtual terminals at the same time.

A terminal window titled 'Terminal 1' with a dark background. The text inside shows the login process for user 'jsmith' on a Red Hat Enterprise Linux Server. The prompt is '[jsmith@localhost ~]\$_'.

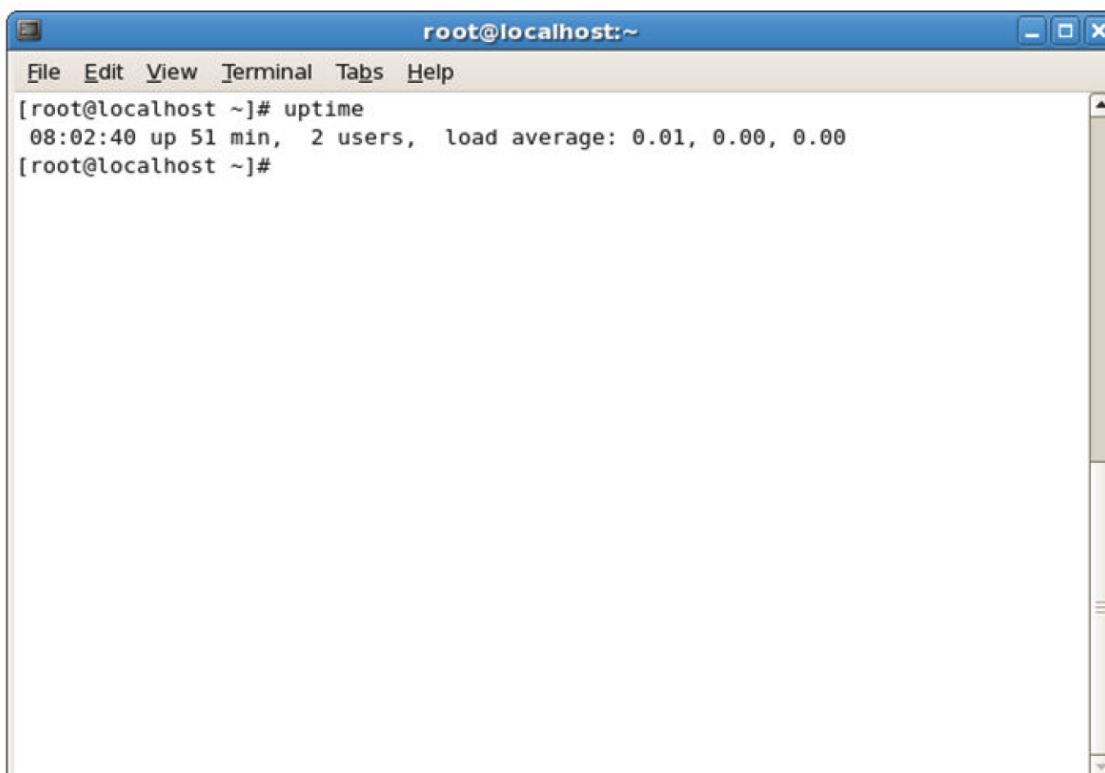
```
Red Hat Enterprise Linux Server release 5.3 (Tikanga)
Kernel 2.6.18-128.el5 on an i686

localhost login: jsmith
Password:
Last login: Tue May 11 04:31:26 on tty1
[jsmith@localhost ~]$_
```

Figure 1-11: Terminal 1 with the user *jsmith* logged in.

The uptime Command

The `uptime` command displays the time from when a system started running. The output of the `uptime` command gives information about the current time, how long the system is running, and how many users are currently logged in.

A terminal window titled 'root@localhost:~' with a light background. The text shows the output of the 'uptime' command. The prompt is '[root@localhost ~]#'.

```
root@localhost:~
File Edit View Terminal Tabs Help

[root@localhost ~]# uptime
08:02:40 up 51 min, 2 users, load average: 0.01, 0.00, 0.00
[root@localhost ~]#
```

Figure 1-12: Using the uptime command to view the time from when the system started running.

Load Average

The last field of the uptime command output displays the system's load averages for the last 1 minute, 5 minutes, and 15 minutes. This information can be used to check whether the system is busy.

The who Command

The who command is used to determine the details of users currently logged in to a system. The output of the who command includes the user name, the name of the system from which the user is connected, and the time since the user is connected.

```
[root@srv02 ~]# who
root      :0                2015-04-01 11:13 (:0)
root      tty2             2015-04-01 11:12
root      pts/0            2015-04-01 11:13 (:0)
jsmith    :1                2015-04-01 11:20 (:1)
[root@srv02 ~]# _
```

Figure 1-13: Displaying user details using the who command.


who Command Options

The -i option can be used to see how long users have been idle. A dot indicates that the users were active up to the last minute, old indicates that the users have been inactive for over 24 hours, and anything between 2 minutes and 23 hours 59 minutes shows the length of time they have been idle.

The am i option displays information only for the user who runs the command. The output is preceded by the hostname.

The whoami Command

The whoami command is used to display the user name with which you are currently logged in to the system. Sometimes, you may need to log in to a system and switch among different users, and you may not be sure with which user you are currently logged in. In such instances, you can use the whoami command to know your current user name.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command '[root@localhost ~]# whoami' has been entered, and the output 'root' is displayed on the next line. The prompt '[root@localhost ~]#' is shown again with a cursor.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# whoami  
root  
[root@localhost ~]#
```

Figure 1-14: *Displaying the user name using the whoami command.*

The hostname Command

The hostname command is used to display the hostname of the system you are currently logged in to. When you log in to different systems using the same terminal, you can use the hostname command to identify the system on which you are presently running the commands.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command '[root@localhost ~]# hostname' has been entered, and the output 'localhost.localdomain' is displayed on the next line. The prompt '[root@localhost ~]#' is shown again with a cursor.

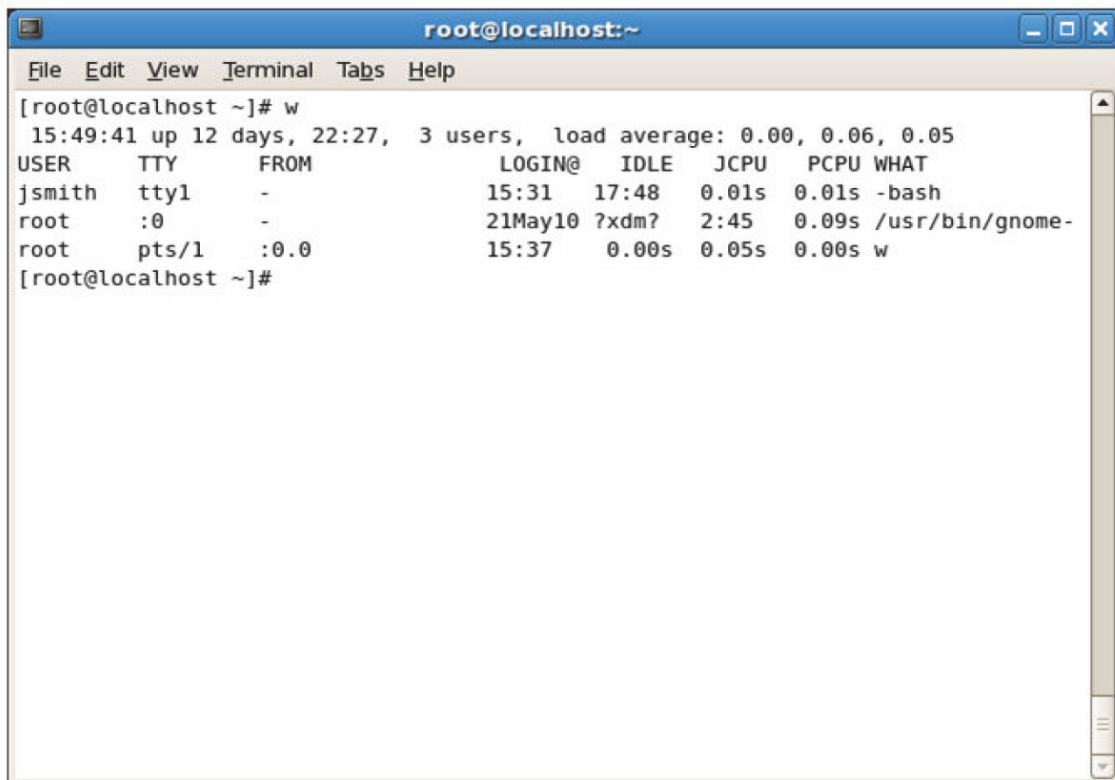
```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# hostname  
localhost.localdomain  
[root@localhost ~]#
```

Figure 1-15: *Viewing the hostname of the Linux system using the hostname command.*

The w Command

The w command is primarily used to display the details of users who are currently logged in to a system and their transactions. The first line of the output displays the status of the system. The second line of the output displays a

table with the first column listing the users logged in to the system and the last column indicating the current activities of the users. The remaining columns of the table show different attributes associated with the users.



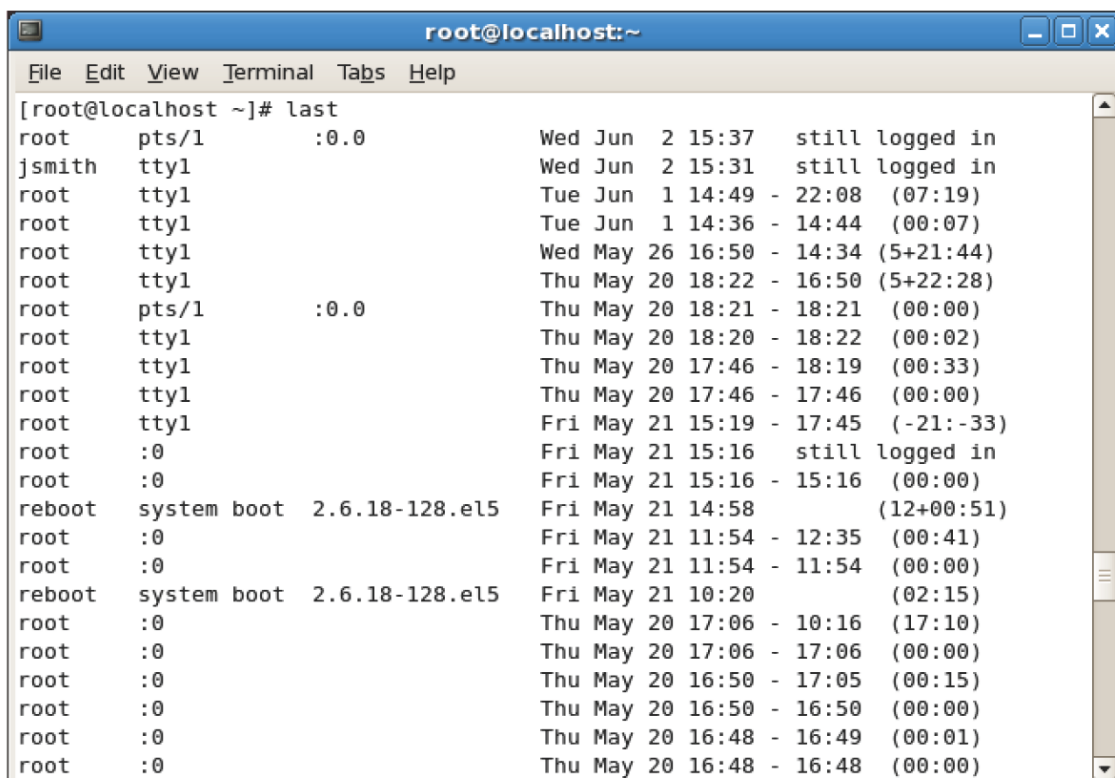
```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# w  
15:49:41 up 12 days, 22:27, 3 users, load average: 0.00, 0.06, 0.05  
USER      TTY      FROM          LOGIN@  IDLE   JCPU   PCPU WHAT  
jsmith    tty1     -             15:31   17:48  0.01s  0.01s -bash  
root      :0       -             21May10 ?xdm?   2:45   0.09s  /usr/bin/gnome-  
root      pts/1    :0.0          15:37   0.00s  0.05s  0.00s w  
[root@localhost ~]#
```

Figure 1-16: Viewing user details using the `w` command.

The last Command

The last command displays the history of user log in and log out, along with the actual time and date. It also has options that enable you to filter users who have logged in through a specific terminal. For example, last 1 will display the details of users who logged in using the first terminal.

The last command retrieves information from `/var/log/wtmp` file.

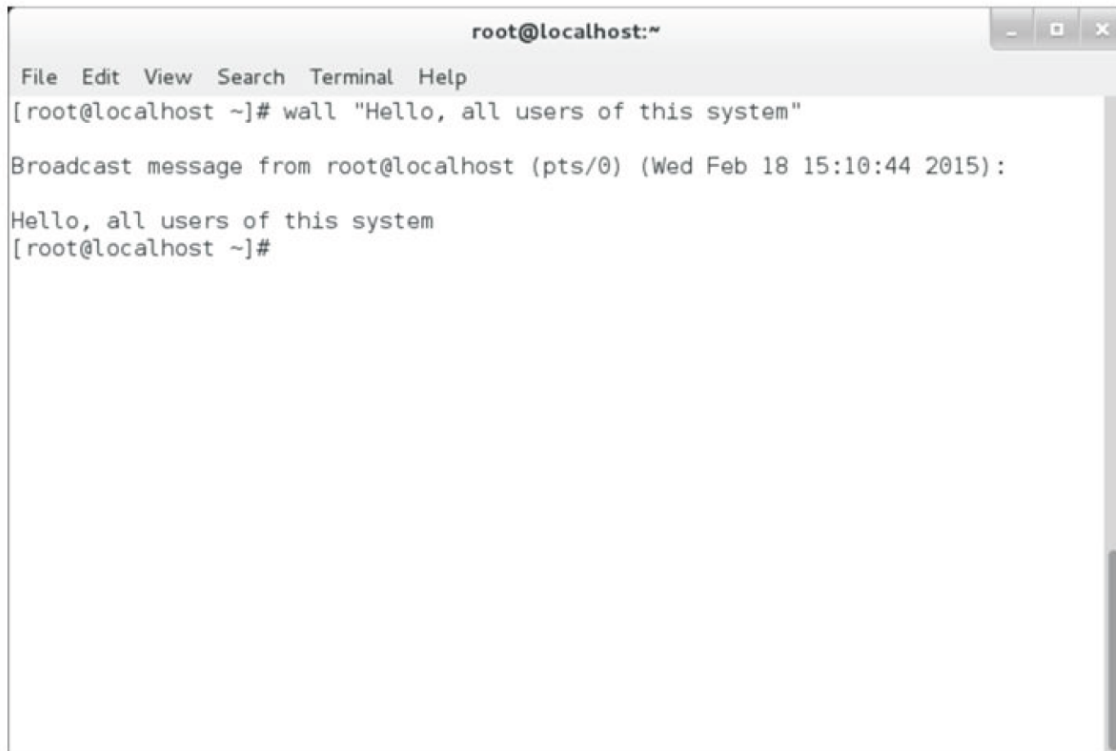


```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# last  
root      pts/1    :0.0          Wed Jun 2 15:37   still logged in  
jsmith    tty1     -             Wed Jun 2 15:31   still logged in  
root      tty1     -             Tue Jun 1 14:49 - 22:08 (07:19)  
root      tty1     -             Tue Jun 1 14:36 - 14:44 (00:07)  
root      tty1     -             Wed May 26 16:50 - 14:34 (5+21:44)  
root      tty1     -             Thu May 20 18:22 - 16:50 (5+22:28)  
root      pts/1    :0.0          Thu May 20 18:21 - 18:21 (00:00)  
root      tty1     -             Thu May 20 18:20 - 18:22 (00:02)  
root      tty1     -             Thu May 20 17:46 - 18:19 (00:33)  
root      tty1     -             Thu May 20 17:46 - 17:46 (00:00)  
root      tty1     -             Fri May 21 15:19 - 17:45 (-21:-33)  
root      :0       -             Fri May 21 15:16   still logged in  
root      :0       -             Fri May 21 15:16 - 15:16 (00:00)  
reboot    system boot  2.6.18-128.el5 Fri May 21 14:58   (12+00:51)  
root      :0       -             Fri May 21 11:54 - 12:35 (00:41)  
root      :0       -             Fri May 21 11:54 - 11:54 (00:00)  
reboot    system boot  2.6.18-128.el5 Fri May 21 10:20   (02:15)  
root      :0       -             Thu May 20 17:06 - 10:16 (17:10)  
root      :0       -             Thu May 20 17:06 - 17:06 (00:00)  
root      :0       -             Thu May 20 16:50 - 17:05 (00:15)  
root      :0       -             Thu May 20 16:50 - 16:50 (00:00)  
root      :0       -             Thu May 20 16:48 - 16:49 (00:01)  
root      :0       -             Thu May 20 16:48 - 16:48 (00:00)
```

Figure 1-17: Viewing the history details of user logins.

The wall Command

The wall command sends a message to all currently logged in users. The length of the message is limited to 20 lines, and it is typically used to inform all currently logged-in users on a multi-user Linux system that a system event is about to occur. For example, a system administrator may use the wall command to notify users that a printer attached to the system will be shutdown for maintenance.

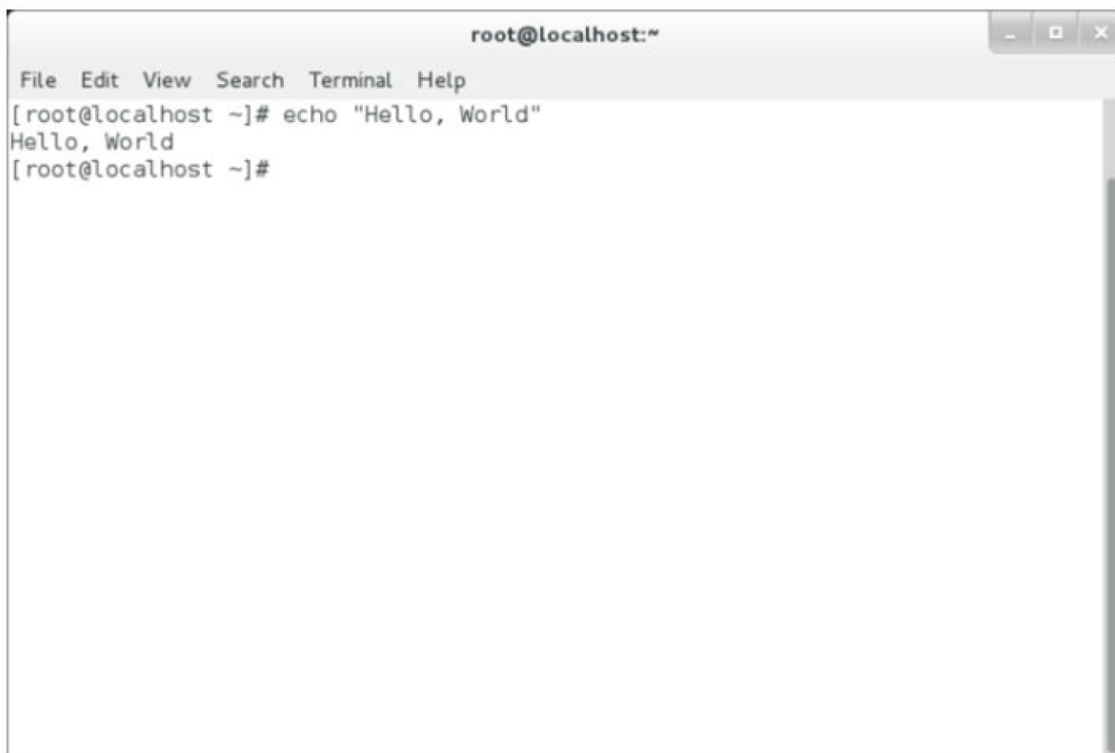
A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command '[root@localhost ~]# wall "Hello, all users of this system"' has been entered. The output shows a broadcast message from root@localhost (pts/0) on Wed Feb 18 15:10:44 2015, followed by the message 'Hello, all users of this system'. The prompt '[root@localhost ~]#' is shown again.

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# wall "Hello, all users of this system"  
Broadcast message from root@localhost (pts/0) (Wed Feb 18 15:10:44 2015):  
Hello, all users of this system  
[root@localhost ~]#
```

Figure 1-18: The wall command sends a message to every user who is logged i.

The echo Command

The echo command is used to display a line of text on the terminal. It is useful for programmers writing shell scripts because it can be used to display additional information. The text that needs to be displayed should be inserted after the echo command. You can also use the echo command to display the value stored in a variable by specifying the variable name after the echo command.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command '[root@localhost ~]# echo "Hello, World"' has been entered, and the output 'Hello, World' is displayed on the next line. The prompt '[root@localhost ~]#' is visible at the bottom.

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# echo "Hello, World"  
Hello, World  
[root@localhost ~]#
```

Figure 1-19: Displaying text using the echo command.

Syntax

The syntax of the echo command is `echo {"string"}`.

The sleep Command

The sleep command is used to pause system activities for a specified time. The command `sleep {time}` hangs up the prompt for the number of seconds specified by the value of the variable **time**.


A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command '[root@localhost ~]# sleep 65' has been entered. The terminal is currently blank, indicating the command is running and pausing the prompt.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# sleep 65
```

Figure 1-20: Pausing activities using the sleep command.

The cat Command

The cat command displays, combines, and creates text files. This command is frequently used to read small text files.



Note: The name of the cat command is a short form of the word concatenate.

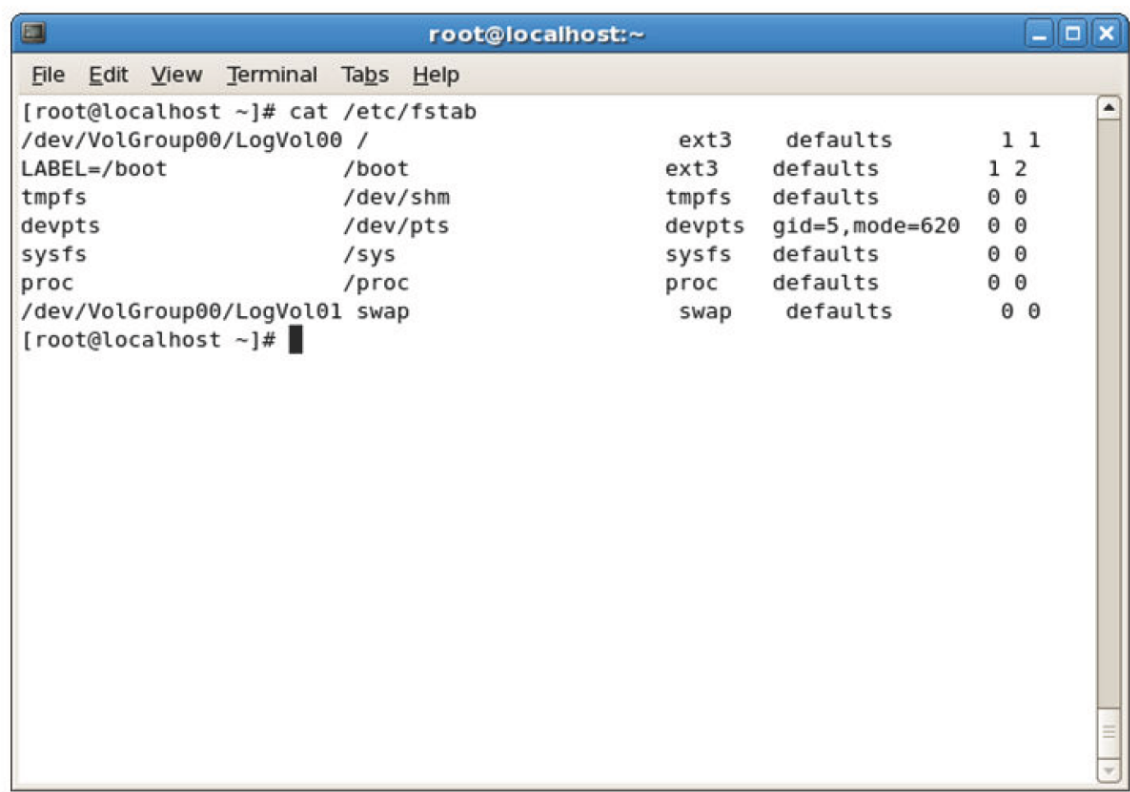


Figure 1-21: The cat command displaying a text file.

The cat command options are described in the following table.

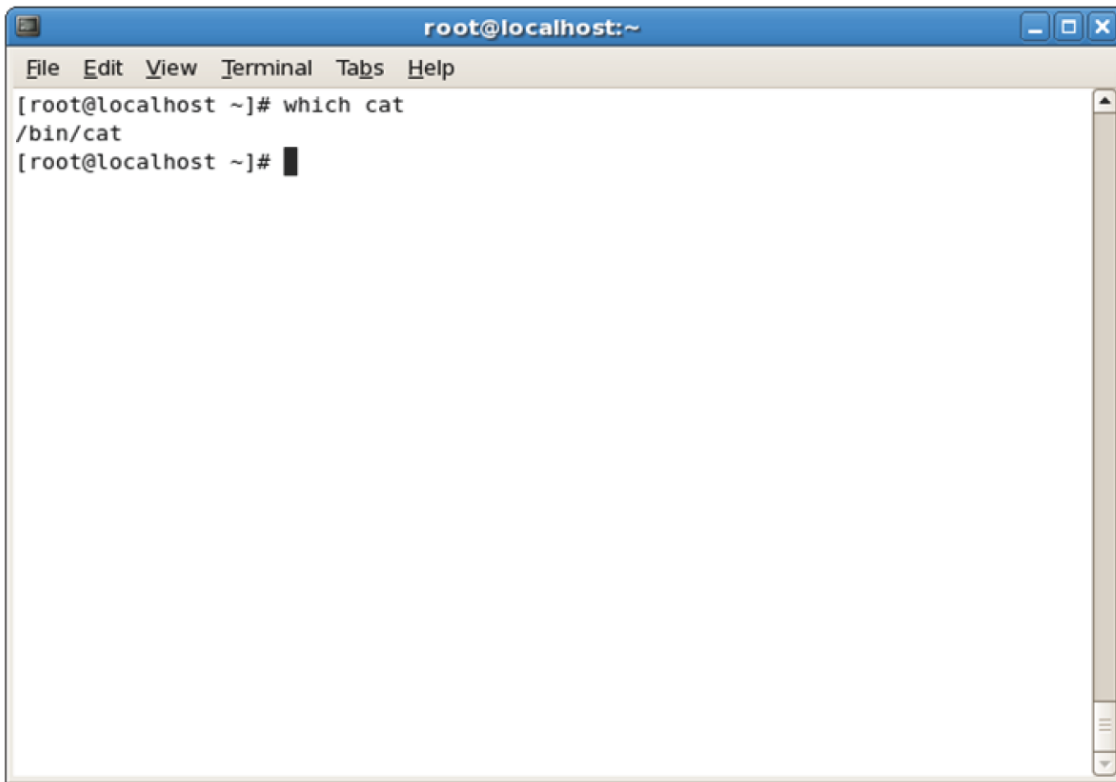
Option	Description
-n	Precedes the output with its respective line number.
-b	Numbers the lines, excluding the blank lines.
-u	Omits to buffer the output.
-s	Omits to display results for nonexistent files.
-v	Displays nonprinting characters as visible characters, other than tabs, new lines, and form-feeds.
-e	Prints a \$ character at the end of each line, prior to the new line.
-t	Prints tabs as ^I and form-feeds as ^L.

Syntax

The syntax of the cat command is `cat [command options] {file name}`.

The which Command

The which command is used to verify whether a user has the right to execute a command. It displays the complete path of the command by searching the directories assigned to the [PATH variable](#). For example, on entering which cat, the following output is displayed: /bin/cat.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the command '[root@localhost ~]# which cat' and its output '/bin/cat'. The prompt '[root@localhost ~]#' is followed by a cursor.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# which cat  
/bin/cat  
[root@localhost ~]#
```

Figure 1-22: The *which* command displays the complete path of the command.

How to Enter Shell Commands

Follow these general procedures to enter shell commands.

Log in to Your System

To log in to your system:

1. Log in to the GUI of the system.
 - a. Select the user's full name on the Login screen.
 - b. In the **Password** text box, enter the password of the user.
2. Log in to the CLI of the Linux system.
 - a. Press **Ctrl+Alt+F2** to switch to the first text terminal (the second terminal).
 - b. To log in to the system, enter the user name.
 - c. Enter the password.

Monitor User Logins

To monitor user logins:

1. Log in as **root**.
2. Monitor user logins in the system.
 - To display the login information about the connected users and the processes associated with those users, enter **w**.
 - To display the users who are currently logged in to the system, enter **who**.

Check the System Date and Calendar Using Commands

To check the system date and calendar:

1. Log in to the CLI of the Linux system.
2. To view the date details, in the terminal, enter the date commands.
 - To check the current date and time on the system, enter `date`.
 - To view the date in the month-date-year format, enter `date +%m-%d-%y`.
 - To view the current time in the hour-minute-second [AM or PM] format, enter `date +%r`.
3. To view a specific calendar, enter the `cal` command.
 - To display the current month's calendar, enter `cal`.
 - To view a specific month's calendar, enter `cal {month} {year}`.
 - To display the calendar in a specific format, enter `cal [option]`.



Note: In addition, in the GUI, you can select **System** → **Administration** → **Date & Time** to view the system date and time in the **Date/Time Properties** dialog box.

4. If necessary, to clear the terminal screen, enter the clear command.

Display System Information Using Commands

To display the system information:

1. Enter a suitable command to view specific system information:
 - To check the duration since the system is running, enter `uptime`.
 - To display the history of logins, enter `last`.
 - To display the history of bad logins on the system, enter `lastb`.
 - To display the hostname of the system you are currently logged in to, enter `hostname`.
 - To display the user name with which you are currently logged in, enter `whoami`.

TOPIC C Get Help Using Linux

Now that you are familiar with the Linux shell, you may want to begin using commands in your system. However, you may need assistance with the various commands available. In this topic, you will identify the help and support options offered by Linux.

By learning about Linux support options, you can increase your access to information about the Linux environment. Doing so will help you support your implementation of Linux. The information provided in the Linux documentation will enable you to easily troubleshoot problems you encounter.

Linux Documentation

Linux documentation is the material that provides information on various Linux commands and blocks of code. Some Linux documentation is available in electronic format and some in print format. Linux documentation is available from sources such as manual pages, online resources, published works, Usenet newsgroups, and mailing lists.

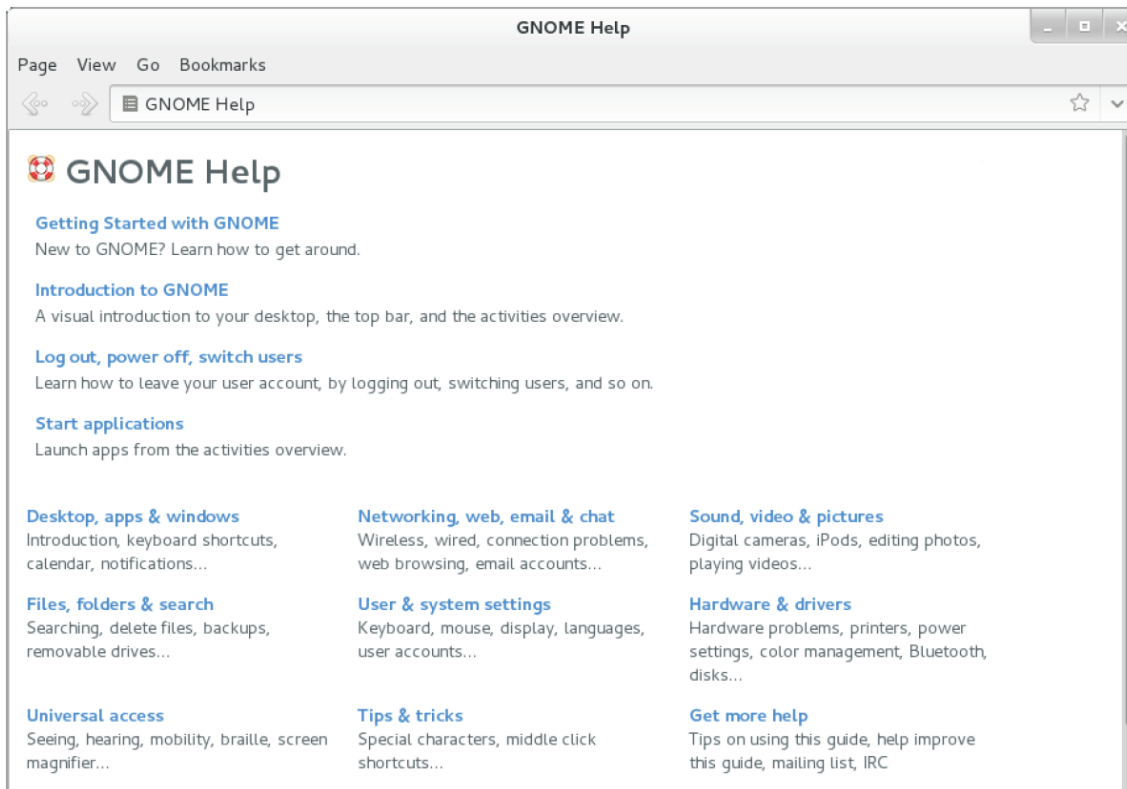


Figure 1-23: Built-in Linux help in the GUI.

System Documentation

System documentation is the term given to the collection of documents that list the system requirements; its functioning capabilities, limitations, design specifications; the internal workings of the system; and the steps for maintaining the system.

Manual Pages

The Linux **manual pages**, or man pages, contain the complete documentation that is specific to every Linux command; they are presented in simple ASCII text format. The man page for a specific command is displayed using the man command. The man pages are available on the system by default. They usually include information such as the name of the command, its syntax, a description of its purpose, the options it supports, examples of common usage of the command, and a list of related commands.

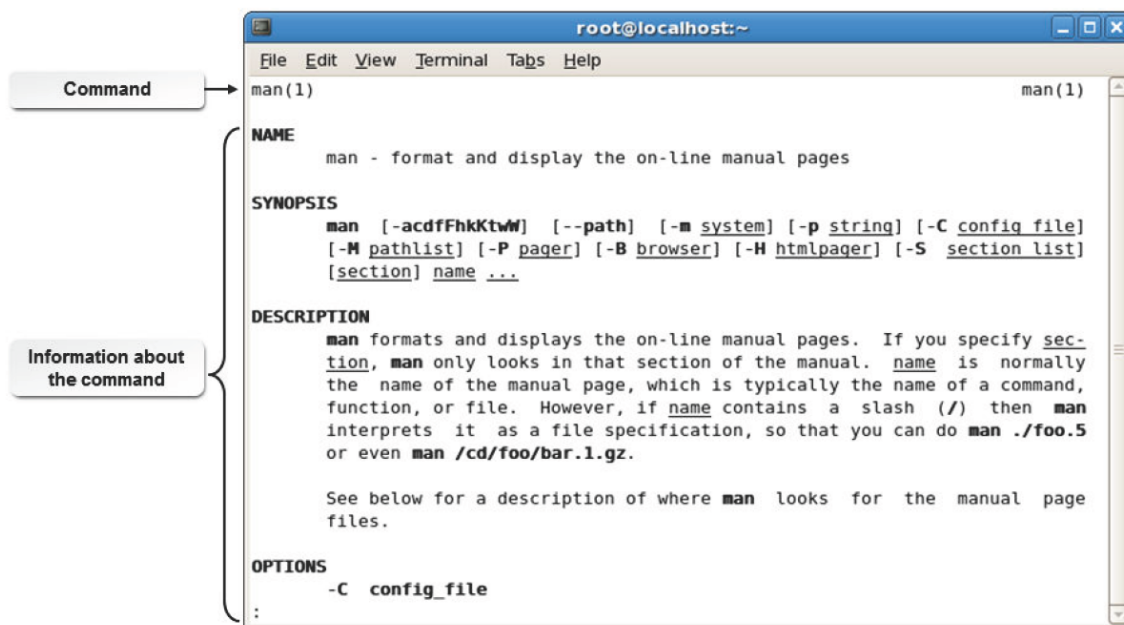


Figure 1-24: Viewing information on the manual pages.

Syntax

The syntax of the `man` command is `man {topic}`.

The man Command Options

The `man` command supports different options. Some of the frequently used options are listed here.

Option	Description
-a	Finds all entries matching the query.
-D	Displays debugging information.
-f	Displays a short description of the command along with the man pages/ sections.
-h	Displays help options for the man command.
-k	Lists all manual pages/sections containing the keyword along with their location.
-K	Searches for the specified string on all pages.
-t	Formats the man pages to enable printing.

Man Page Sections

Man pages for a single command may be listed under several sections. All the available manual page sections for a particular command can be listed using the `whatis` command. When a command has more than one section listed, it means that documentation for the same command is available from more than one source. These sections are identified by the number displayed beside the command, for example, `fsck (8)`.

Various man page sections are given in the following table.

Section Number	What It Contains
1	General commands
2	System calls
3	C library functions
4	Special files (usually found in <code>/dev</code>)

Section Number	What It Contains
5	File formats and conventions
6	Games and screensavers
7	Miscellaneous
8	System administration commands and daemons

Keys to Navigate Through Linux Man Pages

You can navigate through the Linux man pages using a number of keys. The functions of different keys are given in the following table.

Key	Used To
Home	Move to the beginning of the man page.
End	Move to the end of the man page.
Page Up	Scroll up the page progressively.
Page Down	Scroll down the page progressively.
/	Begin a search for a term or text string.
n	Move to the next occurrence of the search term.
p	Move to the previous occurrence of the search term.
q	Quit and return to the shell prompt.

The apropos Command

The apropos command is generally used when a user does not know which command to use to perform a certain action. It can be used with a keyword to display a list of the manual pages containing the keyword along with their man page sections. The apropos command searches a regularly updated database called the *whatis* database for the specified string and returns all matching entries.

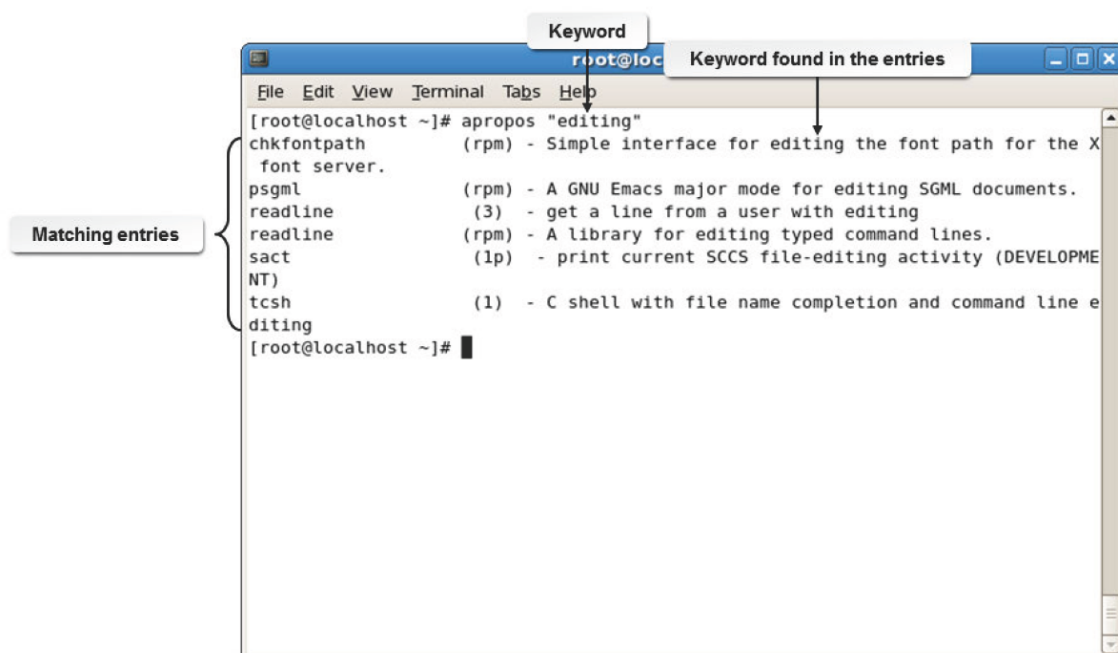


Figure 1-25: Searching for help using the apropos command.

Syntax

The syntax of the apropos command is `apropos {keyword}`.

Other Built-In Help Options

In addition to the `man` and `apropos` commands, Linux offers other built-in options for help.

<i>Help Option</i>	<i>Description</i>
<code>whatis</code>	Displays a short description of the command along with the man pages/sections matching the exact command. This command searches a regularly updated database for documentation. The syntax of this command is <code>whatis {command}</code> .
<code>info</code>	Displays info pages containing additional or recent information about a command. The syntax of this command is <code>info {command}</code> .
<code>command --help</code>	Displays a quick summary of the usage of a command and a list of arguments that can be used. This feature can be used with most commands in Linux. The syntax of this command is <code>command -options</code> .

The `/usr/share/doc` Directory

The `/usr/share/doc` directory contains documents installed on the system, describing in detail certain aspects of configuring or using Linux.

HOWTOs

HOWTO documents may be installed on your system, usually under the `/usr/share/doc` directory.

The HTML version of the files can be displayed in any web browser, including the text-based Lynx browser. The text files can be viewed through any text editor or by using the display commands such as `cat`, `more`, or `less`. HOWTOs can be found on most systems and can also be found on the web at <http://linuxdocs.org/HOWTOs/HOWTO-INDEX/howtos.html> and other sites.

HOWTOs are comprehensive documents, much like FAQs, but generally not in question-and-answer format. However, many HOWTOs contain a FAQ section at the end. There are several HOWTO formats available: plain text, PostScript, PDF, and HTML. In addition to the HOWTOs, there are a multitude of mini-HOWTOs on short, specific subjects.

Getting Help from Info Pages

To display info pages, enter `info`, with or without options and arguments. By itself, the command `info` will display the help file on how to work with info pages. Entering `info [topic]` will display the info page for the specified topic. The command `info --help` displays a brief help description.

When the info page is displayed, any text with an asterisk (`*`) in front of it is a link. Move your cursor (using the arrow keys) to the text, and then press **Enter** to access the linked info page. To return to the previous document, type **U** and then type **D** to return to the top of the page. Type **Q** to return to the command prompt.

LUGs

A good source of information for Linux users and developers is Linux User Groups, or LUGs.

These can be virtual (based on the web) or there may be a group of people who meet in your neighborhood. The virtual ones sometimes take the form of a message board with a question-and-answer database.

Online Help

The Internet is the best place to get documentation for any distribution of Linux. There are dedicated websites and online forums that help Linux users with specific distributions or Linux in general. Documentation for commercial

distributions is available in their respective official websites.

These include the release notes of different versions and updates, the deployment guide, the installation guide, and the virtualization guide. For example, the Red Hat documentation can be accessed from the URL, www.redhat.com/docs/manuals/enterprise.



Figure 1-26: The Red Hat online documentation.

How to Access Help in Linux

Follow these general procedures to access help in Linux.

View Linux man Pages

To view Linux man pages:

1. Enter `man {command}` at the command line, where {command} is the command for which you want to view the man page.
2. View the list of command options available for the command.
3. Close the man page for the specified command.

Display the man Page for a Command

To display the man page for a command:

1. Log in as a user.
2. To display the man page for a specific command under a specific section, enter `man [section] {command}`.
3. Navigate through the man pages.
 - To navigate within a page, use the **Up Arrow** or **Down Arrow** key.

- To navigate through several pages, use **Page Up** or **Page Down**.
 - Search for some specific topic in the man pages as required.
 - a. To search through the man page for the specified string, enter /search string.
 - b. If necessary, to locate the next occurrence of the string in the man page, type n.
4. To close the man page, press **q**.

Use the *whatis* Command

To use the *whatis* command:

1. Log in as a user.
2. To build the *whatis* help database, enter makewhatis
3. To display the man page sections and a short description of the specified command, enter *whatis {command name}*.
4. To view the man page for the specified command under the specified section, enter man *[section]{command}*.
5. To close the man page, press **q**.

Find the Relevant man Pages Using the *apropos* Command

To find the relevant man pages using the *apropos* command:

1. Log in as a user.
2. To search the keyword in the *whatis* database and display the matching man page sections for the specified keyword, enter *apropos {keyword}*.
3. To display the man page for a specific command under a specific section, enter man *[section] {command}*.
4. To close the man page and return to the command prompt, press **q**.

Display the Info Documents of a Command

To display the info documents of a command:

1. Log in as a user.
2. To read the info documents for the specified command, enter info command.
3. Navigate through the info pages.
 - To navigate within a page, use the **Up Arrow** or **Down Arrow** key.
 - To navigate through several pages, use **Page Up** or **Page Down**.
 - To move to the next or previous page, type n or p, respectively.

- To search for a particular string on the info pages, type `s`.
- To go to the next link, press **Tab**.

4. To close the info page, press `q`.

Display the Options of a Command

To display the options of a command:

1. Log in as a user.
2. To display the command syntax and a list of options, enter command `--help`.
3. To use the necessary option to execute the required task, enter command `-options`.

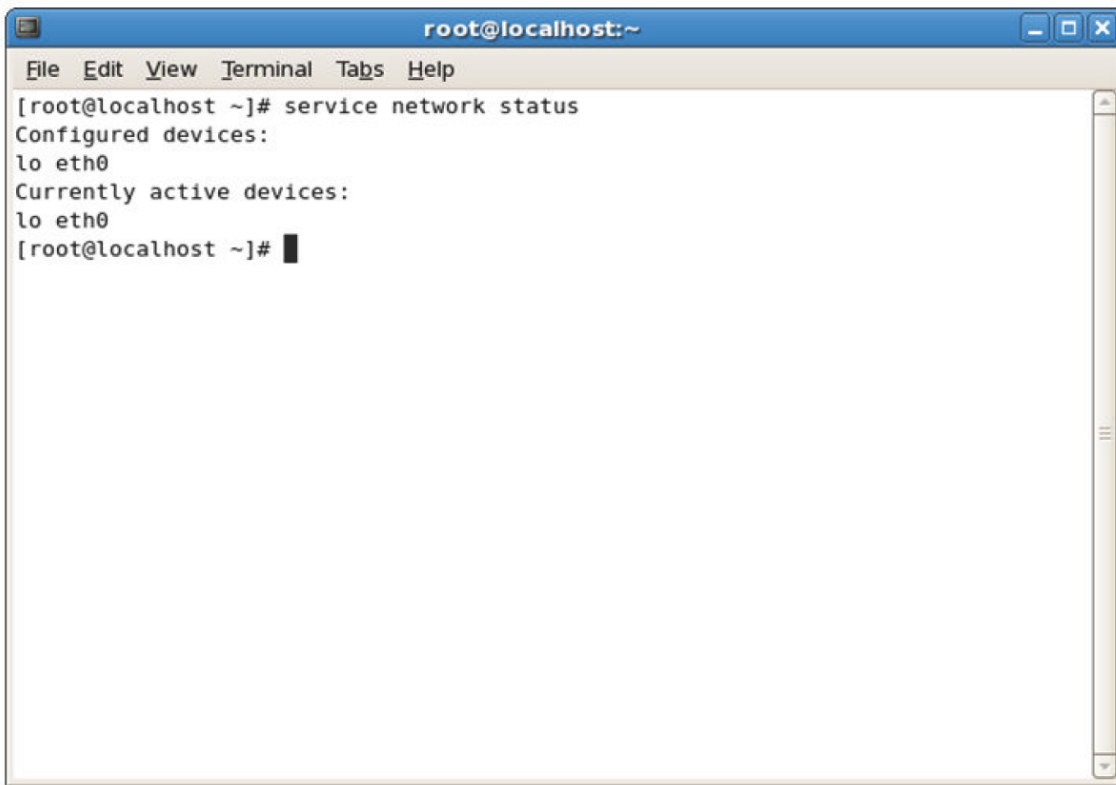
TOPIC D Start and Stop Linux

In the last topic, you identified the help and support options offered by Linux and accessed documentation in the shell. But you cannot troubleshoot system problems without knowing how to start and stop your system. In this topic, you will start and stop the Linux system.

We all expect our operating systems to load and run the necessary processes at boot time. There are occasions, however, when you, as a Linux administrator, may want to start, stop, or restart the system manually. The ability to manage these essential services will help you perform maintenance and upgrades on your system.

Services

A Linux **service** is an application or set of applications that perform tasks in the background. The services running on a Linux system range from basic services to server services. Services can be broadly classified as critical services and noncritical services. Critical services are the core services that are vital for the functioning of the Linux system. Noncritical services are services that are initiated by applications installed on the system.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command 'service network status' has been executed, showing the status of the network service. The output indicates that the configured and currently active devices are both 'eth0'.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service network status  
Configured devices:  
lo eth0  
Currently active devices:  
lo eth0  
[root@localhost ~]#
```

Figure 1-27: The network service displaying its status.

The service Command

The service command allows you to manage services running on your system. The syntax of the service command is: `service {service name} {options}`.

Daemons

A **daemon** is a program that runs in the background without the need for human intervention, often handling commands delivered for remote command execution. It lies dormant until an event triggers it into activity. Some daemons operate at regular intervals. Most daemons are started when the system boots. Daemons are started by the operating system, by applications, or manually.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command 'service syslog status' has been executed, showing the status of the syslog service. The output indicates that 'syslogd (pid 1765) is running...' and 'klogd (pid 1768) is running...'. A callout box labeled 'Daemons' points to the output lines.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service syslog status  
syslogd (pid 1765) is running...  
klogd (pid 1768) is running...  
[root@localhost ~]#
```

Daemons

Figure 1-28: Daemons running in the background for the syslog service.

lpd

The Line Printer Daemon, or lpd, controls the flow of print jobs to a printer. It works in the background and sends the output to the printer without affecting other processes that a user is working on at that time.

Init and Runlevels

init is the first process run on boot, and is responsible for managing the **runlevel** of your system.

The init man page says "Init is the parent of all processes." and it is the system and session manager.

There are several Init daemon types, which vary based on the Linux distribution in use:

- SysV-style Init, or SysVinit (CentOS/RHEL 6 and earlier)
- Upstart Init Daemon (Ubuntu, Fedora 9 - 14)
- Systemd Init Daemon (CentOS/RHEL 7, Fedora 15+)

The runlevel specifies the group of processes that are started, stopped, and otherwise managed on Linux systems. SysVinit scripts create processes at system boot time from a script in the **/etc/ inittab** file. You can change the current runlevel by using the telinit command.



Figure 1-29: The different runlevels that can be specified with the telinit command.

Init and Systemd Runlevels Table

The following table describes the processes that can run at each init level.

<i>SysVinit Runlevel</i>	<i>Systemd Target</i>	<i>Description</i>
0	poweroff.target	Halts the system

<i>SysVinit Runlevel</i>	<i>Systemd Target</i>	<i>Description</i>
1	rescue.target	Single-user mode
2	multi-user.target	Multiuser mode without networking
3	multi-user.target	Multiuser mode with networking
4	multi-user.target	User configurable
5	graphical.target	Used for the GUI (X11 multiuser mode)
6	reboot.target	Reboots the system

The Upstart Init Daemon

The traditional SysVinit system has been replaced by an improved version of init known as **Upstart** in some Linux distributions. The Upstart init system is event-based rather than runlevel-based.

Event-based means that jobs will be automatically started and stopped by changes to the system's state. The original SysVinit was dependency-based and jobs had to start in a particular order. The Upstart init daemon doesn't track runlevels. Runlevels are tracked by the runlevel event generated by telinit or shutdown. The init daemon sets two environment variables from the runlevel event:

RUNLEVEL and PREVLEVEL. These environment variables are the current runlevel and the previous runlevel. Upstart's list of configuration files is located in the **/etc/init** directory.

The Systemd Init Daemon

Systemd is a replacement for SysVinit and Upstart in some Linux distributions, including CentOS/ RHEL 7. It allows for greater concurrency (starting programs at the same time for quicker boot) and reduces shell overhead. Systemd has some advantages over the SysVinit and Upstart systems by allowing aggressive parallelization, including socket and D-Bus activation for starting services.

Control Groups (cgroups) are used to track processes instead of Process IDs (PIDs), which provides better isolation for processes.

The systemctl Command

The systemctl command enables control over the Systemd Init process. You can view running services, manage (enable/disable) services to run at boot or in the current session, determine the status of these services, and manage the system runlevel.

```

root@localhost:~
File Edit View Search Terminal Help
ModemManager.service      loaded active running Modem Manager
network.service           loaded active exited LSB: Bring up/down networkin
NetworkManager.service    loaded active running Network Manager
nfs-lock.service          loaded active running NFS file locking service.
polkit.service            loaded active running Authorization Manager
postfix.service           loaded active running Postfix Mail Transport Agent
rhel-dmesg.service        loaded active exited Dump dmesg to /var/log/dmesg
rhel-import-state.service loaded active exited Import network configuration
rhel-loadmodules.service  loaded active exited Load legacy module configura
rhel-readonly.service     loaded active exited Configure read-only root sup
rngd.service              loaded active running Hardware RNG Entropy Gather
rpcbind.service           loaded active running RPC bind service
rsyslog.service           loaded active running System Logging Service
rtkit-daemon.service      loaded active running RealtimeKit Scheduling Polic
smartd.service            loaded active running Self Monitoring and Reportin
sshd.service              loaded active running OpenSSH server daemon
sysstat.service           loaded active exited Resets System Activity Logs
systemd-fsck-root.service loaded active exited File System Check on Root De
systemd-...a9ee7d3d.service loaded active exited File System Check on /dev/di
systemd-hostnamed.service loaded active running Hostname Service
systemd-journald.service  loaded active running Journal Service
systemd-locale.service    loaded active running Locale Service
systemd-logind.service    loaded active running Login Service
lines 70-92

```

Figure 1-30: The systemctl command is used to list all running services on a system.

Some of the common systemctl command options and their descriptions are given in the following table. For administrators migrating from a SysVinit environment to Systemd, excellent documentation about systemd for CentOS/RHEL 7 is available at <https://access.redhat.com/articles/754933> and <https://access.redhat.com/products/red-hat-enterprise-linux/systemd-intro>

Command	Allows You To
systemctl	List all services on the server and their current status.
systemctl list-unit-files	List all installed unit files and a brief status (enabled, disabled, static).
systemctl isolate {new runlevel target}	Change the current target (runlevel) of the Linux system. For example, systemctl isolate multi-user.target would change the current runlevel to runlevel 3. Alternatively, you can use the telinit 3 command to modify the current runlevel.
systemctl start service	Start (activate) a service immediately.
systemctl stop service	Stop (deactivate) a service immediately.
systemctl restart service	Restart a service immediately.
systemctl status service	Show the status of a service, and whether it is running or not.
systemctl enable service	Enable a service to be started on boot.
systemctl disable service	Disable a service so that it is no longer started on boot.

Syntax

The syntax of the systemctl command is `systemctl [options][command]`.

System Booting

During the installation of Linux, the boot loader you choose will be put on the **Master Boot Record (MBR)**. GRand Unified Bootloader (GRUB) is the Linux boot loader that loads and starts the kernel. Only one boot loader can be used on a system.

Boot Loader

A **boot loader** is a program that loads the kernel so that Linux and other operating systems can boot.

System Shutdown

The **shutdown** command is used to close a system. This closes files and performs other tasks necessary to safely shutdown the system. It warns all users that the system is going to shutdown and no one can log in after the command is issued. After certain installations or removal of hardware, it is necessary to shutdown the Linux system.

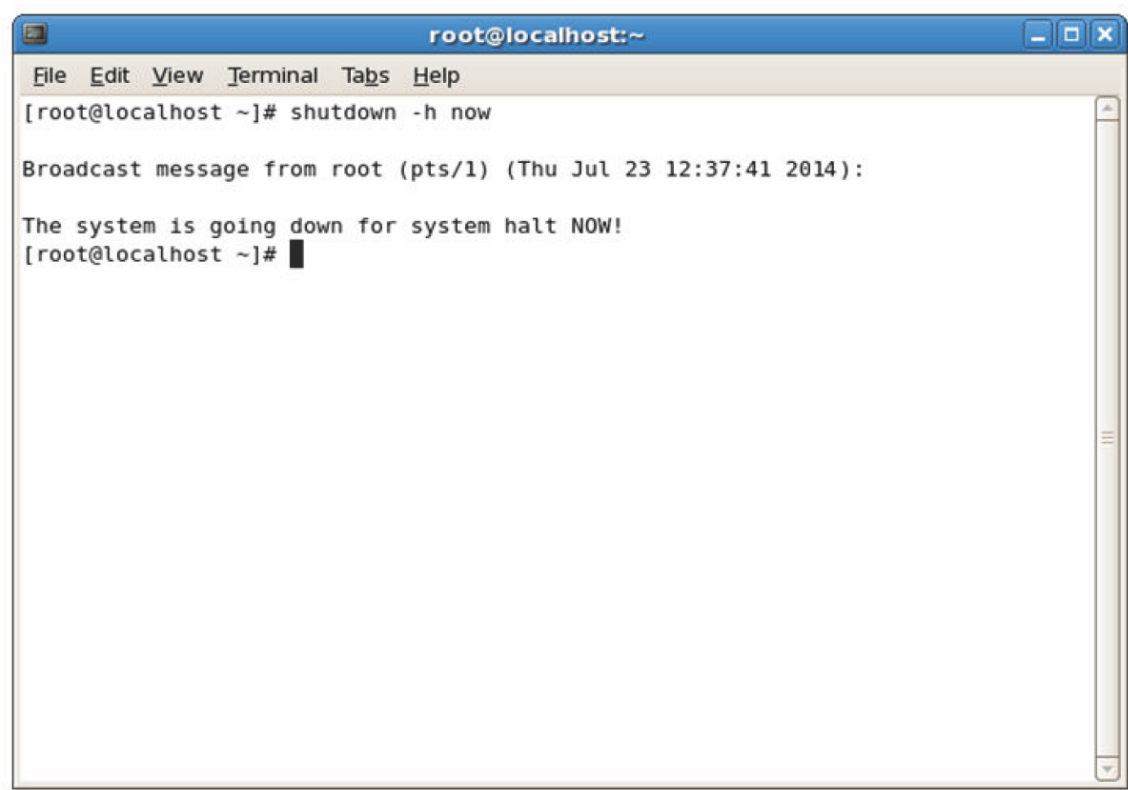


Figure 1-31: Shutting down a Linux system.

Shutdown Command Options

When you need to restart your Linux system, you should use the shutdown command with the appropriate options. The syntax of the shutdown command is shutdown [-t seconds] [-

options] time [warning message].

The -t option specifies how many seconds to wait before changing to another runlevel. The other options are listed in the following table.

Option	Used To
-k	Send warning messages to everyone, but does not really shutdown the system.
-r	Reboot the system after shutdown. Upon reboot, if you are using a boot manager to load various operating systems, you can switch to another operating system or log in to Linux.
-h	Halt the system after shutdown. At this point, you can safely turn off the power.
-n	Shutdown the system without invoking init. It is recommended not to use this option.
-f	Skip the filesystem check on reboot.
-F	Force the filesystem check on reboot.
-c	Cancel a shutdown in progress. This option does not use the time parameter, but can use the warning message option.

Alternative Shutdown Commands

You may also shutdown the system by directly changing the runlevel, which may be done via the `telinit 1` or `systemctl isolate poweroff.target` commands.

The `/etc/nologin` File

When the shutdown program is called with a delay, it will create an empty `/etc/nologin` file, which will restrict further connections. If you manually add this file, it will allow only the superuser or root user to log in and restrict all others. The contents in the `/etc/nologin` file will be displayed on every failed login attempt.

How to Start and Stop Linux

Follow these general procedures to start and stop Linux.

Manage Runlevels from a Shell

To manage runlevels from a shell:

1. At the command prompt, bring up your current runlevel.
2. Switch to runlevel 1.
3. Verify that you are at runlevel 1.
4. Exit back to runlevel 3.

Manage Runlevels via Systemd

To manage runlevels from a configuration file:

1. View the current default target (runlevel) via `systemctl get-default`.
2. Enter `systemctl set-default rescue.target` to start at runlevel 1.
3. Restart the computer.
4. Verify that you are in runlevel 1.
5. Enter `systemctl set-default graphical.target` to start at runlevel 5.
6. Restart the computer.

Alert Users Before Switching Runlevels

To alert all users connected to your system before switching runlevels:

1. Log in to the CLI as **root**.
2. Enter `shutdown -t [seconds] now [message]` or enter `shutdown -k now`.

Scenario

Answer the following review questions.

1. What are the advantages of open source software over licensed software?
2. What are the advantages of using Linux?

Summary

In this lesson, you identified basic Linux concepts and performed basic Linux tasks. These skills can assist you in supporting Linux users and machines.

2 Managing User and Group Accounts

Lesson Time: 1 hour, 30 minutes

Lesson Introduction

You are now familiar with the history of Linux®, its shells, and its help and support options.

This basic knowledge is a good starting point, but there is more to learn. Before users can take advantage of the operating system, user accounts need to be created. You will also have to create group accounts to manage those users. In this lesson, you will manage user and group accounts.

One of the benefits of Linux is its multiuser capabilities. By creating and modifying user and group accounts, you can further tailor the Linux environment to the needs of your organization. You will also be able to provide individualized services to users after creating an account for them.

Lesson Objectives

In this lesson, you will manage user and group accounts. You will:

- Create user and group accounts.
- Configure user profiles.
- Modify user and group accounts.

TOPIC A Create User and Group Accounts

In this lesson, you will manage user and group accounts. The first step in managing them is to create the accounts you need. In this topic, you will create user and group accounts.

As a Linux administrator, you will be required to create user and group accounts on a regular basis.

By creating user accounts, you will enable users to access the Linux system. Group accounts enable you to group users with similar functions. This will considerably reduce the time and effort you invest in monitoring and managing user activities.

User Accounts

A **user account** is a collection of information that defines a user on a system. It is the representation of a user on a computer. User account information includes the user name and password for the user to log in to the system, groups to which the user belongs, and rights and permissions that the user has to access the system and its resources. When an account is created, it is assigned a unique number that is called **User ID (UID)**.

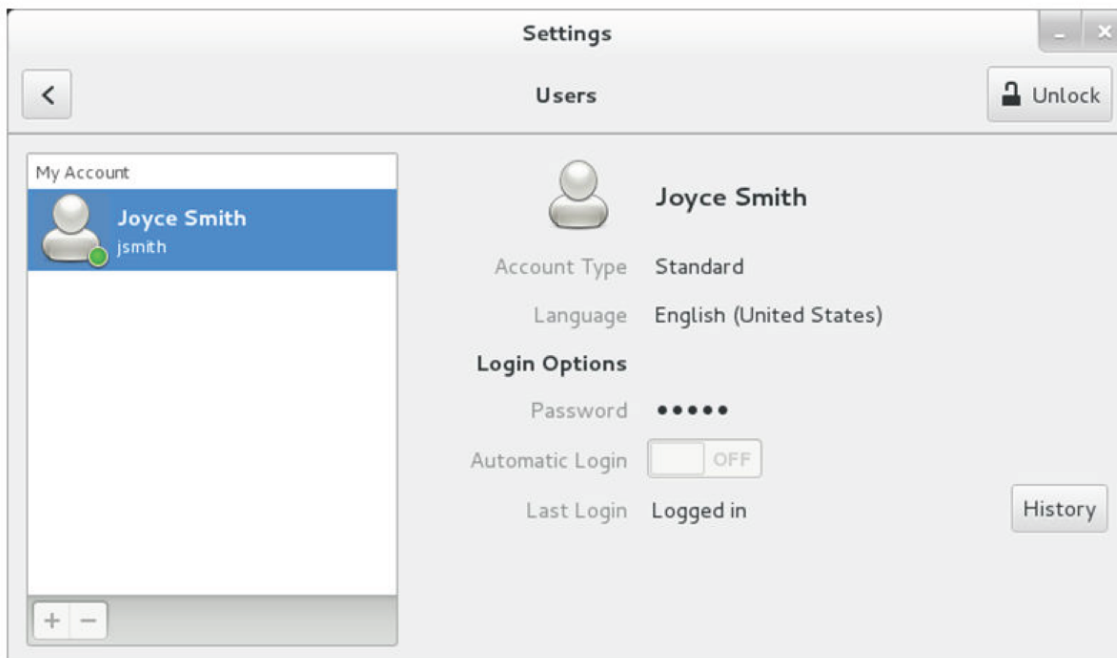



Figure 2-1: The format in which the user account information is stored by a system.

The useradd Command

The `useradd` command is used to add a new user. You need to specify the user name along with the command to create a new user account. Special user accounts are required to run processes associated with certain services. For example, `daemon` is a user account that is used to run the `daemon` service.



Figure 2-2: A new user added using the `useradd` command.



Note: You can use the `adduser` command to perform the same functions as the `useradd` command on CentOS/RHEL systems.

Syntax

The syntax of the `useradd` command is `useradd [options]{username}`.

Special User Accounts

In special user accounts, the UID value for the users will be less than the default UID value, which is 500. Such special users will not have a home directory. You can create a special user account using

the `useradd -r {special user name}` command.

User Accounts

Linux allows you to add user accounts by directly editing the password file. However, this is not recommended because you may damage your system if you accidentally leave something out or alter existing user accounts. If the system is damaged, nobody will be able to log in—not even the root user. In such a case, you will have to reinstall your system and redefine the user accounts.

Default User Accounts

Numerous user accounts are created by default upon system installation. Some of the main user accounts include the following:

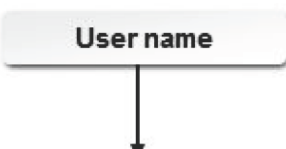
- root
- bin
- daemon
- ftp
- sshd
- nfsnobody
- apache
- rpc
- gnome

The Role of the Root User

Every Linux system has at least one system administrator whose job is to maintain the system and make it available to users. This user is the root user. The root user can perform any task on the Linux system without restrictions. System administrators are also responsible for adding new users to the system and for setting up their initial environment.

Passwords

A password is an entity that allows the Linux system to authenticate a user. Generally, when user accounts are created without passwords, they can be easily misused. For this reason, when you create a user account, you should immediately set a password for the user using the `passwd` command. In Linux, if a password is not set for the user account, the account gets locked automatically. This is to help prevent unauthorized access to the system.



```
[root@localhost ~]# passwd pat
Changing password for user pat.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# █
```

Figure 2-3: Setting a password for the user pat.



Note: You can change the password of your user account using the `passwd` command. You cannot change the password for any other user account because the `passwd` command does not allow you to specify any other user name. Only the root user can change the password for other users by specifying the user name with the `passwd` command.

Syntax

A root user can create a password for a user by entering `passwd [user name]`, where `[user name]` is the name of the user for whom the password is set.

Dictionary Words as Passwords

If you enter a password that is a real word made up solely of alphabetic characters, you will get a bad password message stating that it is based on a dictionary word. The password will still be assigned even though the message is displayed. It is strongly advised that you change it to a more secure password.

The `/etc/passwd` File

When you add a new user, information about the user is saved in the `/etc/passwd` file.

There are various fields in the `/etc/passwd` file.

Field	Description
User name	Stores the user name with which the user logs in to the system. It is recommended to limit user names to eight alphanumeric characters.
Password	Stores the password that is assigned to the user in an encrypted form.
User ID	Stores the unique number that is assigned to each user. Linux tracks users by the UID rather than the user name.
Group ID	Stores the unique number that is assigned to each group. Users can be members of one or more groups.
Full name	Stores the real name of the user.
Home directory	Displays the default directory where the user is placed after logging in.
Login shell	Displays the default shell that is started when the user logs in.

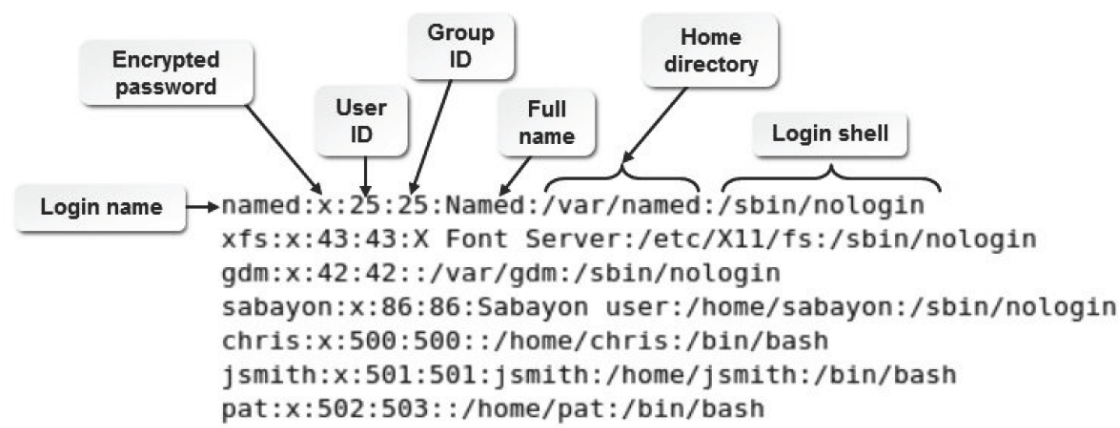


Figure 2-4: The contents of the `/etc/passwd` file.

Shadow Passwords

In earlier UNIX and UNIX-like systems including Linux, each user's password is stored and encrypted in the `/etc/passwd` file. This file needs to be readable, which makes copies of users' encrypted passwords easily obtainable to any person trying to attack the system. Then, by using various techniques, the attackers can decipher passwords. Modern distributions have overcome this problem by using shadow passwords by default. Shadow passwords store the encrypted passwords in a separate highly protected file, the `/etc/shadow` file. This file is readable only to the root user.

Therefore, it is less of a security risk compared to the `/etc/passwd` file because it becomes difficult for attackers to access the file, obtain the user passwords, and then decipher them. The `/etc/passwd` file also contains the account or password expiration values.

The `/etc/shadow` File

The `/etc/shadow` file contains the following information:

- `username`: The user name.
- `passwd`: The encoded password.
- `last`: Number of days since the password was last changed.
- `may`: Number of days before which the password may be changed.
- `must`: Number of days after which the password must be changed.
- `warn`: Number of days pending before which the password will expire.
- `expire`: Number of days after which the password will expire and the user account will be disabled.
- `disable`: Number of days since Jan 1, 1970, that the user account has been disabled.
- `reserved`: A reserved field.

The `id` Command

The `id` command is used to display UID and group ID (GID) information. Entering the command with no options displays information about the user who is currently logged in. You can also specify a user name as an option to display ID information about a specific user.

The `finger` Command

The `finger` command is used to display information about users, including login name, real name, terminal name, write status, idle time, login time, office location, and office phone number. Some of these fields may be empty if no information was included when the user account was created. You can also view information about a specific user by entering `finger [user name]`.

Groups

A **group** is a collection of system users having the same access rights. Every user must be a member of a group. Users can also be members of more than one group. Group membership is used to limit access to files and system resources. The `groupadd` command allows you to add a group.

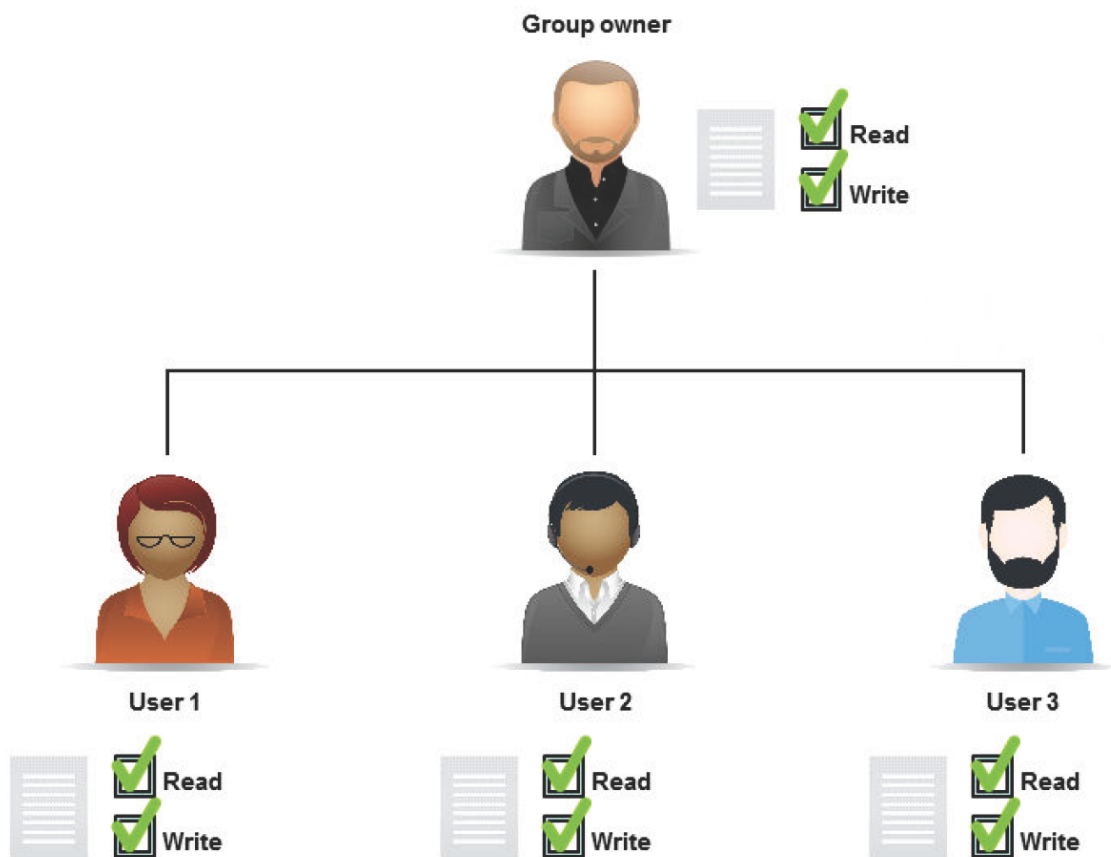


Figure 2-5: A group of users with their permissions.

Syntax

The syntax of the groupadd command is `groupadd {group name}`.

Standard Groups

The table lists the standard groups set up by the installation process.

Group	GID	Default Member
root	0	root
bin	1	root, bin, and daemon
daemon	2	root, bin, and daemon
sys	3	root, bin, and adm
adm	4	root, adm, and daemon
tty	5	None
disk	6	root
lp	7	daemon and lp
mem	8	None
kmem	9	None
wheel	10	root
mail	12	mail
man	15	None
games	20	None
gopher	30	None


Group	GID	Default Member
dip	40	None
ftp	50	None
nobody	99	None
users	100	None

User Private Groups

A **User Private Group (UPG)** is a unique group that is created by default whenever a new user account is created. This is the primary group of the new user account. Only the new user is a member of this group.

The /etc/group File

The **/etc/group** file contains a list of groups, each on a separate line. Each line consists of four fields for attribute definition, separated by colons. The **/etc/group** file is also called the **group database**.

	Note: The /etc/gpasswd file stores the encrypted passwords for groups.
---	--

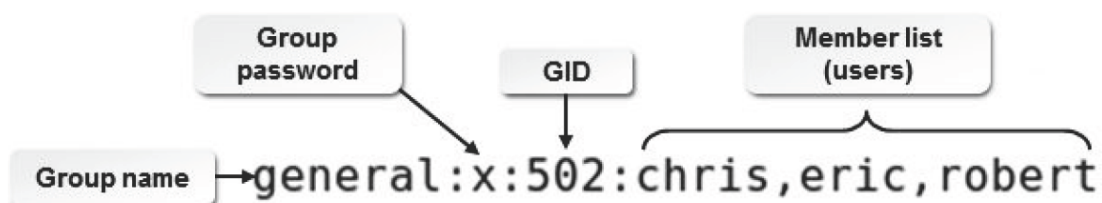


Figure 2-6: An example of an entry in the /etc/group file.

The following table lists the different fields and their usages.

Field	Description
Group name	Stores the name of the group.
Group password	Stores the password of the group in an encrypted form.
GID	Stores the group identifier; similar to a UID for groups. The default GID value is 500.
Members	Stores the names of the members of the group separated by commas.

How to Create User and Group Accounts

Follow these general procedures to create user and group accounts.

Create a User Account

To create a user account:

1. Log in as **root**.
2. To add a new user, at the command prompt, enter `useradd {user name}`.
3. To set a password for the user, enter `passwd {user name}`.
4. Confirm the password.
5. If desired, to log in to the system, use the newly created user name and password.



Note: While creating a user with the `useradd` command, a user private group with the same name is also created.

Create a Group Account

To create a group account:

1. Log in as **root**.
2. To add a new group, at the command prompt, enter `groupadd {group name}`.
3. Verify that the group was created by viewing the `/etc/group` file.

TOPIC B Configure User Profiles

Now that you can create user and group accounts, the next step is to configure user profiles. Each user connected to a system requires a distinct identity to differentiate one user from another. In this topic, you will configure user profiles.

All users connected to the system require customized settings for their systems. Further, there may be files to be shared by default. By configuring user profiles, every user can be given a distinct identity. This differentiates one user from another.

User Profiles

A **user profile** is a set of options, preferences, bookmarks, and other user items that characterize a user. User profiles define settings such as network resources, data, attributes, and permissions that the system assigns to a user. These settings are retained for every session. The user can specify a name for the user profile. Otherwise, the profile will be called "Default User." Each user can create several user profiles for business or personal use.



Figure 2-7: The list of values set for the root user profile.

Modifying Default Options

You can modify default options while configuring a user profile. Some commonly modified options include:

- **PS1:** This variable stores information about the primary prompt, which is the prompt that is displayed when users log in. This variable may or may not be modified.

- PS2: This variable stores information about the secondary prompt.
- PATH: This variable stores information about the search paths for commands. You can modify the PATH if you want to use commands that are not stored in the standard directories.

Hidden Files and Directories

Some files and directories in the system are hidden. The `ls` command lists all files, except hidden files. To display all files, including hidden ones, the `ls -a` command is used. The names of hidden files and directories start with a period. You can also add a period to the names of directories to hide them. Hidden files are usually those files that require minimal editing.

The Profile File

When a user logs in and starts a new Bash session, several commands need to be typed to customize the user's session. It will be tedious to type these commands every time the user logs in. Therefore, these commands are saved in a special executable file from which Bash will run the commands every time the user logs in. This file is called a **profile file** because it contains the commands that are used to tailor the session according to the requirements of the user. Individual profiles for every user are available at the `~/.bash_profile` file in the user's home directory, and changes to this file affect the user's customized settings.

Global User Profiles

A **global user profile** is a set of options, preferences, bookmarks, stored messages, attributes, permissions, and other user items that users have access to, on whichever system they log in to.

Global user profiles are stored on the server. Each time a user logs in, data in the global profile is copied to the local system. While the user is logged in, any changes made to the settings affect only the local copy of the profile.

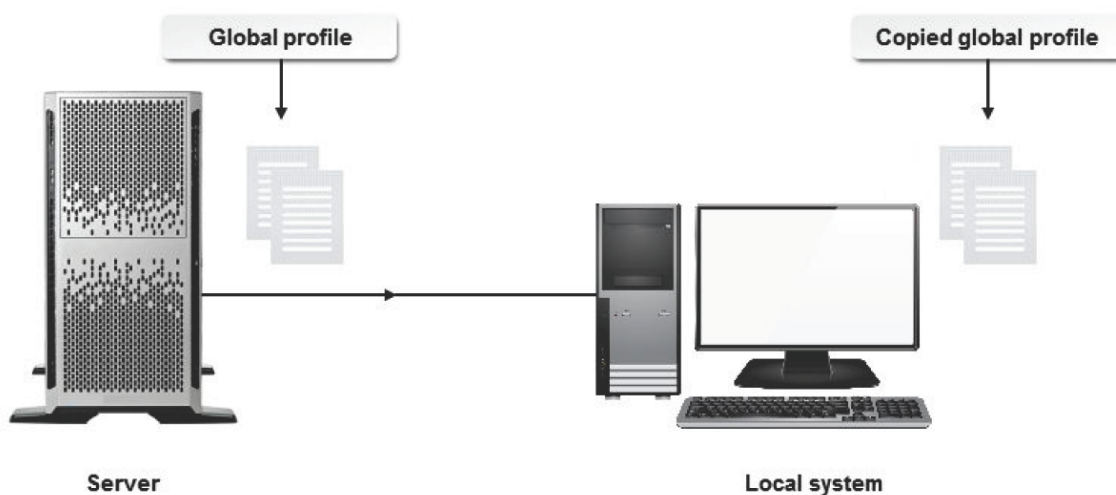


Figure 2-8: A global user profile allows the user to access any system connected to the server in which the profile is saved.

The Skeleton Directory

When a new user account is created, the **skel directory** stores a copy of the files and directories that are placed in the home directory of the new user. The **skel** directory path is `/etc/skel`. This ensures that all new users begin with the same settings. Modifications made to the **skel** directory affect only the new users.

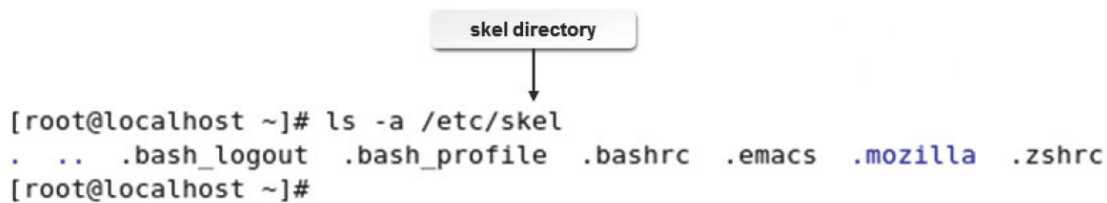


Figure 2-9: Default files in the skel directory.



Note: Skel is derived from the word "skeleton," which implies a basic folder structure.

The Gedit Editor

Gedit is a simple yet powerful GUI-based text editor used in the GNOME desktop. In the GUI environment, you can use the `gedit {file name}` command to open a specific file. Alternatively, you can select

Applications → **Accessories** → **gedit** to launch the gedit application and use the GUI components to open a specific file.

The which Command

The `which` command is used to verify whether a user has the right to execute a command. The `which` command displays the complete path of the command by searching in the `PATH` variable.

For example, on entering `which cat`, the output `/bin/cat` is displayed.

This means that the `cat` command is located in the **/bin** directory. If `/bin` is not located in the `PATH` variable, an error message saying "no cat in [search path]" is displayed.

Managing the /etc/skel Files

By default, the hidden files for configuring a user's environment are stored in the **skel** directory.

These include `.bash_profile`, `.bashrc`, `.bash_logout`, and others. If there are other files that you would like to include in new user accounts, you can add those files to this directory. The files will then be copied to the new users' home directories when new users are created.

The unset Command

The `unset` command is used to remove the set variables temporarily.

How to Configure User Profiles

Follow these general procedures to configure user profiles.

Maintain skel Directories for New User Accounts

To maintain *skel* directories for new user accounts:

1. Log in as **root**.
2. Enter `cd /etc/skel`.
3. Enter `vi .bash_profile`.
4. Make the necessary changes, such as changing `PS1`, `PS2`, or the `PATH` variable for a new user account.

5. Save and close the file.

Delegate Files to New Users

To delegate a file to a newly created user by default:

1. Log in as **root**.
2. Enter `cd /etc/skel`.
3. To move the file as a hidden file, enter `mv /[location of the file]/[file name] [file name]`.
4. Verify that the files have moved as hidden files.
 - a. Create a user.
 - b. Log in as the new user.
 - c. To view the file that was placed in the **/etc/skel** directory, enter `cat .[file name]`.

Change a User's Profile

To change a user's profile:

1. Log in as a user.
2. Enter `vi .bash_profile`.
3. Make the necessary changes to a variable and export it. For example, to change the number of commands to be stored in the history variable, specify the desired size in the HISTSIZE variable as `HISTSIZE={desired value}` and type `export HISTSIZE` to export the variable.
4. Save and close the file.

Set the Command Search Path with the Proper Directory

To set the command search path with the proper directory so as to execute the command from anywhere:

1. Log in as a user.
2. Enter `mkdir {directory name}`.
3. Create or add an executable file in the directory.
4. If necessary, to observe the output of the executable file, enter `./{executable file name}`.
5. Enter `vi .bash_profile`.
6. In the PATH variable, type `:$HOME/{directory name}`. For example, if the PATH variable is given as `PATH=$PATH:$HOME/bin`, then after adding the command search path, the PATH variable will look like `PATH=$PATH:$HOME/bin:$HOME/{directory name}`.
7. Save and close the file.
8. Log out and log in as the same user.

9. To execute the file, enter {executable file name}.

Manage env Variables Globally

To manage an env variable globally:

1. Log in as **root**.
2. If desired, to observe the settings, enter `env | less`.
3. Enter `vi /etc/profile`.
4. To define a variable, enter `{VARIABLE NAME}={value}`.
5. If necessary, you can modify the values of the variables. For example, to modify the HISTSIZE profile globally for all users, navigate to the HISTSIZE variable and make the necessary changes.
6. To export the variable, type `export {VARIABLE NAME}`.
7. Save and close the file.
8. Log out and log in as **root**.
9. If necessary, to observe the changes made to the settings, enter `env | less`.

Manage the set Variable

To manage the set variable:

1. Log in as **root**.
2. If necessary, to observe the set variables, enter `set | less`.
3. If necessary, to verify the output, enter `echo $Set variable {VARIABLE NAME}`.
4. Enter `vi .bash_profile`.
5. To define a variable, enter `Set variable {VARIABLE NAME} = {value}`.
6. To export the variable, type `export {VARIABLE NAME}`.
7. Save and close the file.
8. Log out and log in as **root**.
9. If desired, to verify the changes, enter `set | less`.

TOPIC C Administer User and Group Accounts

In the last topic, you created user and group accounts and even configured user profiles. Your next step will be to manage user and group accounts on an ongoing basis. This will enable you to efficiently organize your Linux environment. In this topic, you will manage user and group accounts.

Once a user or group account is created, there are many tasks that need to be performed to maintain that account. As a system administrator, you will be required to maintain the accounts and passwords of numerous users. This is achieved by effective management of user and group accounts.

The userdel Command

The userdel command allows you to modify the system account files, deleting all entries that refer to the login of an existing user. However, it will not allow you to remove an account if the user is currently logged in. You must kill any running processes that belong to an account before deleting the account.

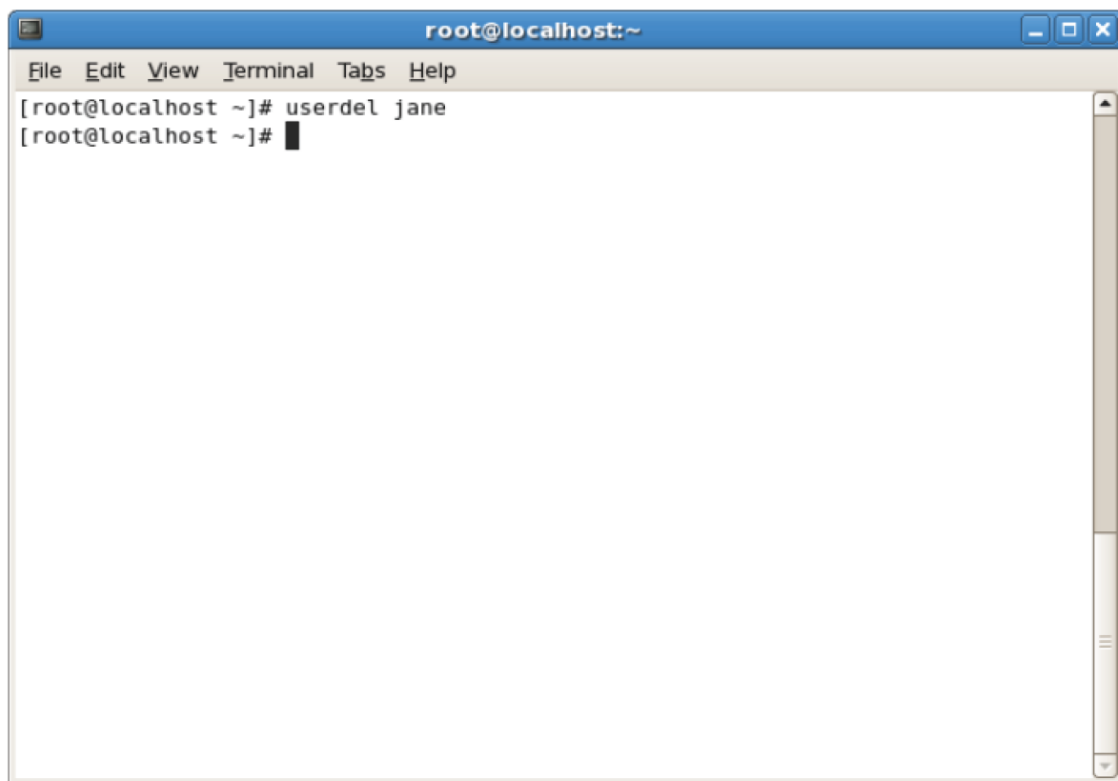


Figure 2-10: The userdel command is used to delete an unused user account.

Syntax

The syntax of the userdel command is `userdel [options] {username}`.

The -r Option

The -r option will delete the files in the user's home directory, along with the home directory itself.

Files owned by this user and located in other locations will have to be searched for and deleted manually.

The usermod Command

The usermod command has options that enable you to modify various user account parameters.

You can change a user's name, default groups, UID, or passwords.



Figure 2-11: The `usermod` command is used to add more information about the user.

Some of the common `usermod` command options and their descriptions are given in the following table.

Option	Allows You To
<code>usermod -l {new login}{login}</code>	Modify the login name of the user.
<code>usermod -c "comment" login</code>	Modify the user's full name, office address, and contact numbers in the password file. Alternatively, you can use the <code>chfn {user name}</code> command to modify the details.
<code>usermod -f {number of days} {login}</code>	Modify the number of days for a password to expire and to disable the account permanently.
<code>usermod -u {new unique user ID} {login}</code>	Modify the numerical value of a user's ID, which has to be unique.
<code>usermod -d {new login directory} login</code>	Modify the user's default login directory.
<code>usermod -L {user name}</code>	Lock the password and suspend the user account temporarily.
<code>usermod -U {user name}</code>	Unlock the password.
<code>usermod -e {yyyy-mm-dd} {user name}</code>	Change the expiration date for the user account.

Syntax

The syntax of the `usermod` command is `usermod [options] {username}`.

The `mkdir` Command

The `mkdir` command allows you to create new directories. The syntax of the command is `mkdir {directory name}`.

The `chown` Command

The chown command is used to change the user or group that owns one or more files or directories.

Lock User Login

In Linux, you can lock a user's login to temporarily prevent a user from logging in to a system. This is done by disabling the user's password using the `passwd -l` or `usermod -L` command. The user's login is usually locked as a security measure, to prevent unauthorized usage when the user is unavailable.

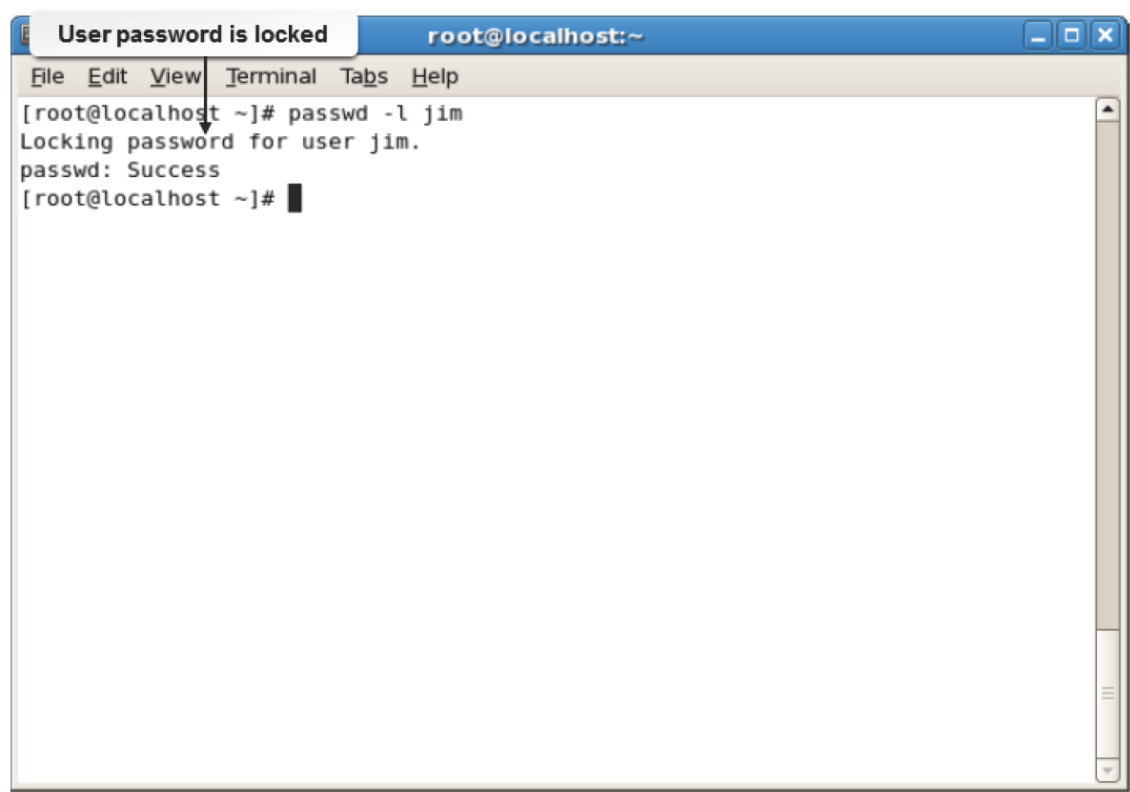


Figure 2-12: The `passwd -l` command is used to lock the user's password.

Group Management

Groups, like users, are identified by a system with a unique number known as GID. In Linux, users can be members of one primary group and multiple supplemental groups. The `groupdel` and `groupmod` commands are useful in managing groups.

Command	Allows You To
groupdel	Delete a group from the system.
groupmod	Change the group's name and the numerical value of the group's ID by modifying the system account files.

Syntax

The syntax of the `groupdel` command is `groupdel {group name}`.

The syntax of the `groupmod` command is `groupmod -g{GID}`.

Group Account with GID

To add a new group to the system with a name of `print_users` and a GID of 700, enter `groupadd -g 700 print_users` at the command line.

Adding Users to a Group

As with users, the group file can be directly edited to add groups. You can use the `groupadd` command to add users instead of editing the group file.

How to Manage User and Group Accounts

Follow these general procedures to manage user and group accounts.

Change a User's Home Directory

To change the home directory of a user:

1. Log in to the CLI as **root**.
2. To create a directory, enter `mkdir /{name of the directory}`.
3. To create a directory for the specified user, enter `mkdir /{name of the directory}/{user name}`.
4. To change the ownership of the directory, enter `chown {user name}[:{group name}] /{name of the directory}`.
5. To set the user's new home directory, enter `usermod -d /{name of the directory}/{user name} {user name}`.

Modify User Settings in the CLI

To modify user settings in the CLI:

1. Log in as **root**.
2. To modify user settings, enter `usermod [options] {user name}`.

Modify User Settings in the GUI

To modify user settings in the GUI:

1. Log in as **root** in the GUI.
2. To open the User Manager window, select **System → Administration → Users and Groups**.
3. Modify user settings as desired.
 - To add a new user, select **Add User**.
 - To modify user settings, double-click a user name.
 - To remove a user account, select the user and select **Delete**.

Remove User Accounts

To remove user accounts:

1. Log in as **root**.
2. To delete user accounts, enter `userdel [options] {user name}`.

Manage Default Password Aging Information

To manage default password aging information:

1. Log in as **root**.
2. To open the **/etc/login.defs** file, enter `gedit /etc/login.defs`.
3. Manage the password aging information.
 - To control the maximum number of days a password may be used, modify the value next to the **PASS_MAX_DAYS** variable.
 - To control the minimum number of days allowed between password changes, modify the value next to the **PASS_MIN_DAYS** variable.
 - To control the minimum password length, modify the value next to the **PASS_MIN_LEN** variable.
 - To control the number of days to issue warnings before a password expires, modify the value next to the **PASS_WARN_AGE** variable.
4. Save and close the file.

Set or Change Password Aging Information

To set or change password aging information:

1. Log in as **root**.
2. To change the password aging information of the specified user, enter `change [options] {user name}`.

Modify or Delete Groups

To modify or delete groups:

1. Log in as **root**.
2. Manage the groups.
 - To change the GID, use the `groupmod -g {GID} {group name}` command.
 - To delete a group, use the `groupdel {group name}` command.

ACTIVITY 2-1

Managing User and Group Accounts Review

Scenario

Answer the following review questions.

1. How is organizing users into groups useful to you?
2. Why is it essential to configure a user profile?

Summary

In this lesson, you created and managed user and group accounts. This will help you efficiently organize and maintain a Linux environment with numerous users.

3 Managing Partitions and the Linux Filesystem

Lesson Time: 2 hours, 30 minutes

Lesson Introduction

You are now familiar with user and group accounts in Linux®. Further, you need to manage the Linux filesystem. In this lesson, you will create partitions on the hard disk and navigate, manage, and maintain the Linux filesystem.

Data organization facilitates efficient resource management and faster retrieval of information. Data organization is done by sorting data into filesystems. The Linux filesystem is part of what sets Linux apart from other operating systems. Understanding the structure and workings of the filesystem will assist you in storage, management, and troubleshooting of data.

Lesson Objectives

In this lesson, you will manage partitions and the Linux filesystem. You will:

- Create partitions.
- Navigate through the Linux filesystem.
- Manage the Linux filesystem.
- Maintain the Linux filesystem.

TOPIC A Create Partitions

Before you work with the Linux filesystems, you should partition the hard disk of your system.

Proper partitioning of the hard disk will ensure that users have enough space to store their data. In this topic, you will create and manage disk partitions.

The hard disk is the most critical component for data storage in any system. Without effective disk partitioning, data on the disk will be unorganized and cluttered, or the users might run out of available storage space prematurely. Improper disk partitioning may also contribute to a system crash. As a Linux administrator, it is your responsibility to ensure that disks are partitioned properly such that users have enough space to store their data in an efficient manner.

Filesystems

A **filesystem** is a method that is used by an operating system to store, retrieve, organize, and manage files and directories on mass storage devices. A filesystem maintains information, such as the date of creation and modification of individual files, their file size, file type, and permissions. It also provides a structured form for data storage. A filesystem by itself does not interpret the data contained in files because this task is handled by specific applications. Filesystems vary depending on several parameters, such as the purpose of the filesystems, the information they store about individual files, the way they store data, and data security.

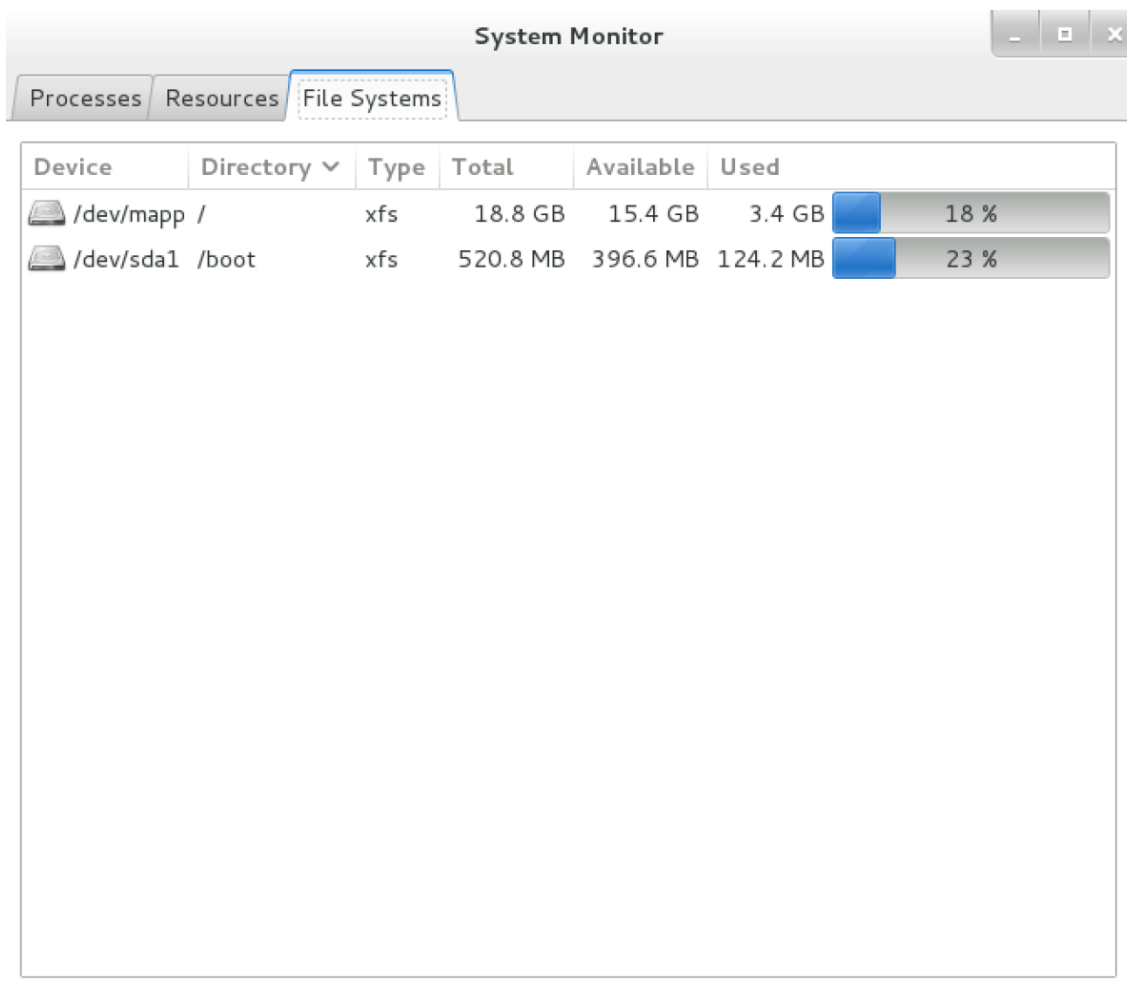


Figure 3-1: Files are stored in directories.

Filesystem Labels

Filesystem labels are assigned to filesystems for easy identification. The labels may be up to 16 characters long and can be displayed or changed using the `e2label` command.

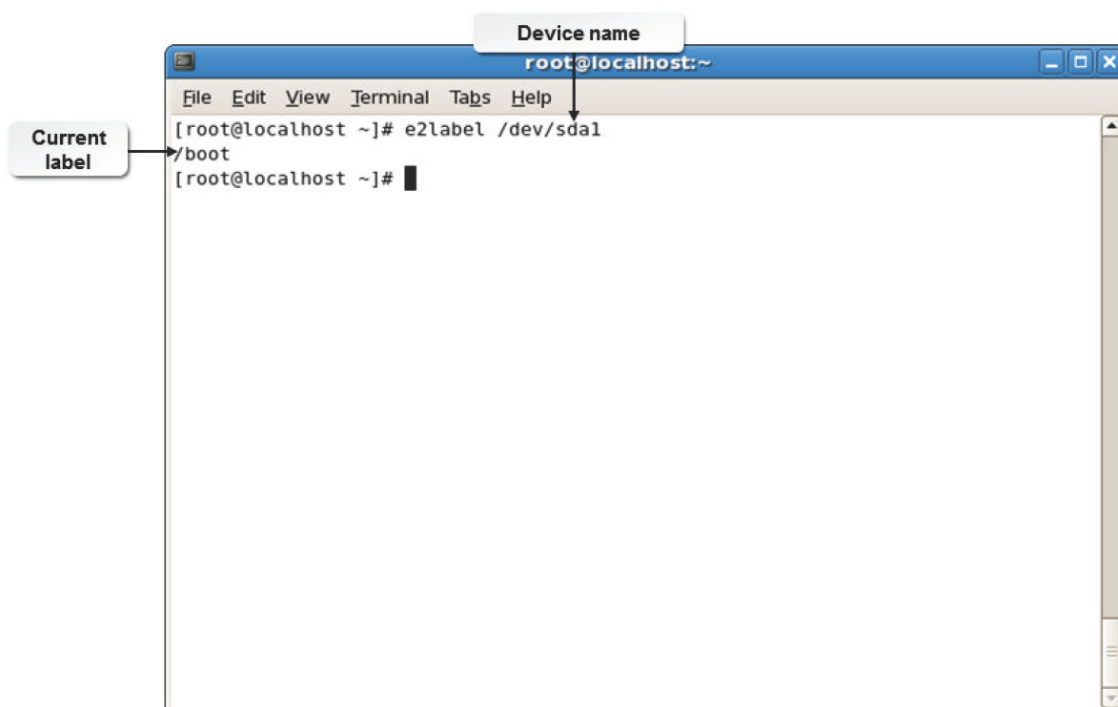


Figure 3-2: An example of a filesystem label.

Syntax

The syntax for setting filesystem labels is `e2label /dev/{device name}{partition number} {label name}`. They can also be set using the `tune2fs -L {volume label} {device}`

command.

Filesystem Types

Linux supports many common filesystem types. Some are described in the following table.

<i>Filesystem Type</i>	<i>Description</i>
ext2	This used to be the native Linux filesystem of some of the previous releases. It is still supported in the current releases of Linux.
ext3	This is an improved version of ext2. In case of an abrupt system shutdown, ext3 is much faster in recovering data and better ensures data integrity. You can easily upgrade your filesystem from ext2 to ext3.
ext4	The newest default filesystem for Linux distributions. It is backwards- compatible with the ext2 and ext3 filesystems. Among ext4's improvements over ext3 are journaling, support of volumes of up to one exbibyte (EiB), and files up to 16 TiB in size. Ext4 is the default filesystem for CentOS/RHEL 7 and Ubuntu installations.
XFS	This is a 64-bit, high-performance journaling filesystem that provides fast recovery and can handle large files efficiently. XFS is the default filesystem for CentOS/RHEL 7 installations.
ReiserFS	This can handle small files efficiently. It handles files smaller than 1K and is faster than ext2 and ext3. If appropriately configured, it can store more data than ext2.
vfat	This is a 32-bit filesystem and supports long file names. It is compatible with the FAT filesystem of Microsoft Windows XP and Microsoft Windows NT.
JFS	This is a 64-bit journaling filesystem that is fast and reliable. It is better equipped to handle power failures and system crashes.
swap	This is not a true filesystem, but rather is a portion of the hard disk that is used in situations when Linux runs out of physical memory and needs more of it. Linux pushes some of the unused files from RAM to "swap" to free up memory.
ISO 9660	This is a filesystem standard defined by the International Organization for Standardization (ISO), and is also called a CDFS (Compact Disc File System). Linux allows you to access DVDs and CDs that use this filesystem.

Access to Other Filesystems

Linux allows you to access other filesystems and mount them when required. However, you cannot install Linux on these filesystems.

<i>Filesystem</i>	<i>Description</i>
FAT	The FAT (File Allocation Table) filesystem is compatible with different operating systems, including all versions of Windows, MS-DOS, and UNIX. It is primarily used for formatting floppy disks.
NTFS	NTFS (New Technology File System) is the recommended filesystem for Windows-based computers. NTFS provides many enhanced features over FAT or vfat, including file- and folder-level security, file encryption, disk compression, and scalability to very large drives and files.

Partitions

A **partition** is a section of the hard disk that logically acts as a separate disk. Partitions enable you to convert a large hard disk to smaller manageable chunks, leading to better organization of information. A partition must be formatted and assigned a filesystem before data can be stored on it. Partitions are identified using a partition table, which is stored in the boot record. The partition table can contain entries for a maximum of four primary partitions. Partitions can be classified into primary and extended partitions. The size of each partition can vary but cannot exceed the total free space of the hard disk.

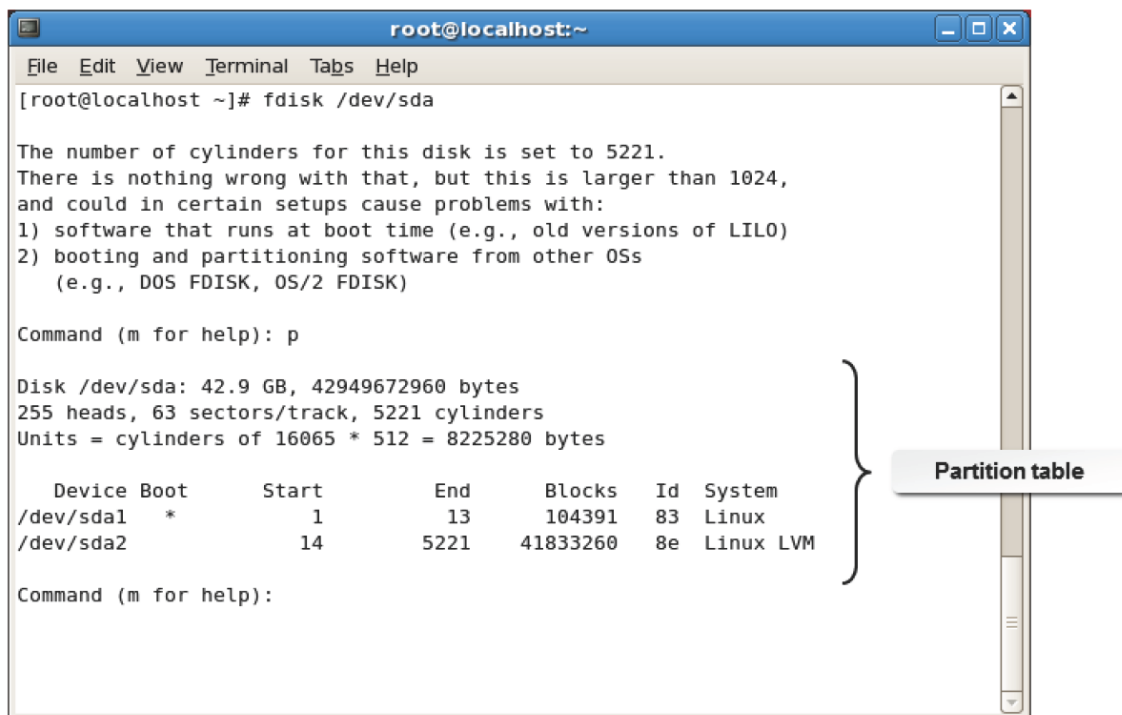


Figure 3-3: Partitions on a hard disk.

Hard Disk Size Specification

Before proceeding with the installation process, you need to plan the hard disk layout based on your requirements. Each partition has a recommended size specification. The following table lists the recommended size specification for partitions.

Partition	Recommended Size
/	Minimum 1 GB.
/boot	100 MB.
swap	Double the RAM size.
/var	Minimum 250 MB. If the possibility of the installation of many applications exists in the future, allocate the appropriate size.
/home	Varies based on the number of users.

Disk Partitioning

Most operating systems, including Linux, use disk partitions. Data of different types can be stored in separate locations on the hard disk. The partition size can be specified by a user. However, the filesystem size must be considered before specifying the partition size. Disk partitioning enables the user to separate system files from user accessible ones. Corrupted partitions do not affect the other partitions, and they can be recovered separately.

Partition Types

There are three types of partitions: primary, extended, and logical. The functionality of the hard disk depends on the types of partitions on it.

Each partition has a set of specific features. The three types of partitions are described in the table.

Partition Type	Description
-----------------------	--------------------

Partition Type	Description
Primary	A disk partition that can contain one filesystem or logical drive and is sometimes referred to as volumes. A maximum of four primary partitions are allowed. The swap filesystem and the boot partition are normally created in a primary partition.
Extended	An extended partition can contain several filesystems, which are referred to as logical disks or logical drives. There can be only one extended partition, which can be further subdivided. This partition type does not contain any data and has a separate partition table.
Logical	A part of a physical disk drive that has been partitioned and allocated as an independent unit and functions as a separate drive. A logical partition is created within an extended partition. There is no restriction on the number of logical partitions, but it is advisable to limit it to 12 logical partitions per disk drive.

The fdisk Utility

fdisk is a menu-driven utility program that is used for creating, modifying, or deleting partitions on a disk drive. Using fdisk, a new partition table can be created, or existing entries in the partition table can be modified. The fdisk utility understands the DOS and Linux type partition tables.

Depending on the partition table created, the DOS FDISK or the Linux fdisk program is invoked.

The fdisk utility also allows you to specify the size of partitions.

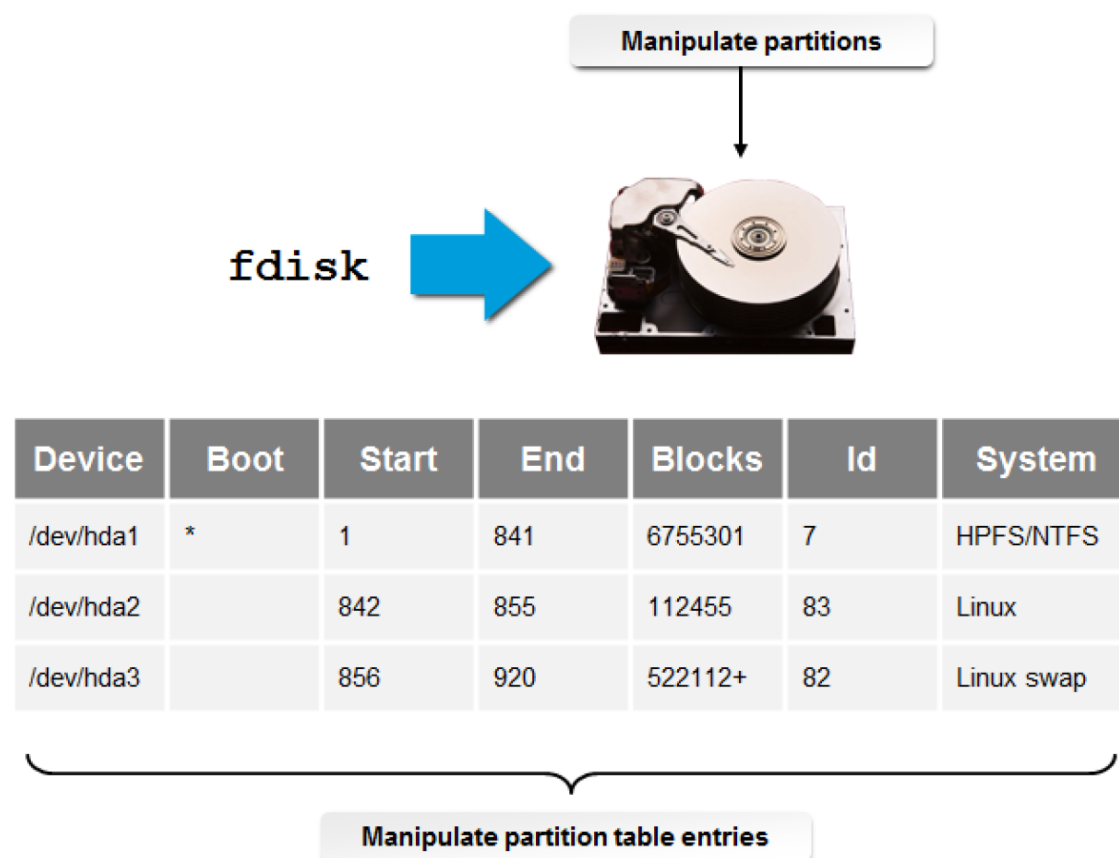


Figure 3-4: A partition table created with the fdisk utility.

Syntax

The syntax of the fdisk utility is `fdisk [options] {device name}`.

Command Line Options Supported by fdisk

The fdisk utility supports a number of command line options.

Option	Enables You To
-b <i>sector size</i>	Specify the number of disk sectors.
-H <i>heads</i>	Specify the number of disk heads.
-S <i>sectors</i>	Specify the number of sectors per track.
-s <i>partition</i>	Print the partition size in blocks.
-v	List the fdisk version.
-l	List partition tables for devices.

fdisk Utility Options

The fdisk utility provides various options for partitioning disks according to the requirements of users. Some of the fdisk options are described in the following table.

Option	Enables You To
n	Create a new partition. The sub-options allow you specify the partition type and partition size.
d	Remove a partition.
p	List the existing partitions.
w	Write the changes to the disk and exit the utility.
q	Cancel the changes made and exit the utility.

The fstab File

The **fstab** file is a configuration file that stores information about storage devices and partitions and where and how the partitions should be mounted. The **fstab** file is located in the **/etc** directory. It can be edited only by a root user. The **fstab** file consists of a number of lines—one for each filesystem.

Each line in an fstab file has six fields that are separated by spaces.

Field	Description
Device or partition name	Specifies the name of the device or filesystem that has to be mounted.
Default mount point	Indicates where the filesystem has to be mounted.
Filesystem type	Specifies the type of filesystem used by the device or partition.
Mount options	Specifies a set of comma-separated options that will be activated when the filesystem is mounted.
Dump options	Indicates if the dump utility should back up the filesystem. Usually, zero is specified as the dump option to indicate that dump can ignore the filesystem.
fsck options	Specifies the order in which the fsck utility should check filesystems.

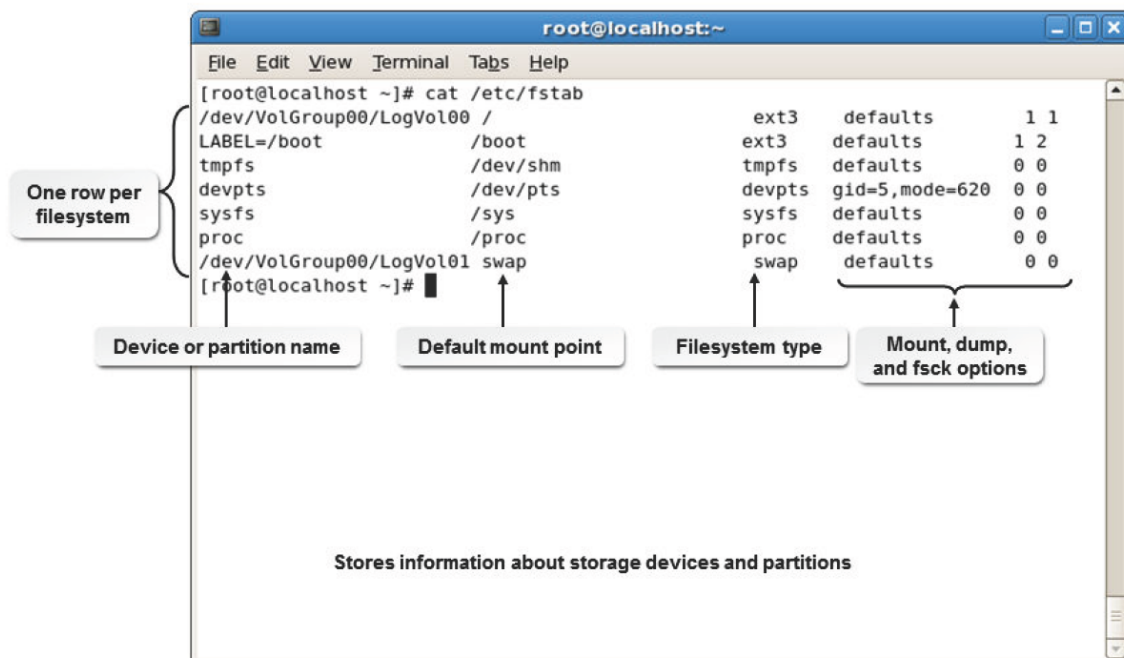


Figure 3-5: The `/etc/fstab` file contains partition and filesystem settings.

The `mkfs` Command

The `mkfs` command is used to build a Linux filesystem on a device, which is usually a hard disk partition. The following table lists some options of the `mkfs` command and their description.

Option	Allows You To
<code>-v</code>	Produce verbose output, where the output message will keep changing constantly as the program is processing.
<code>-V</code>	Produce verbose output, including all filesystem-specific commands that are executed.
<code>-t {fstype}</code>	Specify the type of filesystem to be built.
<code>fs-options</code>	Pass filesystem-specific options to the filesystem builder.
<code>-c</code>	Check the device for bad blocks before building the filesystem.
<code>-l {file name}</code>	Read the list of bad blocks from a specified file.

Building New Linux Filesystems Using the `mkfs` Commands

The `mkfs` commands are used to build a new Linux filesystem. The different `mkfs` commands are given in the following table.

If You Need To Build	Use This <code>mkfs</code> Command
An ext2 filesystem	<code>mkfs.ext2 /dev/hdaPartition number</code>
An ext3 filesystem	<code>mkfs.ext3 /dev/hdaPartition number</code>
An ext4 filesystem	<code>mkfs.ext4 /dev/hdaPartition number</code>
An XFS filesystem	<code>mkfs.xfs /dev/hdaPartition number</code>
A reiserfs filesystem	<code>mkfs.reiserfs /dev/hdaPartition number</code>
A btrfs filesystem	<code>mkfs.btrfs /dev/hdaPartition number</code>
A vfat filesystem	<code>mkfs.vfat /dev/hdaPartition number</code>
A JFS filesystem	<code>mkfs.jfs /dev/hdaPartition number</code>

Syntax

The syntax of the mkfs command is `mkfs [filesystem type] [options] {device}`.


The mke2fs Utility

The [mke2fs](#) utility is used to create ext2, ext3, and ext4 filesystems, and it has various options. This command is a more specific version of the [mkfs](#) command described previously that may be used to create ext2, ext3, and ext4 filesystems only. Some of the options are listed in the following table.

Option	Enables You To
-t {filesystem type}	Specify the filesystem type to create (i.e., ext2, ext3, ext4, etc.).
-b {block size}	Specify the size of the block in bytes.
-c	Check the device for errors in the blocks, before creating the filesystem.
-f	Specify the fragment size in bytes.
-j	Create a journaled ext3 filesystem.
-M	Set the directory that was last accessed for the filesystem to be mounted.
-V	Print the version number of the mke2fs utility.

Syntax

The syntax of the mke2fs utility is `mke2fs[options] {device}`.



Note: The command `mke2fs -t ext4 /dev/sdaPartition number` will allow you to build an ext4 filesystem. **WARNING:** Running this command will format your disk, deleting all contents!

Device Recognition by the MBR

Device recognition is performed by the MBR at system startup by recognizing the hard disk and all the partitions on it. The MBR has two main components that help it to detect any devices that are connected to the system.

Component	Description
The Master Partition Table	Contains the list of partitions on the hard disk. Technically, the hard disk can have many partitions. The table displays the partition id, its starting cylinder, and the number of cylinders occupied by the partition.
The Master Boot Code	Contains the program for loading the operating system on the hard disk. This program is loaded to initiate the boot process.

The Cylinder

The [cylinder](#) is the aggregate of all tracks that reside in the same location on every disk surface. On multiple-platter disks, the cylinder is the sum total of every track with the same track number on every surface. On a hard disk, a cylinder comprises the top and corresponding bottom tracks.

Partition Management

[Partition management](#) is the process of creating, destroying, and manipulating partitions to optimize system performance. Effective partition management enables you to keep track of the data in the partitions and avoid data overflow. Various utilities, such as sfdisk, GNU parted, gdisk, and partprobe, are available for partition management.

The sfdisk Utility

The [sfdisk](#) utility is used to manipulate partitions. This utility manages partitions by listing the number of partitions and their sizes, checking the partitions, and repartitioning a storage device.

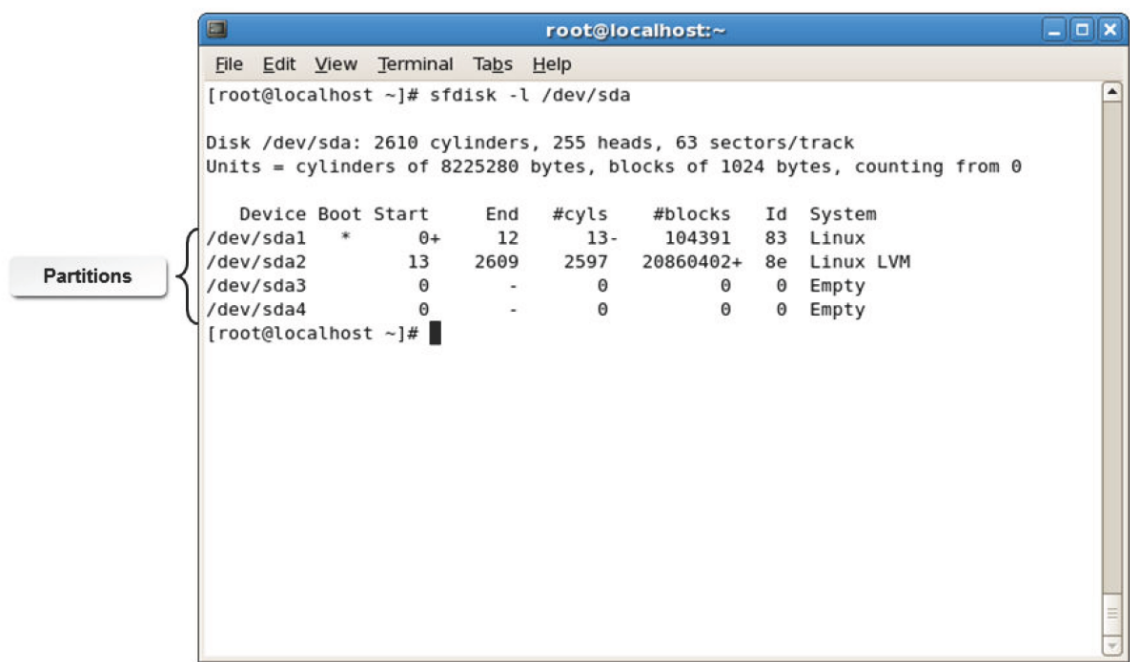


Figure 3-6: Listing partitions on the hard disk using the sfdisk utility.

Various options are available in the sfdisk utility to manage partitions.

Option	Enables You To
-s	List the partition size.
-l {device}	List partitions on all hard disks.
-V {device}	Check for consistency in all partitions.
device	Repartition hard disks. However, if the code is wrongly entered, it may lead to loss of data.
-i	Increment numbers starting with 1 instead of 0 for all cylinders in the hard disk.
-A {number}	Activate the partition indicated by the partition number while making all other partitions inactive.

Syntax

The syntax of the sfdisk utility is `sfdisk [options] device`.

The GNU parted Utility

The [GNU parted](#) utility is also used to manage partitions. It is particularly useful when creating partitions on new hard disks. It can be used to create, destroy, and resize partitions. This utility is generally not used for resizing ext3 partitions.

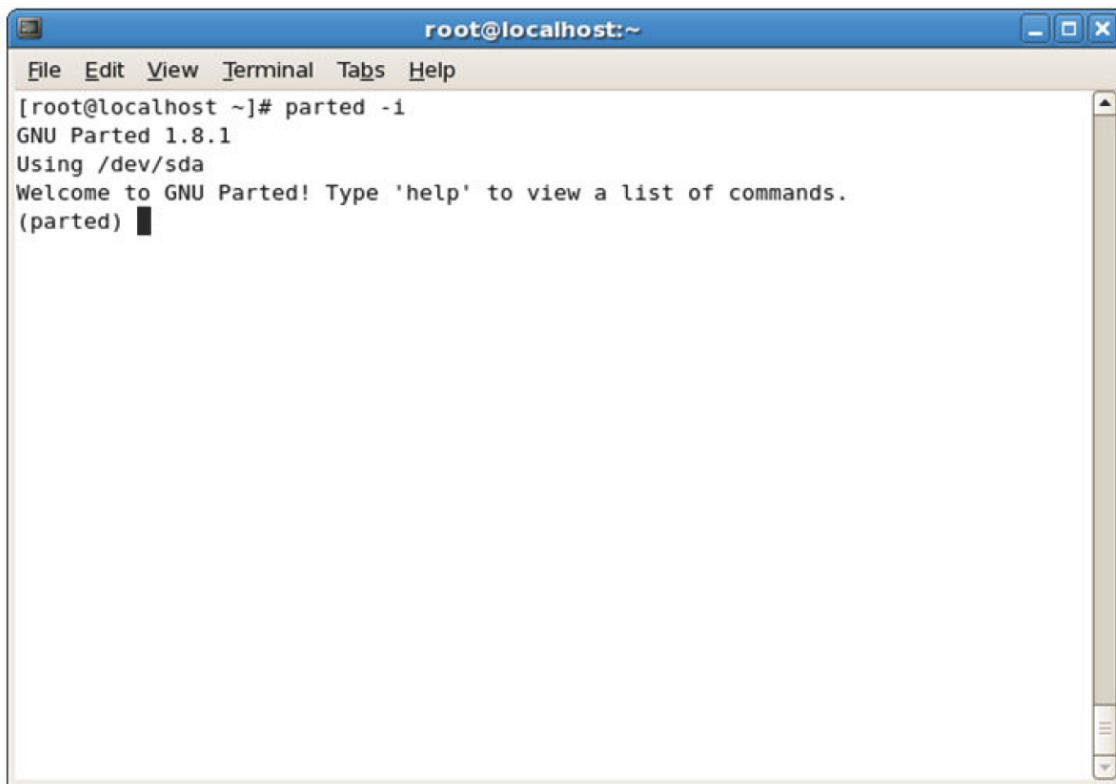


Figure 3-7: The welcome screen of the GNU parted utility.

A number of options are available in the GNU parted utility.

<i>Option</i>	<i>Enables You To</i>
-h	Display a help message.
-v	Display the version of GNU parted.
-i	Configure parted to ask for user input.
-s	Stop parted from asking for user input.

Syntax

The syntax of the parted utility is `parted [option] device {command [argument]}`.

The gdisk Utility

The [gdisk](#) utility is used to manipulate disk partitions and is short for GPT fdisk. This utility manages partitions in the newer Globally Unique Identifier (GUID) Partition Table (GPT) format by listing the number of partitions and their sizes, checking the partitions, and repartitioning a storage device.

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# gdisk /dev/sdb  
GPT fdisk (gdisk) version 0.8.6  
  
Partition table scan:  
  MBR: protective  
  BSD: not present  
  APM: not present  
  GPT: present  
  
Found valid GPT with protective MBR; using GPT.  
  
Command (? for help): p  
Disk /dev/sdb: 16777216 sectors, 8.0 GiB  
Logical sector size: 512 bytes  
Disk identifier (GUID): 7CE46550-1712-45E2-BEE5-561B00B544FF  
Partition table holds up to 128 entries  
First usable sector is 34, last usable sector is 16777182  
Partitions will be aligned on 2048-sector boundaries  
Total free space is 2014 sectors (1007.0 KiB)  
  
Number  Start (sector)    End (sector)  Size      Code  Name  
   1           2048         16777182   8.0 GiB   8300   Linux filesystem  
  
Command (? for help): █
```

Figure 3-8: Viewing the current partition table with the gdisk utility.

Command Line Options Supported by gdisk

The gdisk utility supports a single command line option.

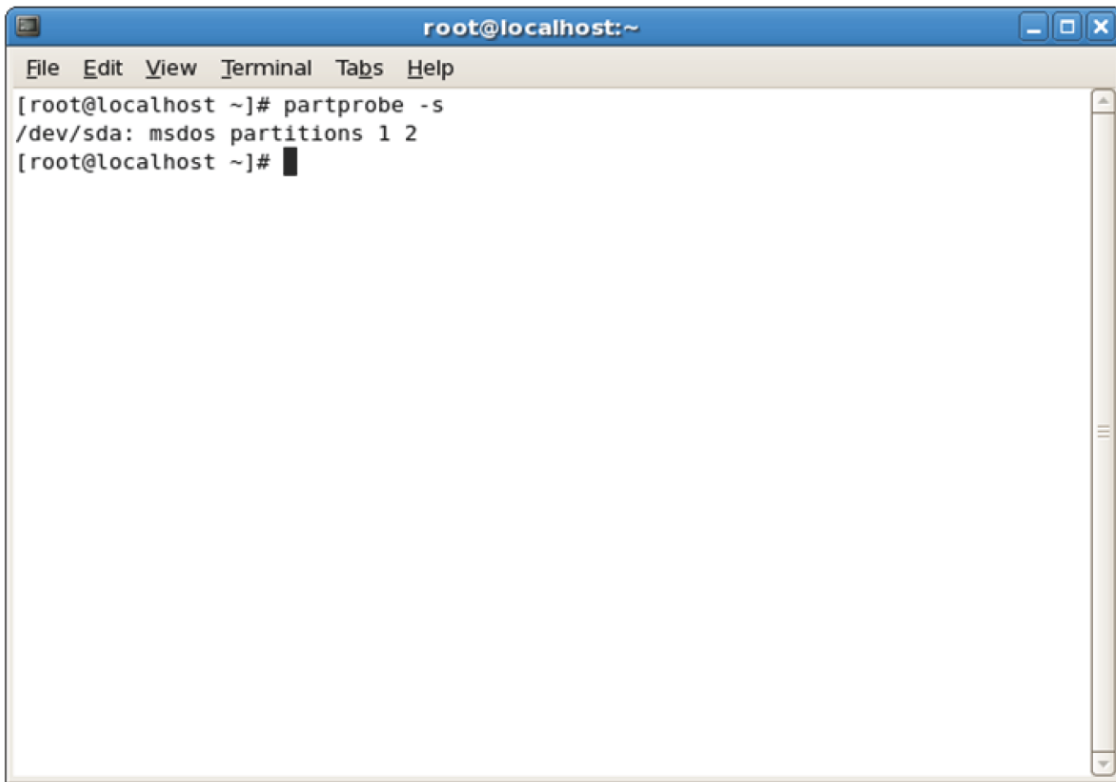
Option	Enables You To
-l	List partition tables for devices and exit.

Syntax

The syntax of the gdisk utility is `gdisk [options] {device name}`.

The partprobe Program

The [partprobe](#) program is used to update the kernel with changes in the partition tables. The program first checks the partition table, and if there are any changes, it automatically updates the kernel with the changes.

A screenshot of a terminal window titled 'root@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal shows the command '[root@localhost ~]# partprobe -s' and its output: '/dev/sda: msdos partitions 1 2'. The prompt '[root@localhost ~]#' is shown again with a cursor.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# partprobe -s  
/dev/sda: msdos partitions 1 2  
[root@localhost ~]#
```

Figure 3-9: Using the *partprobe* utility to display the storage devices and their partitions.

The *partprobe* program has several options.

Option	Enables You To
-d	Cancel any updates.
-s	Display the storage devices and their partitions.
-v	Display the version of the <i>partprobe</i> program.

Syntax

The syntax of the *partprobe* utility is *partprobe [options] [device]*.

How to Create Disk Partitions

Follow these general procedures to create disk partitions.

Create a Primary Partition

To create a primary partition:

1. Log in as **root**.
2. To partition the disk, enter `fdisk /dev/{device name}`.
3. To create a partition, enter `n`.
4. Create a primary partition.
 - a. To create a primary partition, enter `p`.
 - b. To accept the default starting point of the partition, press **Enter**.
 - c. Specify the partition size.

- To accept the default partition size, press **Enter**.
 - To specify a custom partition size:
 - In blocks, enter **+[Required size]**.
 - In kilobytes (KB), enter **+[Required size]K**.
 - In megabytes (MB), enter **+[Required size]M**.
5. To write the partition table on the disk and exit the utility, enter **w**.
 6. To update the partition table, enter **partprobe** or reboot the system.
 7. To list the partition table, enter **sfdisk -l /dev/{device name}**.

Create an Extended Partition

To create an extended partition:

1. Log in as **root**.
2. To begin disk partitioning, enter **fdisk /dev/{device name}**.
3. To create a partition, enter **n**.
4. Create an extended partition.
 - a. To create an extended partition, enter **e**.
 - b. To accept the default starting point of the partition, press **Enter**.
 - c. To accept the default partition size, press **Enter**.
 - d. To create a logical partition within the extended partition, enter **n**.
 - e. To accept the default starting point of the partition, press **Enter**.
 - f. Specify the partition size.
 - To accept the default partition size, press **Enter**.
 - To specify a custom partition size:
 - In blocks, enter **+[Required size]**.
 - In kilobytes (KB), enter **+[Required size]K**.
 - In megabytes (MB), enter **+[Required size]M**.
5. To write the partition table on the disk and exit the utility, enter **w**.
6. To update the partition table, enter **partprobe** or reboot the system.
7. To list the partition table, enter **sfdisk -l /dev/{device name}**.

Apply Labels to a Partition

To apply labels to a partition:

1. Log in as **root**.
2. To apply a label to the partition, at the command prompt, enter `e2label /dev/{device name} {partition number} {label name}` for an ext2, ext3, or ext4 filesystem. For an XFS filesystem, the command is `xfs_admin -L {label name} /dev/{device name} {partition number}`.
3. To view the applied or associated label, enter `e2label /dev/{device name}{partition number}` for an ext2, ext3, or ext4 filesystem. For an XFS filesystem, the command is `xfs_admin -l /dev/{device name}{partition number}`.
4. If necessary, to mount the partition using its label, enter `mount LABEL ={label name} {mount point}`.

TOPIC B Navigate Through the Linux Filesystem

Now that you partitioned your hard disk properly and efficiently, it is time to move around the Linux filesystem. In this topic, you will navigate through the filesystem.

Navigating through the Linux filesystem will allow you to access, create, and delete files and directories. While being able to navigate through the filesystem in the GUI environment may be easier, the CLI will give you more direct control over the workings of the Linux system.

Filesystem Hierarchy

Linux contains regular files that include text files, executable files or programs, input for programs, and output from programs. Besides these, the Linux filesystem consists of other types of files, as described in the following table.

<i>File Type</i>	<i>Description</i>
Directories (d)	Contains the lists of all files.
Special files	Includes system files. These files are in the /dev format. These can be block special files (b) or character special files (c). Block special files are large files that are used for data storage. Character special files are small files that are used for streaming of data.
Links (l)	Makes a file accessible in multiple parts of the system's file tree.
Domain sockets (s)	Provides inter-process networking that is protected by the filesystem's access control.
Named pipes (p)	Allows processes to communicate with each other, without using network sockets.



The file Command

The file command is used to determine the type of file. The syntax of the command is file

[options] {file name}.

The FHS

The **Filesystem Hierarchy Standard (FHS)** is a collaborative document that specifies a set of guidelines for the names of files and directories and their locations. The important advantages of implementing the guidelines of the FHS include compatibility between the systems that are FHS compliant and restriction on users changing the /usr partition that contains common executable files.



	Note: The restriction on users to prevent changes to the /usr partition is achieved by mounting /usr as a read-only partition.
	Note: The complete documentation for FHS is available at www.pathname.com/fhs/ .

Standard Directories

The Linux operating system comprises directories that enable you to organize user files, drivers, logs, programs, and utilities into different categories. In Linux, a forward slash (/) represents the **root** directory, which is the topmost directory, and all other directories are subdirectories under it.

Some of the standard root directories are described in the following table.

<i>Directory</i>	<i>Description</i>
/boot	Stores the files necessary to boot the Linux operating system. The /boot partition must be present in the first sector of the hard disk, from which the system boots. For example, the /boot/grub/menu.lst file.
/bin	Stores essential command line utilities and binaries. For example, the /bin/ls file.
/dev	Stores hardware and software device drivers. It maintains filesystem entries that represent the devices connected to the system. For example, the /dev/sda1 driver.
/etc	Stores basic configuration files. For example, the /etc/samba/smb.conf file.
/lib	Stores shared program libraries required by the kernel, command line utilities, and binaries. For example, the /lib/libc.so.6 file.
/sbin	Stores binaries that are used for completing the booting process and also the ones that are used by the root user—the administrator. For example, the /sbin/ifconfig file.
/usr	Stores small programs and files accessible to all users. For example, the /usr/share/doc file.
/var	Stores system log files, printer spools, and some networking services' configuration files. For example, the /var/log/messages file.
/tmp	Stores temporary files. For example, the /tmp/filename.tmp file.
/opt	Stores files of large software packages. These packages normally create a subdirectory bearing their name under the /opt directory and then place their files in the subdirectory. For example, the /opt/nessus file.
/mnt	Is the mount point for temporarily mounting data from locations such as floppy disks, CDs, DVDs, and network partitions.
/media	Allows access to temporary and removable filesystems such as CD-ROMs and floppy disks.

	Note: In Linux, every user, except the root user, is assigned a specific directory by default in / home to work. Users can then create subdirectories and files within this directory. As soon as the users log in to the system, their own control is automatically placed in their home directory. The home directory of the root user is /root.
	Note: Based on your need, allocate disk space to each directory in the FHS. For example, when connected to a network with more than a hundred users, you can allocate more disk space to the /home directory so that users each can be allotted more storage space on their home directory.

/usr Subdirectories

The **/usr** directory contains some important subdirectories.

<i>Subdirectory</i>	<i>Description</i>
/usr/bin	Includes executable programs that can be executed by all users.
/usr/local	Includes custom build applications that are stored here by default.
/usr/lib	Includes object libraries and internal binaries that are needed by the executable programs.
/usr/lib64	Serves the same purpose as /usr/lib, except that it is meant only for 64-bit systems.
/usr/share	Includes read-only architecture independent files. These files can be shared among different architectures of an operating system.

File Naming Conventions

A file name is a string of characters that identify a file. By using the right combination of characters in file names, you can ensure that the files are unique and easy to recognize.

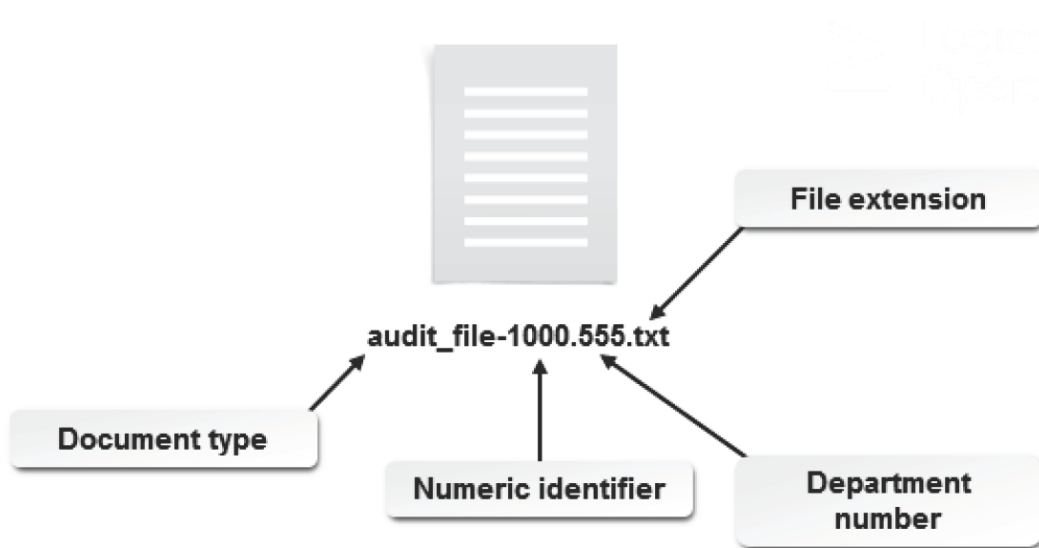


Figure 3-10: File naming conventions.

On an ext4 filesystem, a file name may be up to 255 bytes long and contain any byte except NULL ('\0') and the forward slash ('/'). File names of user files may not be "." and ".." as these are special reserved file names. Various filesystems may enforce different requirements for file names.

Creating File Names with Space

You can create file names with spaces between characters by including a backward slash (\) along with a space in the file name. For example: `touch audit_file-1000.555.txt` will create a file named `audit_file-1000.555.txt`. Although file names may contain a space, convention on Linux systems dictates that words in a file name are more frequently demarcated by capitalization, a hyphen, or an underscore, as these are all easier to manage on the command line. For example:

`AuditFile.txt` or `Audit_File.txt`.

File Browsers

In Linux, you can navigate through a filesystem using a file browser. The default file browser on GNOME desktops in Red Hat and Fedora distributions is the Nautilus browser. This browser operates in two modes.

Mode	Description
Spatial	This is the default mode that enables you to open a particular window at exactly the same position on the screen by remembering the last position. Each folder or directory that you select is opened in a new browser window.
Browser	This is the mode that enables you to display the selected folder in the same window. You need to modify the preferences in the Computer window.

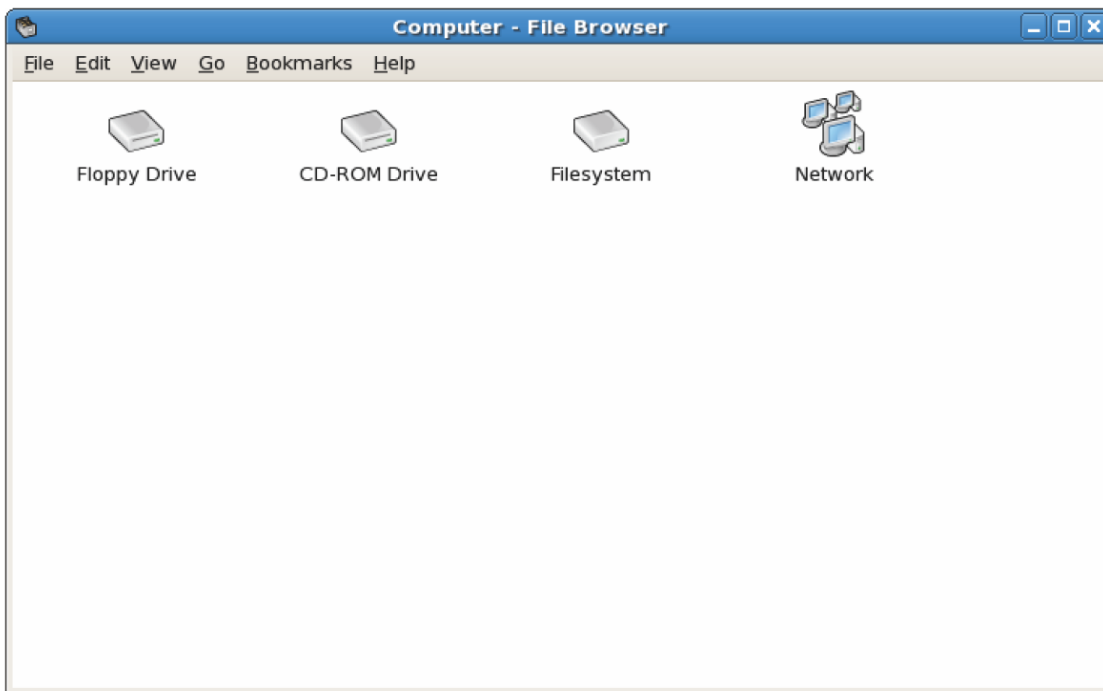


Figure 3-11: The Nautilus file browser.

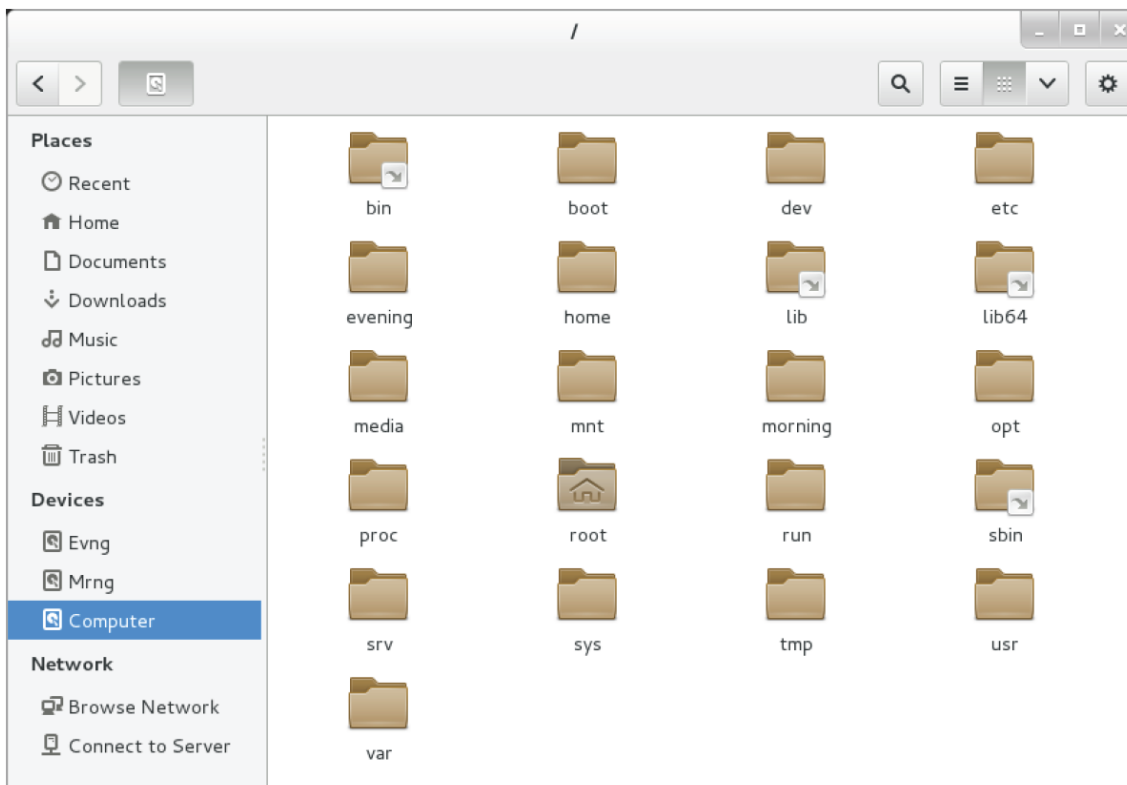


Figure 3-12: The GNOME Files file browser.

The Home Directory

The **home directory** is where you are placed when you log in to the system. In Linux, by default, every user, except the root user, is assigned a specific directory in /home. In many shells, including Korn, C shell, and Bash, the tilde character (`~`) represents your home directory. A user can create subdirectories and files within this directory. As soon as the user logs in to the system, control is automatically transferred to the user's home directory. The home directory of the **root user** is **/root**.

The root user can access all files and resources on the system.

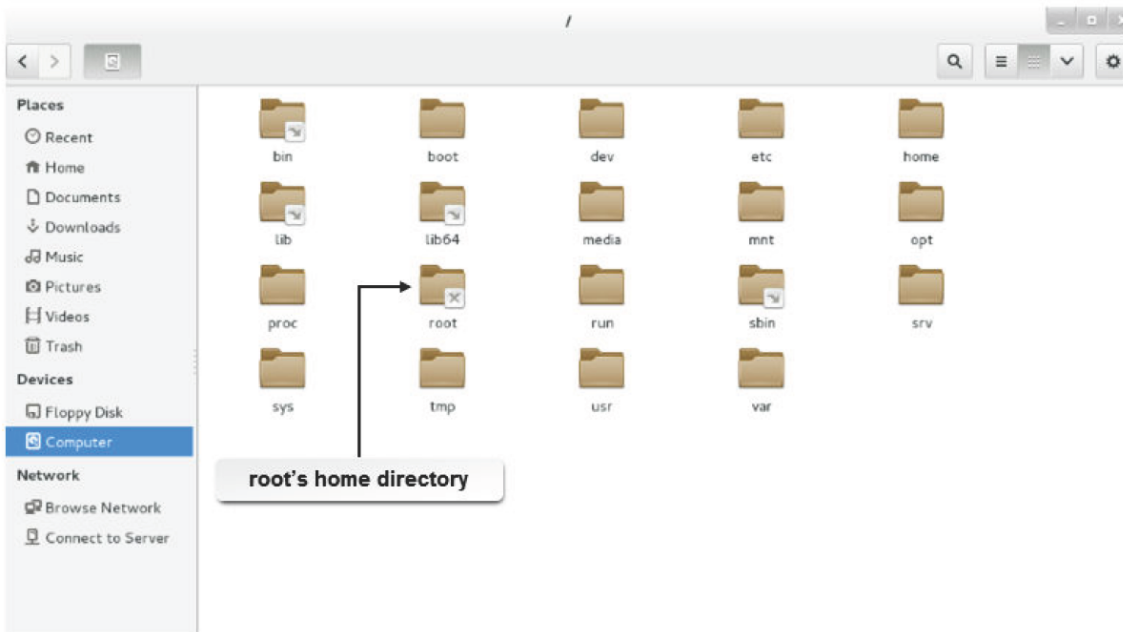


Figure 3-13: The home directory of the root user.

The Current Working Directory

The **current working directory** is the location on the system that you are accessing at any point in time. For example, when you log in to a system, you are placed in your home directory. So, your current working directory is your home directory. The current working directory can be listed in shorthand with a period (.).



Figure 3-14: Viewing the current working directory using the `pwd` command.

The `pwd` Command

When you navigate through a filesystem, you may need to know your current working directory. The `pwd` command displays your current working directory relative to the root directory. It displays the full path name.

The Parent Directory

The **parent directory** is one level above your current working directory. All directories, except the root directory, have a parent directory. You can use the double period notation to switch to the parent directory.

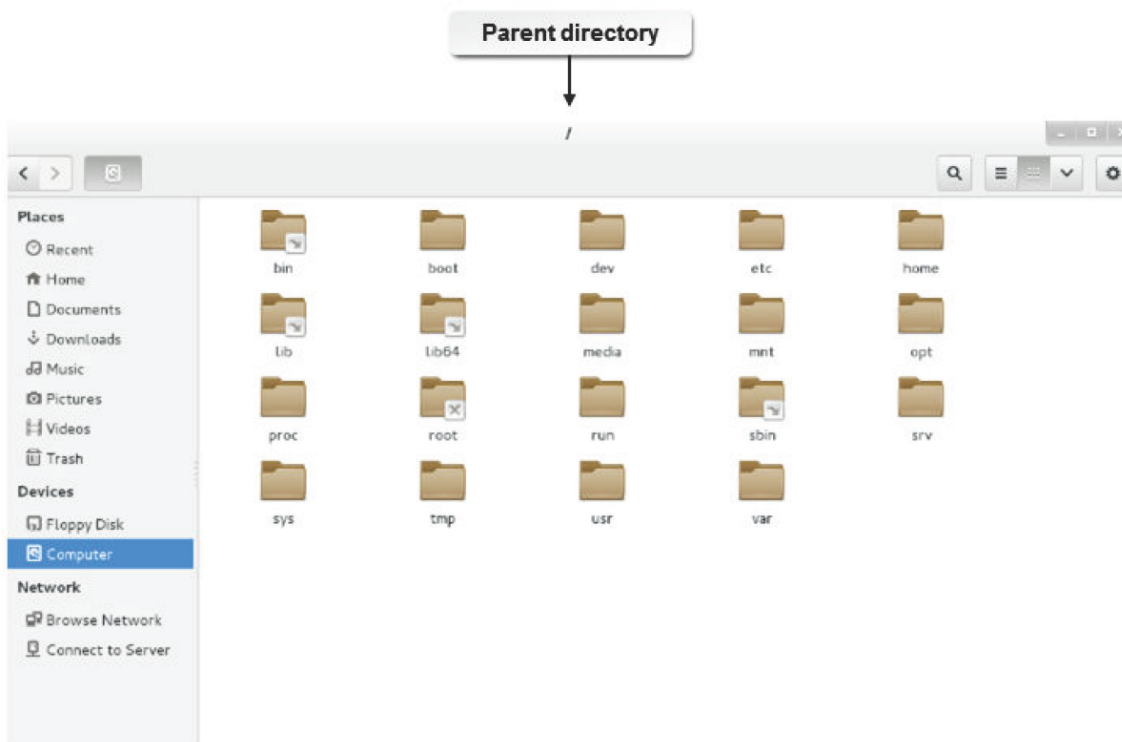


Figure 3-15: The root (/) directory is the parent directory for all other directories.

Paths

A **path** specifies a location in the filesystem. It begins with the root directory, the directory at the top of the directory tree, and ends with the directory or file you want to access.

You can refer to a particular file by providing a path to the specific directory that contains the file.

For example, the directory **jsmith** contains a subdirectory, **work**, which contains a file named **mywork**. To refer to that file, use the following path name: `/home/jsmith/work/mywork`. Notice that the forward slash (/) character is used to separate items in the path. The slash that precedes `jsmith` represents the root directory, from which the path to the file, **mywork**, begins.

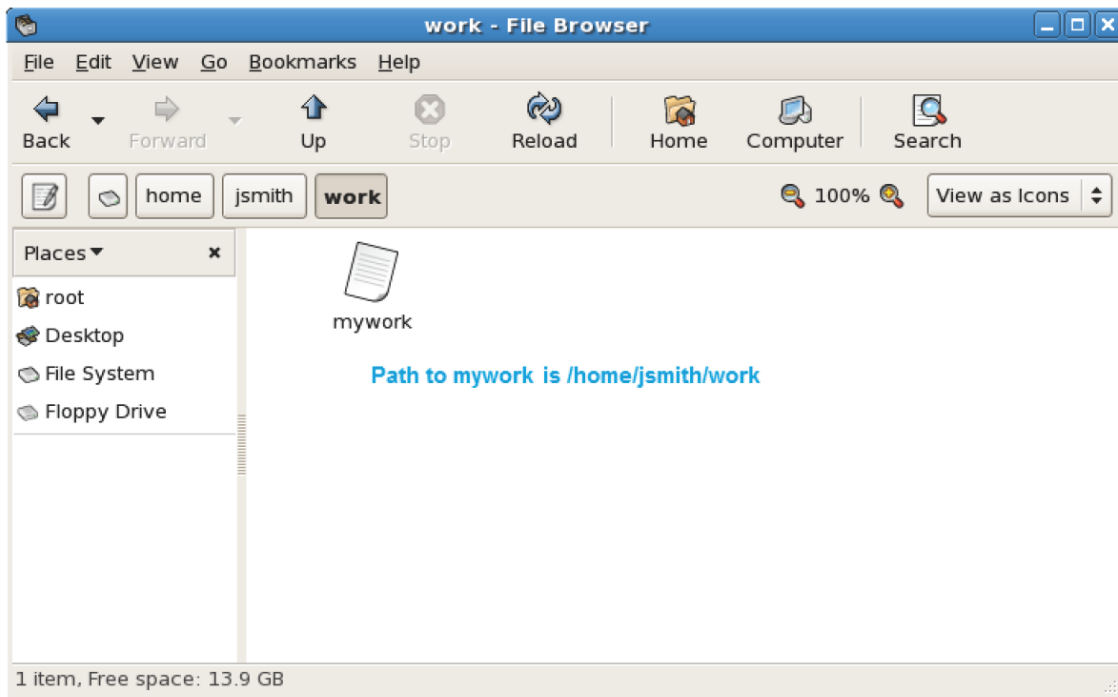


Figure 3-16: The path to the file mywork.

Absolute and Relative Paths

Paths are of two types—absolute and relative. **Absolute path** refers to the specific location, including the domain name, irrespective of the current working directory or combined paths. These paths are usually written with reference to the root directory, and therefore start with a forward slash. Paths that do not begin with a forward slash are called relative paths. A **relative path** is the path relative to the current working directory; therefore, the full absolute path need not be included.

These paths can contain the period [.] and double period [..], which are indications for the current and parent directories.

Figure 3-17: Listing files in a directory using the absolute path.

Basic Filesystem Commands

There are some basic filesystem commands that will allow you to modify files and display information within the Linux filesystem.

Command	Enables You To
cd	Traverse the directory structure. There are several ways to specify the path name of the directory you need to switch to. The syntax of the cd command is <code>cd {absolute or relative path}</code> .
ls	List the files in the current working directory. This command displays only the file name when the command is run without any options. However, it can be used to list information such as size, file type, and permissions by running the command with the respective options. The syntax of the ls command is <code>ls [options][absolute or relative path of the directory]</code> .
mv	Move files and directories from one directory to another, or rename a file or directory. The syntax of the mv command is <code>mv {absolute or relative path}/{file or directory name} {absolute or relative path}/{new file or directory name}</code> .
cp	Copy a file. The syntax of the cp command is <code>cp [options] {absolute or relative path of the file or directory to be copied}/ {file or directory name} {absolute or relative path of the destination}</code> . You can use the -R option of the cp command to copy files along with the source directory recursively. The syntax is: <code>cp -R / {source directory} / {target directory}</code> .
rm	Delete files or directories. The syntax of the rm command is <code>rm [options] {absolute or relative path of file or directory}/ {file or directory name}</code> . You can use the -R option of the rm command to recursively remove files, subdirectories, and the directory itself. The syntax is: <code>rm -R {directory and content that needs to be deleted}</code> .
touch	Change the time of access or modification time of a file to the current time. In addition, the touch command creates an empty file if the file name specified as an argument does not exist. The syntax of the touch command is <code>touch {file name}</code> .
mkdir	Create a directory. The syntax of the mkdir command is <code>mkdir {directory name}</code> .
rmdir	Delete directories. The syntax of the rmdir command is <code>rmdir {directory name}</code> .
pushd	Add a directory at the top of a stack of directories or rotate a stack of directories. The syntax of the pushd command is <code>pushd [options] {directory name}</code> .
popd	Remove entries from a stack of directories. When no option is specified, it removes the top directory from the stack. The syntax of the popd command is <code>popd [options]</code> .

The -v Option

-v is a command option that can be used with the basic file management commands. This option explains the running of the command to produce the desired output, in a verbose manner.

The ls Command Options

The ls command options are described in the following table.

Option	Description
-l	Displays a long list including the permissions, number of hard links, owner, group, size, date, and file name.
-F	Displays the nature of a file, such as * for an executable file and / for a directory.
-a	Displays all files present in the directory, including the files whose names begin with a period (.).
-R	Recursively displays all subdirectories.
-d	Displays information about symbolic links or directories rather than the link's target or the contents of the directory.
-L	Displays all files in a directory, including symbolic links.

Changing the Current Directory

There are times when you need to move out of your home directory into another directory in the filesystem. In such situations, you can use the cd command to change directories. The cd command enables you to traverse the

directory structure. There are several ways to specify the path name to the directory that you wish to make your working directory:

- The `cd` command without a path name takes you to your home directory, irrespective of your current directory.
- The `cd [path name]` command takes you to the path name specified. The path name can be the full path name (from the root down to the specified directory) or the relative path name (starting from your current working directory).
- The `cd ~/[path name]` command takes you to the specified directory, relative to your home directory. Remember to replace `~/[path name]` with `$HOME`, if necessary.

How to Navigate Through the Linux Filesystem

Follow these general procedures to navigate through the Linux filesystem.

Change Directories

To change directories:

1. Log in as a user.
2. To view the present working directory, enter `pwd`.
3. To change to the target directory, enter `cd {absolute or relative path of the target directory}`.
4. If necessary, to verify if you have changed to the target directory, enter `pwd`.

List Files and Directories

To list files and directories:

1. Log in as a user.
2. To list the files and directories, enter the `ls [options]` command.

Work with Files and Directories Using the GNOME Files Browser in Spatial Mode

To work with files and directories using the GNOME Files browser in spatial mode:

1. Log in as a user in the GUI.
2. Double-click the **Computer** icon on the desktop.
3. Double-click the desired directory to open it.
4. Double-click the desired file in the directory.
5. If necessary, to close all parent windows, press **Ctrl+Shift+W**.
6. To close the GNOME Files browser, click the **Close** button.

Work with Files Using the GNOME Files Browser in Browser Mode

To work with files using the GNOME Files browser in browser mode:

1. Log in as a user in the GUI.
2. From the menu bar, choose **Application → System Tools → File Browser**.
3. Double-click the desired file or directory either in the right pane or in the left pane to view it.
4. Close the GNOME Files browser.

Set the Nautilus Browser to Open Always in Browser Mode

To set the Nautilus browser to open always in browser mode:

1. Log in as a user in the GUI.
2. Double-click the **Computer** icon on the desktop.
3. In the Computer window, choose **Edit → Preferences**.
4. In the **File Management Preferences** dialog box, select the **Behavior** tab.
5. In the **Behavior** section, check the **Always open in browser windows** check box and select **Close**.
6. Close the Computer window.
7. To open the Nautilus browser in browser mode, double-click the **Computer** icon.

TOPIC C Manage the Filesystem

Now that you can navigate through the Linux filesystem, it is time to learn how to manage it. In this topic, you will manage the Linux filesystem.

Managing the Linux filesystem will allow you to customize the system to suit your requirements.

Also, you will be able to organize files and directories, mount additional drives, and use recordable media for backups or storage.

Burning Discs

While managing your filesystem, you may have to back up some data on discs. Linux allows you to burn CDs and DVDs with GUI-based programs, as well as from the CLI. GUI-based programs guide you through the burning process.

Note: You must have a CD or DVD writer installed on your system to be able to burn discs.
--

ISO Images

An **ISO image** or **disk image** is an archive file format for files that are to be written to optical discs such as CDs and DVDs. It is a standard defined by the International Organization for Standardization (ISO) and has a file extension of .iso.

Mount Points

A **mount point** is an access point to information stored on a local or remote storage device. The mount point is typically an empty directory on which a filesystem is loaded, or mounted, to make the filesystem accessible to users. If the directory already has content, the content becomes invisible to the users until the mounted filesystem is unmounted.

	Note: You can use the <code>/etc/fstab</code> file to list the filesystem to be mounted and unmounted when the Linux system boots and shuts down, respectively.
--	--

Figure 3-18: The process of mounting a filesystem.

The mount Command

In Linux, a filesystem cannot be accessed directly. It has to be associated with a directory to make it accessible to users. This association is brought about by loading, or mounting, the filesystem in a directory by using the mount command. After using the filesystem, it needs to be disassociated from the directory by unloading, or unmounting, the filesystem using the umount command.

Figure 3-19: A list of currently mounted filesystems.

mount Command Options

You can specify various mount options for a filesystem.

<i>Option</i>	<i>Enables You To</i>
auto	Specify that the device has to be mounted automatically.
noauto	Specify that the device need not be mounted automatically.
nouser	Specify that only the root user can mount a device or a filesystem.
user	Specify that all users can mount a device or a filesystem.
exec	Allow binaries in a filesystem to be executed.
noexec	Prevent binaries in a filesystem from being executed.
ro	Mount a filesystem as read-only.
rw	Mount a filesystem with read and write permissions.
sync	Specify that input and output operations in a filesystem should be done synchronously.
async	Specify that input and output operations in a filesystem should be done asynchronously.

Binaries

Binaries are source codes that are compiled into executable programs, or are assembled so that they are readable by the computer system. Binaries are encoded so that they can be transmitted over the Internet. In addition, binaries can be pictures, word processing files, or spreadsheet files. Some binaries may contain viruses that can harm the system.

Swap Space

Swap space is a partition on the hard disk that is used when the system runs out of physical memory. Linux pushes some of the unused files from the RAM to the swap space to free up memory. Usually, the swap space equals twice the RAM capacity.

Figure 3-20: Swap space being created on a hard disk.

Swap space can be one of three types.

Swap Type	Description
Device swap	Device swap space is configured when you partition the hard disk. It is used by the operating system to run large applications.
Filesystem swap	Filesystem swap space is configured primarily when you install Linux. It is utilized by the operating system as an emergency resource when the available swap space runs out.
Pseudo-swap	Pseudo-swap space allows large applications to run on computers with limited RAM.

Swap Files

Swap files are created for storing data that is to be transferred from a system's memory to a disk. It is dynamic and changes in size when data is moved in and out of the memory. It is used as a medium to transfer data from RAM on to the hard disk.

Swap Partitions

A swap partition is an area of virtual memory on a hard disk to complement the physical RAM in the computer. Swap partitions are created by Linux because they perform better than swap filesystems.

The mkswap Command

The **mkswap** command is a system administration command that is used to create swap space on a disk partition. It provides options to perform various tasks.

Option	Enables You To
-c	Verify that the device is free from bad sectors before mounting the swap space.
-f	Force a swap partition of an area larger than the permissible limit.
-p	Set the page size to be used by the mkswap command.
-L {label}	Activate the swap space using labels applied to partitions or filesystems.

Syntax

The syntax of the mkswap command is **mkswap [options] device {size}**. The device argument of mkswap is generally a disk partition, such as /dev/hda2 or /dev/sdb3, but it can also be a file.

Swap Partition Management Commands

A number of commands are used to manage swap partitions. The most important commands are

swapon and swapoff.

Command	Description
swapon	Used to activate a swap partition on a specified device. It provides a number of options for specifying devices.
swapoff	Used to deactivate the swap space on devices.

The swapon and swapoff Command Options

Some of the frequently used swapon and swapoff command options are given in the following table.

Option	Description
---------------	--------------------

<i>Option</i>	<i>Description</i>
swapon -e	It is used to skip devices that do not exist.
swapon -a	It is used to activate all the swap space.
swapoff -a	It is used to deactivate all the swap space.

How to Manage the Linux Filesystem

Follow these general procedures to manage the Linux filesystem.

Burn Discs from the CLI

You will need a CD or DVD writer installed on your system to be able to burn discs. To burn discs from the CLI:

1. Create a directory and copy the files you would like to burn.
2. Make an ISO image of the files using the mkisofs command.
3. Burn the CD using the cdrecord command.

Create a Mount Point

To create a mount point:

1. Log in as **root**.
2. To create a mount point, enter `mkdir {mount point}`.
3. To set the user as the owner of the mount point, enter `chown {user name} {mount point}`.
4. To set the group as the owner of the mount point, enter `chgrp {group name} {mount point}`.

Mount a Filesystem

To mount a filesystem:

1. Log in to the CLI as **root**.
2. To mount the specified device on the specified mount point, enter `mount [options] /dev/ {device name} {partition number} {mount point}`.
3. To verify that the filesystem is mounted on the specified mount point, enter `mount {mount point}`.

Mount Filesystems at Startup

To mount filesystems at startup:

1. Log in to the CLI as **root**.
2. To open the **/etc/fstab** file, enter `vi /etc/fstab`.
3. To add an entry for the new filesystem, type `{filesystem label} {device or partition name} {mount point} {filesystem type} {mount options} {dump options} {fsck options}`.

4. Save and close the file.
5. To reload the mount table with recent changes from the **/etc/fstab** file, reboot the system or enter `mount -a`.
6. Verify that the filesystem has been automatically mounted at startup.
 - a. Log in as **root**.
 - b. To view all the mounted filesystems, enter `mount`.

Unmount a Filesystem

To unmount a filesystem:

1. Log in to the CLI as **root**.
2. Unmount a filesystem.
 - To unmount the filesystem, enter `umount [options] /dev/{device name}{Partition number}`.
 - Enter `umount [options] {mount point}`.

Manage a Filesystem

To manage a filesystem:

1. Log in to the CLI as **root**.
2. To display the details about the processes using the filesystem, enter `fuser {mount point}`.
3. To kill all processes using the filesystem, enter `fuser -km {mount point}`.
4. To unmount the filesystem, enter `umount {mount point}`. Note that the filesystem cannot be unmounted while it is being used by another process.

Manage Swap Partitions

To manage swap partitions:

1. Log in to the CLI as **root**.
2. To create a swap partition, enter `mkswap /dev/{device name}{partition number}`.
3. To add the partition entry, open the **/etc/fstab** file in `vi` and type `{filesystem label} {device name}{partition number} none swap {mount options}{dump options} {fsck options}`.
4. To activate the swap partition, enter `swapon {device name}`.
5. To deactivate the swap partition and convert it into a standard Linux filesystem, enter `swapoff {device name}`.

Format a Partition with a Filesystem

To format a partition with a filesystem:

1. Log in to the CLI as **root**.
2. Format a partition with a filesystem.
 - To create a specified filesystem on a specified partition of the device, enter `mkfs -t {filesystem type} /dev/{device name}{partition number}`.
 - To create an ext2 filesystem on the specified partition of the device, enter `mke2fs [options] /dev/{device name}{partition number}`.

TOPIC D Maintain the Filesystem

After managing a Linux filesystem, it is important to learn how to keep it up and running. In this topic, you will maintain the Linux filesystem.

Maintaining the Linux filesystem will assist you in troubleshooting and general maintenance of your system. If a power outage or other unplanned shutdown occurs, you will need to know how to verify the data integrity of your local drives. Maintaining a filesystem in Linux involves the following tasks:

- Checking the integrity of the filesystem.
- Managing the size of partitions.
- Removing temporary files.
- Performing system recovery.

Storage Devices

There are different types of storage devices in Linux. Each device has a particular use associated with it.

Device	Description
Hard disk	An internal device that can store large amounts of data. It can be accessed quickly.
Floppy drive	<div><div>A removable medium that can store smaller amounts of data. It cannot be accessed as quickly as a hard disk.</div><div>Note: Floppy drives are legacy technology, so if you encounter them in the field, they will very likely be installed in older systems.</div></div>
Tape drive	A device that is used to store large amounts of data on a magnetic tape. Tape drives can be internal or external. External tape drives are portable, whereas in internal tape drives only the tape is removable. Data is accessed sequentially in a tape drive.
Flash drive	A small, portable, storage device that is used to store files that need to be carried around.
CD-R(W)	A removable optical disc that stores 650-700 MB of data. It can be accessed faster than other removable storage media.
DVD-R(W)	A removable optical disc that stores 4.5 GB (or more) of data. It can be accessed faster than other removable storage media.

Mass Storage Devices

Mass storage devices are types of storage devices that provide fast access to large amounts of data in a small, reasonably reliable, physical package. Hard disks, tape drives, flash drives, CD-R(W), DVD-R(W), and zip drives are some of the common mass storage devices.

ATAPI

AT Attachment Packet Interface (ATAPI) is a protocol for controlling mass storage devices.

ATAPI provides commands that are used for hard disks, CD-ROM drives, tape drives, and other devices.

Journaling Filesystems

A **journaling filesystem** is a method that is used by an operating system to quickly recover after an unexpected interruption, such as a system crash. Journaling filesystems can remove the need for a filesystem check when the system boots. By using journaling filesystems, the system does not write modified files directly on the disk. Instead, a journal is maintained on the disk. The journaling filesystem process involves the following phases:

1. The journal describes all the changes that must be made to the disk.
2. A background process makes each change as and when it is entered in the journal.
3. If the system shuts down, pending changes are performed when it is rebooted.
4. Incomplete entries in the journal are discarded.

Performance Issues with Journaling

A journaled filesystem works well with small files and small drives. With the growth of file and drive sizes, performance will suffer. Some of the reasons for poor performance include:

- Filesystem recovery time after a power failure or improper shutdown.
- Bitmap method of tracking the filesystem.
- Wasted space and fragmentation.

The fsck Command

The **fsck** command is used to check the integrity of a filesystem. **Filesystem integrity** refers to the correctness and validity of a filesystem. Most systems automatically run the fsck command at boot time so that errors, if any, are detected and corrected before the system is used. Filesystem errors are usually caused by power failures, hardware failures, or improper shutdown of the system.

Note: The fsck command is similar in concept to the chkdsk and scandisk commands you may be familiar with from DOS and Windows-based systems.
--

Figure 3-21: Checking the integrity of a filesystem from single-user mode.

Syntax

The syntax of the fsck command is `fsck -t {filesystem type} [options]`.

Repair Filesystems

You can use the `fsck -r/dev/{filesystem}` command to repair a filesystem. The command will prompt you to confirm your actions. If you are simultaneously checking multiple filesystems, you should not use this option because it allows you to repair only a single filesystem at a time.

The e2fsck Command

The `e2fsck` command allows you to check ext2, ext3, and ext4 filesystems, and is identical to running the `fsck` command with ext2, ext3, or ext4 specified as the filesystem type. You need to unmount the filesystem before running the `e2fsck` command to prevent damage to the filesystem.

The syntax of the `e2fsck` command is `e2fsck /dev/{filesystem}`.

The xfs_repair Command

The `xfs_repair` command allows you to check an XFS filesystem. As with the `fsck` and `e2fsck` commands, you need to unmount the filesystem before running the `xfs_repair` command to prevent damage to the filesystem.

The syntax of the `xfs_repair` command is `xfs_repair [options]/dev/{filesystem}`.

The tune2fs Utility

The [tune2fs](#) utility helps tuning parameters associated with a Linux filesystem. Using this utility, a journal can be added to an existing ext2 or ext3 filesystem. If the filesystem is already mounted, the journal will be visible in the root directory of the filesystem. If the filesystem is not mounted, the journal will be hidden. The `tune2fs` utility is available with most Linux distributions.

Tunable Parameters

Using the tune2fs utility, you can adjust the parameters of the extended filesystems, such as ext2, ext3, and ext4, that can be tuned on a Linux machine even after installation. Tunable parameters allow you to remove reserved blocks; alter reserved block count; and specify the number of mounts between checks, the time interval between checks, and the behavior of the kernel code, among others.

Options of the tune2fs Utility

The tune2fs utility has various options.

<i>Use This Option</i>	<i>To Do This</i>
-j {partition}	Convert the existing filesystem to an ext3 filesystem.
-i d m w	Specify the maximum time interval between filesystem checks in days, months, or weeks.
-c maximum mounts count	Specify the maximum number of mounts between filesystem checks.
-C mount count	Specify the number of times the filesystem can be mounted.
-r reserved blocks count	Specify the number of reserved filesystem blocks.
-e continue remount-ro panic	Specify the behavior of the kernel code, whether the filesystem should continue with normal execution, remount the filesystem in read-only mode, or cause a kernel panic, when errors are detected.
-l	List the contents within the superblock of the filesystem.
-U UUID	Set the specified Universally Unique Identifier (UUID) for the filesystem.

Syntax

The syntax of the tune2fs utility is `tune2fs [options] {device name}`.

The xfs_admin Command

The xfs_admin command allows you to manage the parameters of an XFS filesystem. As with the tune2fs command, you need to unmount the filesystem before using the xfs_admin command to change parameters.

The syntax of the xfs_admin command is `xfs_admin [options] /dev/{filesystem}`.

The dumpe2fs Utility

The [dumpe2fs](#) utility is used for managing ext2, ext3, and ext4 (extended) filesystems. It dumps the status of the extended filesystem onto the standard output device and prints the block group information for the selected device.

The dumpe2fs utility has various options.

<i>Option</i>	<i>Enables You To</i>
-x	Print a detailed report about block numbers in the filesystem.
-b	Print the bad blocks in the filesystem.
-f	Force the utility to display the filesystem status irrespective of the filesystem flags.
-i	Display filesystem data from an image file created using the e2image utility.

Syntax

The syntax of the dumpe2fs command is `dumpe2fs [options] [block size] {device name}`.

The debugfs Utility

The debugfs utility allows you to examine and modify ext2, ext3, and ext4 filesystems. When executed, the debugfs utility opens an interactive shell that can be used to examine and modify the extended filesystem.

Commands Supported by the debugfs Utility

The table provides some common commands supported by the debugfs utility in the interactive shell.

<i>If You Need To</i>	<i>Use This Command</i>
Open a filesystem	open /dev/{filesystem}
Close the filesystem	close
View the filesystem information	stats
Find a free block	ffb

xfs Tools

There are many xfs tools that allow you to work with the XFS filesystem.

<i>xfs Tool</i>	<i>Enables You To</i>
xfs_info	Display details about the XFS filesystem.
xfs_metadump	Copy the metadata information of the XFS filesystem to a file.
xfs_grow	Expand the XFS filesystem to fill the disk size.
xfs_repair	Repair and recover a corrupt XFS filesystem.
xfs_db	Debug the XFS filesystem.

How to Maintain the Linux Filesystem

Follow these general procedures to maintain Linux filesystems.

Create ext2 Filesystems

To create ext2 filesystems:

1. If you already have a mounted drive with an existing filesystem:
 - a. Back up all data on the drive to removable media or another drive.
 - b. Unmount the drive using the umount command.
 - c. Build the ext2 filesystem using the mkfs -t ext2 command.
2. If you have an empty drive, build the ext2 filesystem using the mkfs -t ext2 command.
3. Mount the drive using the mount command.
4. To reflect the changes that were done to the filesystem, update the */etc/fstab* file.

Create ext3 Filesystems

To create ext3 filesystems:

1. If you have a drive with the ext2 filesystem:
 - a. Unmount the drive using the `umount` command.
 - b. Convert the filesystem using the `tune2fs -j {partition}` command.
2. If you already have a mounted drive with an existing filesystem other than ext2:
 - a. Back up all data on the drive to removable media or another drive.
 - b. Unmount the drive using the `umount` command.
 - c. Build the ext3 filesystem using the `mkfs -t ext3` command.
 - d. Mount the drive using the `mount` command.
3. If you have an empty drive:
 - a. Build the ext3 filesystem using the `mkfs -t ext3` command.
 - b. Mount the drive using the `mount` command.
4. To reflect the changes that were done to the filesystem, update the `/etc/fstab` file.

Create ext4 Filesystems

To create ext4 filesystems:

1. If you have a drive with the ext2 filesystem:
 - a. Unmount the drive using the `umount` command.
 - b. Convert the filesystem to ext3 using the `tune2fs -j {partition}` command.
 - c. Enable ext4 features on the ext3 filesystem using the `tune2fs -O extents,uninit_bg,dir_index {partition}` command.
 - d. Fix some of the on-disk structures that `tune2fs` has modified using the `e2fsck -fDC0 {partition}` command.
2. If you have a drive with the ext3 filesystem:
 - a. Unmount the drive using the `umount` command.
 - b. Enable ext4 features on the ext3 filesystem using the `tune2fs -O extents,uninit_bg,dir_index {partition}` command.
 - c. Fix some of the on-disk structures that `tune2fs` has modified using the `e2fsck -fDC0 {partition}` command.
3. If you already have a mounted drive with an existing filesystem other than ext2 or ext3:
 - a. Back up all data on the drive to removable media or another drive.
 - b. Unmount the drive using the `umount` command.
 - c. Build the ext4 filesystem using the `mkfs -t ext4` command.
 - d. Mount the drive using the `mount` command.
4. If you have an empty drive:

- a. Build the ext4 filesystem using the `mkfs -t ext4` command.
 - b. Mount the drive using the `mount` command.
5. To reflect the changes that were done to the filesystem, update the `/etc/fstab` file.

Create XFS Filesystems

To create XFS filesystems:

1. If you already have a mounted drive with an existing filesystem other than XFS:
 - a. Back up all data on the drive to removable media or another drive.
 - b. Unmount the drive using the `umount` command.
 - c. Build the XFS filesystem using the `mkfs -t xfs` command.
 - d. Mount the drive using the `mount` command.
2. If you have an empty drive:
 - a. Build the XFS filesystem using the `mkfs -t xfs` command.
 - b. Mount the drive using the `mount` command.
3. To reflect the changes that were done to the filesystem, update the `/etc/fstab` file.

Create reiserfs Filesystems

To create reiserfs filesystems:

1. Verify that you have a kernel version later than 2.4.16.
2. If you already have a mounted drive with an existing filesystem:
 - a. Back up all data on the drive to removable media or another drive.
 - b. Unmount the drive using the `umount` command.
 - c. Build the reiserfs filesystem using the `mkreiserfs` command.
 - d. Mount the drive using the `mount` command.
 - e. To reflect the changes that were done to the filesystem, update the `/etc/fstab` file.
3. If you have an empty drive:
 - a. Build the reiserfs filesystem using the `mkfs -t reiserfs` command.
 - b. Mount the drive using the `mount` command.
 - c. Add the drive to the `/etc/fstab` file.

Manage Local Filesystems

To manage local filesystems:

1. Switch to single-user mode (runlevel 1) by using the `telinit 1` or `systemctl isolate rescue.target` command.
2. Check the filesystem using the `fsck` command.
3. Return to graphical multiuser mode (runlevel 5) by using the `telinit 5` or the `systemctl isolate graphical.target` command.

Manage the ext2, ext3, or ext4 Filesystem Using the debugfs Utility

To manage an ext2, ext3, or ext4 filesystem using the debugfs utility:

1. Log in to the CLI as **root**.
2. To access the debugfs prompt, enter `debugfs /dev/{filesystem}`.
3. To view the commands that are supported in this prompt, at the debugfs prompt, enter `help`.
4. Enter *{command supported by debugfs}*.
5. To quit the debugfs command prompt, enter `quit`.

ACTIVITY 3-1

Managing Partitions and the Linux Filesystem Review

Scenario

Answer the following review questions.

1. When do you think formatting a partition is necessary? Why?
2. What filesystem types have you worked with? What advantages and disadvantages have you encountered with each?

Summary

In this lesson, you created partitions and filesystems on the hard disk. You also navigated, managed, and maintained filesystems.

4 Managing Files in Linux

Lesson Time: 2 hours, 45 minutes

Lesson Introduction

In the previous lesson, you managed the Linux® filesystem. Now it is time to learn how to manipulate files and directories within Linux. In this lesson, you will manage various types of Linux files.

As a Linux administrator, you should keep your files well organized on your system.

Learning how to create, edit, locate, link, back up, and restore files will help you tailor the system to your needs.

Lesson Objectives

In this lesson, you will manage various files in Linux. You will:

- Create a text file in Linux.
- Locate files within the Linux filesystem.
- Search text files using regular expressions.
- Apply filters to text streams.
- Manage links to a file.
- Back up and restore files.
- Manage a database using MariaDB.

TOPIC A Create and Edit Text Files

In the last lesson, you worked with several Linux filesystem types. Now you can move on to creating and editing files within those filesystems. In this topic, you will create and edit text files.

Working with text files is a basic and routine task for most users. Consider a scenario where you may need to submit a report on your current project. You will require an application, such as a text editor, to create the report. You can also use a text editor to create and edit configuration files, which will allow you to customize your system.

Text Editors

A **text editor** is an application that allows you to view, create, or modify the contents of text files. It was originally created to write programs, but is now being used even to edit ordinary text files. Text editors work on different modes such as command mode and insert mode. Various types of text editors, such as Vim, gedit, and nano, are compatible with Linux. However, text editors do not always support the formatting options that word processors provide. Text editors may work either in the CLI or GUI.

[illegible]

Figure 4-1: The Vim text editor in insert mode.

The screenshot shows a terminal window titled "root@localhost:~". The menu bar includes File, Edit, View, Terminal, Tabs, and Help. The terminal displays the output of the 'df' command, which lists disk space usage for various filesystems. A red circle highlights the prompt ':q' at the bottom left.

Filesystem	Size	Used	Avail	Mounts
/dev/VolGroup00/LogVol00	ext3	defaults	1 1	
LABEL=/boot	ext3	defaults	1 2	
tmpfs	tmpfs	defaults	0 0	
devpts	devpts	gid=5,mode=620	0 0	
sysfs	sysfs	defaults	0 0	
proc	proc	defaults	0 0	
/dev/VolGroup00/LogVol01	swap	defaults	0 0	

```
:q
```

Figure 4-2: The Vim text editor in command mode.

List of Text Editors

Many text editors are compatible with Linux.

<i>Text Editor</i>	<i>Description</i>
--------------------	--------------------

Text Editor	Description
<i>Vim</i> (<i>Vi</i>)	The Vi IMproved, or VIM, text editor is the default text editor in Linux. It is widely used in programming and for processing simple text files. It is a powerful editor that optimizes speed by employing simple keystrokes to perform complex text editing.
<i>Emacs</i>	A flexible, powerful, and popular text editor used in Linux and UNIX. It offers numerous features such as content-sensitive editing modes and support for various languages. It can be easily customized.
<i>Gvim</i>	The graphical version of the Vim editor.
<i>KWrite</i>	A flexible GUI-based text editor used in KDE.
<i>gedit</i>	A simple yet powerful GUI-based text editor used in the GNOME desktop.
<i>nano</i>	A small, user-friendly text editor that evolved from the Pico text editor.

Vim

Vim, a contraction of Vi IMproved, is an extended version of the vi editor. Vim and vi implement a text-based user interface to advanced text editing, and is favored by many system administrators and software engineers for its efficiency and ability to be extensively customized.

Emacs

Emacs is derived from "Editor MACroS." It was written by Richard Stallman. Emacs may be extensively customized and includes a variety of extensions provided by its support for Emacs Lisp, an embedded macro definition system that supports a dialect of the Lisp programming languages.

KDE

KDE is an alternative GUI desktop for Linux. It provides basic desktop functions, applications, tools, and documentation for developers to write applications for the system.

The vim Command

The vim command invokes the Vim editor. However, the vi command may also be used for this purpose because it automatically redirects the user to Vim. When entered without a file name as an argument, the vim command opens a welcome screen by default. To open a file, the syntax `vim {file name}` is used. If the file does not exist, Vim creates a file by the name specified and opens the file for editing. Vim supports multiple files being opened simultaneously.

0,0-1	All
-------	-----

Working with Multiple Windows

Vim Modes

Some of the common modes in Vim are listed here.

Switch Modes

You can switch from command mode to any other mode by using a single keystroke.

Key	Function
i	Switches to insert mode and inserts text to the left of the cursor.
A	Switches to insert mode and adds text at the end of a line.
I	Switches to insert mode and inserts text at the beginning of a line.
o	Switches to insert mode and inserts text on a new line below the cursor.
O	Switches to insert mode and inserts text on a new line above the cursor.

Key	Function
v	Switches to visual mode to enable selection, one character at a time.
V	Switches to visual mode to enable selection, one line at a time.
:	Switches to execute mode to enable users to enter commands.
Esc	Returns to command mode.

Execute Mode Commands

In command mode, when the colon(:) operator is entered, a small command prompt section appears at the bottom-left of the editor. This indicates that the user is in execute mode and can run commands supported by Vim.

Some commands supported by Vim are listed in the following table.

Command	Function
<code>:w {file name}</code>	Saves a file with a file name if it is being saved for the first time.
<code>:q</code>	Quits when no changes have been made after the last save.
<code>:q!</code>	Quits ignoring the changes made.
<code>:qa</code>	Quits multiple files.
<code>:wq</code>	Saves the current file and exits.
<code>:e!</code>	Reverts to the last saved format without closing the file.
<code>!:any Linux command}</code>	Executes the command and gets the result in the Vim interface.
<code>ZZ</code>	Writes the file only if changes were made and quits the Vim editor.

Vim Help Options

A major source of built-in documentation for Vim can be accessed using the `:help` command. To find topic-specific help, you can add the necessary topic as an argument. To quit the help manual, you can use `:q`. The `vimtutor` command helps first time users learn the basics of Vim by allowing them to practice Vim commands and shortcuts.

```

@localhost:~
File Edit View Search Terminal Help
=====
=  Welcome to the VIM Tutor - Version 1.7  =
=====

Vim is a very powerful editor that has many commands, too many to
explain in a tutor such as this. This tutor is designed to describe
enough of the commands that you will be able to easily use Vim as
an all-purpose editor.

The approximate time required to complete the tutor is 25-30 minutes,
depending upon how much time is spent with experimentation.

ATTENTION:
The commands in the lessons will modify the text. Make a copy of this
file to practise on (if you started "vimtutor" this is already a copy).

It is important to remember that this tutor is set up to teach by
use. That means that you need to execute the commands to learn them
properly. If you only read the text, you will forget the commands!

Now, make sure that your Shift-Lock key is NOT depressed and press
the  j  key enough times to move the cursor so that Lesson 1.1
completely fills the screen.

```

Figure 4-4: Viewing the help file using the Vim text editor.

Motions

Motions are single-key shortcuts that are used to navigate through files in command mode. These keys position the cursor anywhere within a document. They can be used for moving the cursor through characters, words, lines, or even huge blocks of text.

<i>Navigation Key</i>	<i>Used To</i>
h	Move left one character.
j	Move down one line.
k	Move up one line.
l	Move right one character.
^	Move to the beginning of the current line.
\$	Move to the end of the current line.
w	Move to the next word.
b	Move to the previous word.
e	Move to the end of the current word or to the end of the next word if you are already at the end of the word.
Shift+L	Move the cursor to the bottom of the screen.
Shift+H	Move the cursor to the first line of the screen.
(Line number) Shift+G	Move the cursor to the specified line number.
gg	Move the cursor to the first line of the file.
Shift+G	Move the cursor to the last line of the file.

Navigation Using the Arrow Keys

In addition to using the **h**, **j**, **k**, and **l** keys to navigate through the editor, you can also use the **Up**, **Down**, **Left**, and **Right Arrow** keys. The conventional navigation keys such as **Home**, **End**, **Page Up**, and **Page Down** also work in Vim.

Editing Operators

Editing operators in command mode are powerful tools that can be used to manipulate text with simple keystrokes. They can also be used in combination with motions to edit multiple characters.

Some of the frequently used editing operators are listed here.

<i>Editing Operator</i>	<i>Used To</i>
x	Delete the character selected by the cursor.
d	Delete text.
dd	Delete the current line.
p	Paste text on the line directly below the cursor.
P	Paste text on the line directly above the cursor.
/[text string]	Search through the document for specific text.
?[text string]	Search backward through the document for specific text.
y	Yank or copy text.
yy	Copy the line directly above the cursor.
c[range of lines]c	Begin a change in the specified range.

<i>Editing Operator</i>	<i>Used To</i>
u	Undo the latest change.
U	Undo all changes in the current line.



Note: In case any editing was undone by mistake, you can press **Ctrl+R** to redo the latest undone changes.

Case Sensitivity

Most Vim options are case sensitive. For example, the **p** option pastes text you cut on the line directly below the cursor, whereas the **P** option pastes text you cut on the line directly above the cursor.

Counts

A **count** is a number that multiplies the effect of keystrokes in Vim. It can be used in combination with motions or operators or both. When used with a motion, cursor movement is multiplied according to the count specified. When used with editing operators, the action gets repeated the number of times specified.

Syntax

If count, motions, and operators are used together, their syntax is operator *[count] {motion}*.

This makes the cursor move and perform the action as many times as specified by the count.

The diff Command

The **diff** command is used to compare individual text files or contents of directories. The command displays the two files and the differences between them.

The diff command has various options that allow you to specify the nature of the output.

<i>Option</i>	<i>Description</i>
-b	Ignores spacing differences.
-i	Ignores case differences.
-t	Expands tab characters in output lines.
-w	Ignores spacing differences and tabs.
-c	Displays a list of differences with three lines of context. The output displays the identification of the files involved and their creation dates, and each change is separated by a line with a dozen asterisks (*). The lines that are removed from the first file are marked with hyphens (-); those that are added to the second file are marked with the plus sign (+). Lines that are shifted from one file to the other are marked in both the files with exclamation points (!).

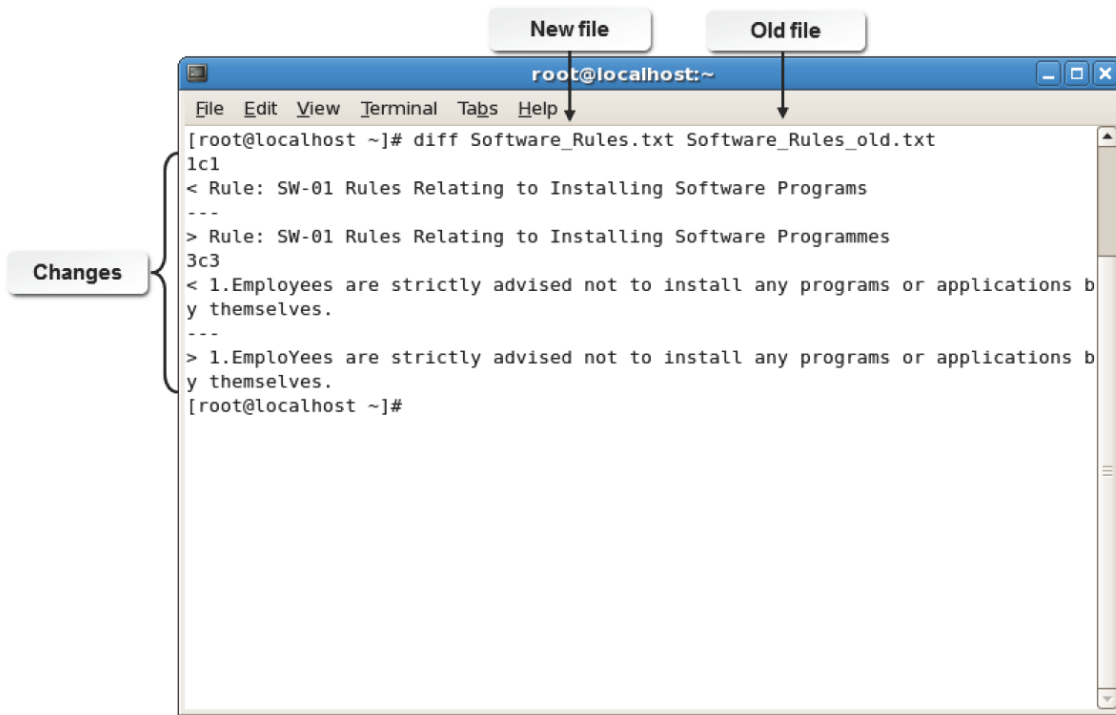


Figure 4-5: Comparing two versions of the same document with the diff command to track changes.

Syntax

The syntax of the diff command is `diff {file name 1} {file name 2}`.

The patch Command

The **patch** command updates text files with changes according to instructions contained in a patch file. This patch file contains listings produced by the diff command.

Comparing Text Files

The vimdiff command allows you to compare two text files. The syntax of this command is similar to the syntax of the diff command.

The wc Command

The **word count (wc)** command is used to count the number of lines, words, and characters of text files. If multiple files are specified, then the command displays the count for each file and the total count for all files.

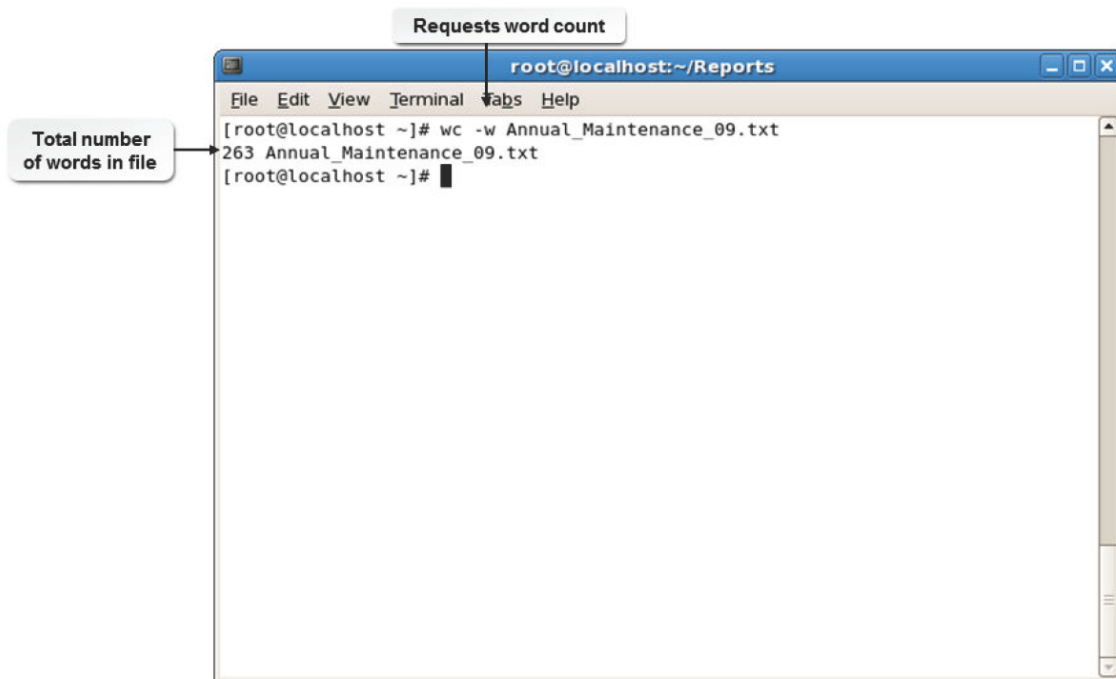


Figure 4-6: Counting words using the `wc` command.

The `wc` command provides various options that allow you to specify the nature of the output.

<i>Option</i>	<i>Description</i>
-c	Displays the byte count.
-m	Displays the character count.
-l	Displays the newline count.
-w	Displays the word count.

Syntax

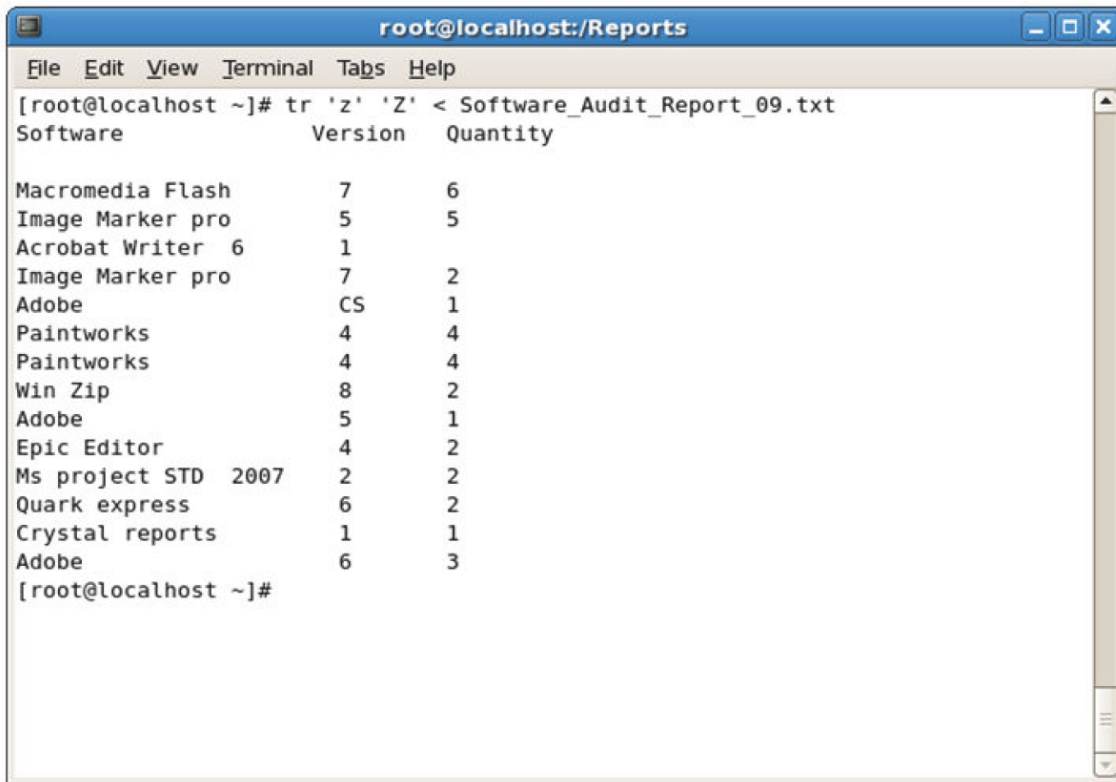
The syntax of the `wc` command is `wc [options] {file name}`.

The `aspell` Utility

[*aspell*](#) is a utility that functions as a spell checker in Linux. The syntax of the `aspell` command is `aspell [options]`. The `-c` option checks the file for incorrect spellings. The `-l` option produces a list of misspelled words from the standard input.

The `tr` Command

The [*translate \(tr\)*](#) command is used to translate strings from the standard input to the standard output. It is predominantly used to change the case of letters. This command acts only on a stream of characters and does not accept file names as arguments.



```
root@localhost:/Reports
File Edit View Terminal Tabs Help
[root@localhost ~]# tr 'z' 'Z' < Software_Audit_Report_09.txt
Software          Version    Quantity
Macromedia Flash      7         6
Image Marker pro      5         5
Acrobat Writer 6      1
Image Marker pro      7         2
Adobe                CS         1
Paintworks            4         4
Paintworks            4         4
Win Zip               8         2
Adobe                5         1
Epic Editor           4         2
Ms project STD 2007   2         2
Quark express         6         2
Crystal reports       1         1
Adobe                6         3
[root@localhost ~]#
```

Figure 4-7: Editing text using the tr command.

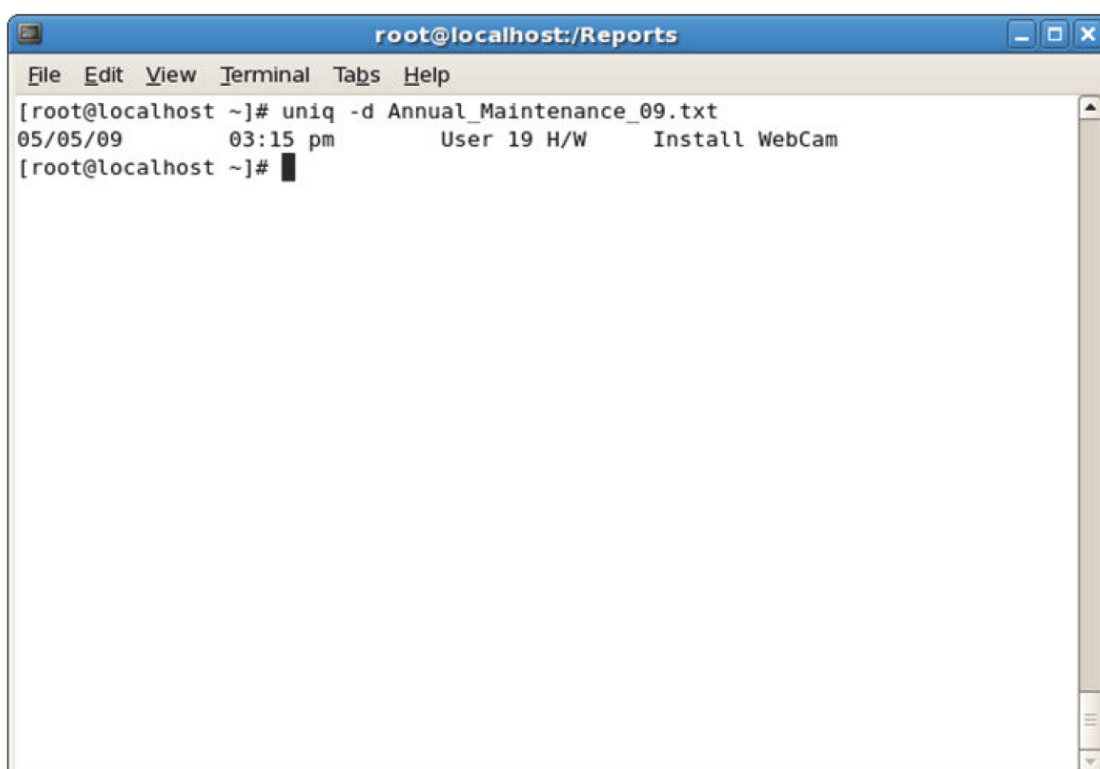
Syntax

The syntax of the tr command is `tr {character 1} {character 2} < {file name}`,

where character 1 is the character to be replaced.

The uniq Command

The [uniq](#) command is used to display unique lines from a sorted file after ignoring successive duplicated lines. Because it compares only consecutive lines, the uniq command requires sorted input.



```
root@localhost:/Reports
File Edit View Terminal Tabs Help
[root@localhost ~]# uniq -d Annual_Maintenance_09.txt
05/05/09      03:15 pm      User 19 H/W      Install WebCam
[root@localhost ~]#
```

Figure 4-8: Duplicate content displayed using the uniq command.

The uniq command provides various options that allow you to specify the nature of the output.

Option	Description
-u	Displays only unique lines.
-d	Displays only duplicated lines.
-c	Displays lines prefixed by the number of occurrences.

Syntax

The syntax of the uniq command is `uniq [options] {file name}`.

Input and Output Redirection

When you want to redirect the contents of an existing file to another command for processing, the input redirection symbol, less than (<), and the output redirection symbol, greater than (>), can be used. The output redirection symbol tells the shell to redefine standard output as a file. If the file does not exist, the shell creates it. The input redirection symbol tells the shell to redefine standard input as something other than the keyboard input, usually a file.

For example, the `ls > list` command causes the shell to send the output of the ls command to a file named *list*.

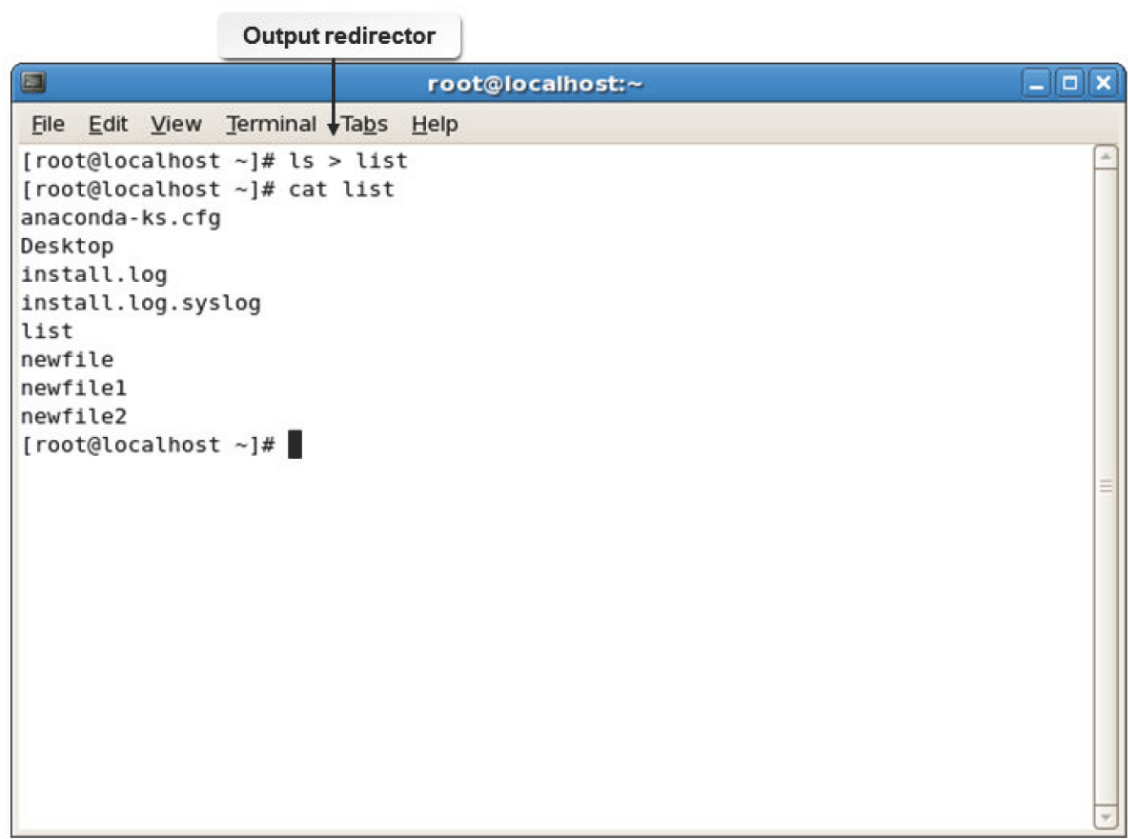


Figure 4-9: Redirecting the output of the ls command to the list file.

How to Create and Edit Text Files

Follow these general procedures to create and edit text files.

Create a File and Enter Text Using the Vim Editor

To create a file and enter text using the Vim editor:

1. Log in as a user.
2. To create a file, at the command prompt, enter `vim {file name}`.
3. To switch to insert mode, press `i`.
4. Type the required content.
5. To return to command mode, press **Esc**.
6. To save and close the file, enter `:wq`.

Create a Text File from the Command Prompt

To create a text file from the command prompt:

1. Log in as a user.
2. In the CLI terminal, navigate to the directory where you want to create the file. If necessary, create a new directory at the desired location and make it the current directory.
3. At the command prompt, enter `cat > {file name}`.
4. To move to a new line, type the contents of the file and press **Enter**.
5. To save the file and return to the command prompt, press **Ctrl+D**.
6. If necessary, to view the file contents, type `cat {file name}`.

Edit Text Files in Vim Command Mode

To edit text files in Vim command mode:

1. Log in as a user.
2. To open a file, enter `vim {file name}`.
3. To make necessary changes, use the appropriate vim shortcuts.
4. To save and close the file, enter `:wq`.

Open Multiple Windows Using the vim Command

To open multiple windows using the vim command:

1. Log in as a user.
2. Open multiple windows.
 - To open different files in multiple windows, enter `vim -o {file name 1} {file name 2} ... {file name n}`.
 - To open a new file in a new window, press **Ctrl+W+N**.
 - To navigate through the windows, hold down **Ctrl+W** and use the arrow keys.
3. To make necessary changes, use the appropriate Vim shortcuts.

4. If necessary, to return to command mode, press **Esc**.
5. Save and close the files.
 - To save and close the files one by one, enter :wq.
 - To close all files at the same time, enter :qa.

Count the Words in a File

To count the words in a file:

1. Log in as a user.
2. Count the words in a file.
 - To count the number of words, lines, bytes, and characters in the file, enter `wc [options] {file name}`.
 - To count the number of words, lines, bytes, and characters in the output of the command, enter `{command} | wc [options]`.

Remove Duplicate and Adjacent Lines in a File

To remove duplicate and adjacent lines in a file:

1. Log in as a user in the CLI.
2. Remove duplicate and adjacent lines in a file.
 - To remove duplicate and adjacent lines from a file, enter `uniq [options] {file name}`.
 - To remove the duplicate and adjacent lines from the output of the command, enter `{command} | uniq [options]`.

Compare Files in the CLI

To compare files in the CLI:

1. Log in as a user in the CLI.
2. To compare files for differences in their content, enter `diff {file name 1} {file name 2}`.

Compare Files in the GUI

To compare files in the GUI:

1. To switch to the GUI, press **Ctrl+Alt+F1**.
2. From the menu bar, select **Application → Utilities → Terminal**.
3. To compare files for differences in content, enter `vimdiff {file name 1} {file name 2}`.
4. To return to the terminal, enter :q two times.

Replace Characters

To replace characters:

1. Log in as a user in the CLI.
2. Replace characters.
 - To replace one character with another one and display the contents of the file, enter `tr '{character 1}' '{character 2}' < {file name}`.
 - To replace one character with another one from the output of the command, enter `{command} | tr '{character1}' '{character 2}'`.

TOPIC B Locate Files

In the last topic, you created and edited files. As a user or administrator, you will have to frequently locate files within the Linux filesystem before you can edit them. In this topic, you will locate files within the Linux system.

Learning how to quickly locate files within Linux will reduce the amount of time you spend searching for files. There are different techniques available for locating files within Linux, and these techniques will save you time and effort as you manage larger filesystems.

The locate Command

The **locate** command performs a quick search for any specified string in file names and paths stored in the *mlocate* database. This database must be updated regularly for the search to be effective. The results displayed may be restricted to files that users have permissions to access or execute.

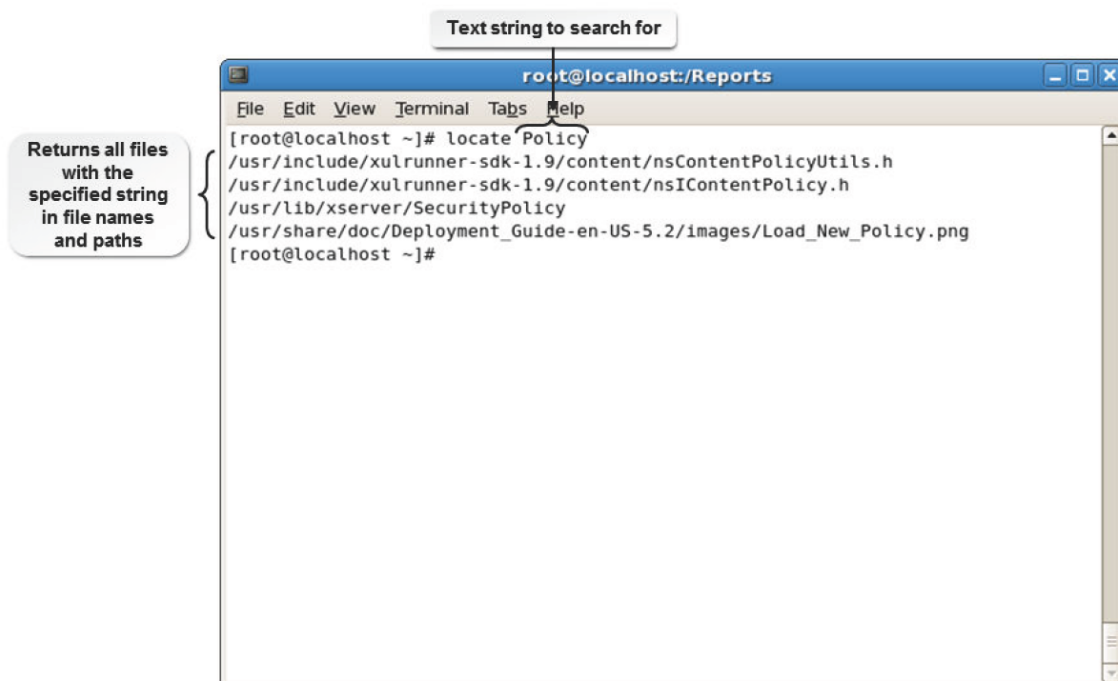


Figure 4-10: Searching files using the locate command.

Syntax

The syntax of the locate command is `locate [options] {string}`.

The locate Command Options

The locate command supports different options that enable you to make your search more effective. Some of the options are described in the table.

<i>Option</i>	<i>Description</i>
-r	Uses regular expressions in searching for file names.
-c	Displays only the number of matching entries found, rather than the file names.
-e	Returns only files that exist at the time of search.
-i	Ignores the casing in file names or paths.
-n number of entries	Returns only the first few matches up to the specified number.

Updating the mlocate Database


The updatedb command is used to build a database of files based on the **/etc/updatedb.conf** file.

This command is used to update the **/var/lib/mlocate/mlocate.db** database. The **/etc/ updatedb.conf** file consists of the paths that should be excluded while building the database. To add a path that needs to be excluded while building the database, open the **/etc/ updatedb.conf** file and, in the PRUNEPATH variable, specify the path that need not be included while building the database. For example, PRUNEPATH="/etc" will exclude the **/etc** directory while building the database.

Though this is the default database searched by the locate command, there may be more databases containing file paths. If the database is not updated before performing a search, all files created after the last update will be excluded from the search.

Using grep

In its simplest form, grep is a search tool. It allows you to perform search actions, such as finding any instance you are searching for, in a file. For example, entering `grep foo test` returns all the lines that have a string matching "foo" in the file "test." The grep command can also be used to search a directory for a certain file. The `ls -l | grep audit` command returns a long listing of any files in the current directory whose name contains "audit."

	Note: The term grep is derived from "Globally search a Regular Expression and Print"
---	---

The whereis Command

The **whereis** command is used to view various details associated with a command. The whereis command has various options.

<i>Option</i>	<i>Used To</i>
-b	Search only for binaries.
-m	Search only for manual sections.
-s	Search only for sources.
-u	Search for unusual entries.

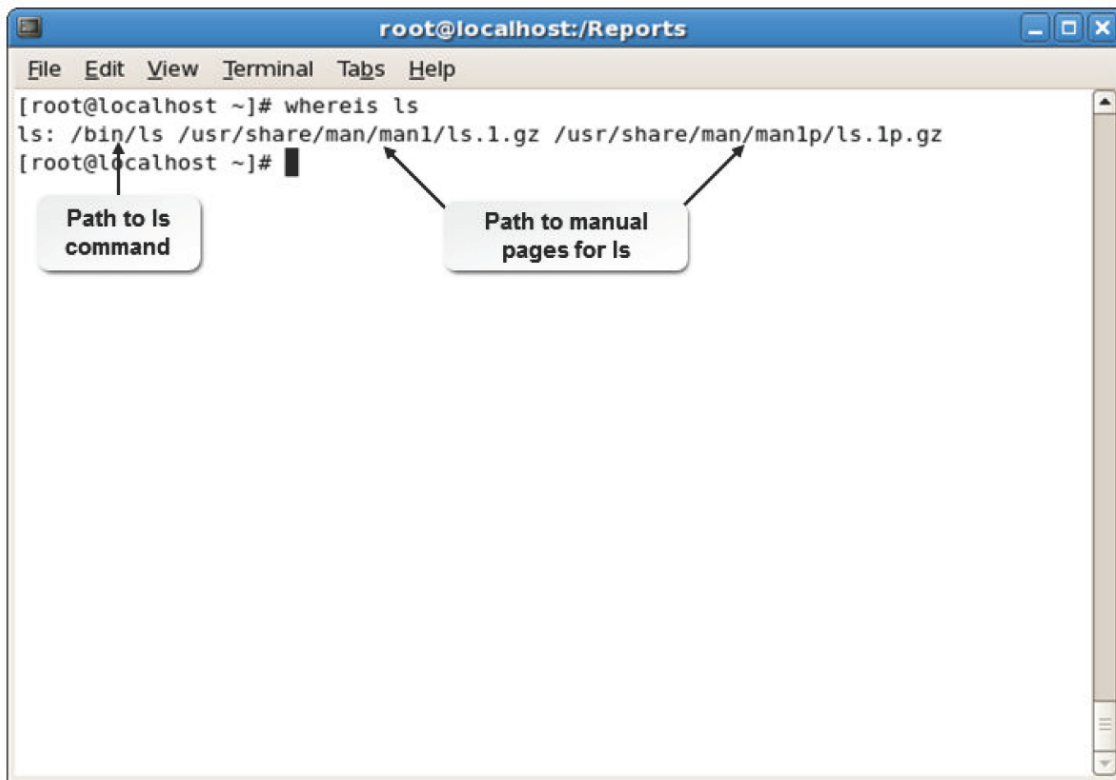


Figure 4-11: The path to the ls command.

Syntax

The syntax of the whereis command is `whereis [options] [directory] {file name}`.

Using the whereis Command

On entering whereis ls, the following output is displayed: `ls: /bin/ls /usr/share/man/man1/ls.1.gz /usr/share/man/man1p/ls.1p.gz`.

Where `/bin/ls` indicates the location of the ls command and `/usr/share/man/man1/ls.1.gz`

`1.gz /usr/share/man/man1p/ls.1p.gz` indicates the location of the man pages for the ls command.

The find Command

The [find](#) command enables you to search a specific location for files and directories that adhere to some search criteria. It recursively searches the directory structure, including any subdirectories and their contents, beginning with the search location you enter. You can perform one or more actions on the files found.

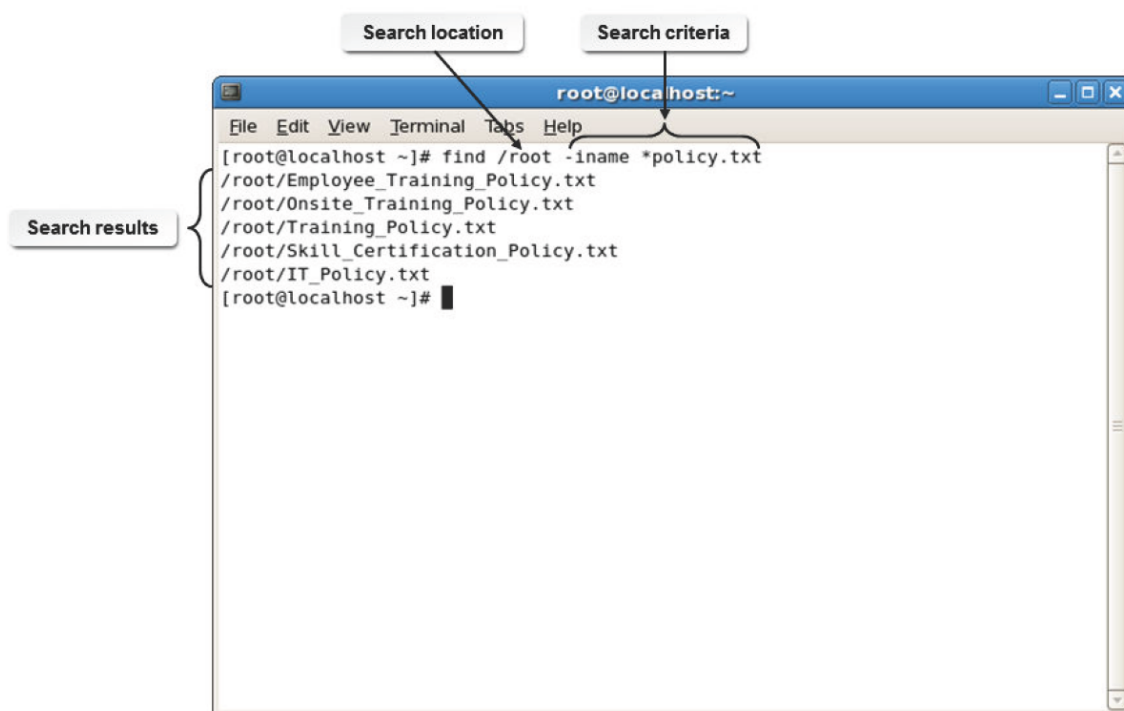


Figure 4-12: Searching files using the find command.

Syntax

The syntax of the find command is `find [options] {search locations} {search criteria} [actions]`.

find vs. locate Commands

The locate command searches a database and retrieves information on files present on your system. However, failure to keep this database updated may produce outdated results. The find command, on the other hand, performs a live search of the filesystem and may concentrate on a specific location. The find command may take more time to complete a search than the locate command.

Working of the find Command

You can use the find command to search the entire directory structure for a file even if you remember a portion of the file name. One or more search paths can be designated and directory notations can be used as the search path. If no directory is specified, the find command uses the current working directory as the location to start the search. One or more criteria can be used to specify the conditions of a file or directory. In case of more than one search criterion, the file or directory must meet all the conditions specified before the results are displayed. The results displayed may be restricted to files users have permissions to access or execute. You can check the manual pages of the find command for more options.

Options for Files Found

When the system finds a listing that meets your criteria, there are several actions that can be performed on the results. Several of these options are outlined in the following table.

<i>Option</i>	<i>Action Performed</i>
-print	Displays the location of the files found.
-exec	Executes the command that follows.
-ok	Executes the command that follows interactively.
-delete	Deletes files found.

<i>Option</i>	<i>Action Performed</i>
-fprint	Stores results in the target file.

find Command Conditions

The find command can be used with one or more conditions. These conditions accept strings or numbers as arguments.



Figure 4-13: Using the `-name` condition in the `find` command.

Some of the frequently used conditions are listed in the following table.

<i>Condition</i>	<i>Description</i>
-name	Matches by name. Regular expressions may be used as arguments.
-iname	Matches by name, ignoring the case.
-user	Matches by user name or UID of the owner.
-group	Matches by group or GID of the owner group.
-size	Matches by size.
-perm	Matches by symbolic or octal permissions.
-type	Matches by file type.
-newer	Matches by comparing the modification time. Returns files modified later than reference files.
-atime	Matches by access time in days.
-mtime	Matches by modification time in days.
-ctime	Matches by time of latest changes in a file. Arguments are counted in days.

Logical Operators for Conditions

You can combine conditions using logical operators. When more than one criterion is specified, by default, only those files that satisfy all conditions are returned. However, the logical operator OR can be applied to these conditions by using -o between conditions. Also, the NOT operator can be applied to conditions by using -not to negate a condition.

Numeric Arguments

Numeric arguments are used to specify a numeric value. The following table lists the numeric arguments for size and their description.

<i>Use This Argument</i>	<i>If You Need To</i>
<i>n</i>	List files that are equal to n units (default option).
<i>+n</i>	List files that are greater than n units.
<i>-n</i>	List files that are less than n units.

The following table lists the numeric arguments for time and their description.

<i>Use This Argument</i>	<i>If You Need To</i>
<i>n</i>	List files that were accessed n days ago (default option).
<i>+n</i>	List files that were accessed more than n days ago.
<i>-n</i>	List files that were accessed less than n days ago.

How to Locate Files

Follow these general procedures to locate files.

Search for Files from the Database

To search for files from the database:

1. Log in as **root**.
2. To update the **mlocate** database with the file name and path information, enter `updatedb`.
3. To search the updated database for the specified string, enter `locate {string}`.

Search for Files Using `find`

To search for files using **find**:

1. Log in as **root**.
2. To open the search tool, choose **Places** → **Search for Files**.
3. To search for files with the suffix `.conf` in the **/etc** directory, enter `find /etc -name *.conf`.
4. To search for all files that have been modified in the past day, enter `find / -mtime -1`.
5. To search for all `.conf` files in the **/etc** directory modified this week, enter `find /etc -name *.conf -mtime -7`.

TOPIC C Search Text Using Regular Expressions

In the last topic, you located files within the Linux system. In Linux, you can search text files by specifying a particular portion of text or even just some characters of the file. You can then make the desired modifications to these characters. In this topic, you will search text files to locate text and characters.

Consider that you have created a new text file and that there are pages of content within it. Now you discover that you need to modify a particular word. Going through lines and lines of text to locate one word is a daunting task. Using the plain text search option will also be time consuming. It would be much easier if you can locate a particular word with a few simple keystrokes, and this is where regular expressions can help.

Regular Expressions

Regular expressions are strings of characters that form a pattern for searching another string, often a word, a set of words, or a sentence. Finding and replacing text and manipulating strings are the main uses of regular expressions.

To find all of the lines in the `/etc/services` file that contain either "apple", "microsoft", or "ibm", `egrep "apple|microsoft|ibm" /etc/services` is used. The regular expression in this command is `apple|microsoft|ibm` and specifies that any of those three words may match the text string.

A regular expression with notational elements is a search string formed by combining wildcards, numbers, and characters.

For example, `[^e]?b[1-9]` as a whole is called a regular expression with notational elements, where the notational elements are `^`, `?`, and `[1-9]`. This expression searches for a **file/word/** directory that starts with the letter "e" followed by a character/number, then by the letter "b", and finally by a number ranging between 1 and 9.

Another example of a regular expression with notational elements is `1{5}`, where `{` and `}` are the notational elements. This expression searches for the occurrence of the number "1" repeated consecutively five times. Regular expression is often referred to as `regex`.

Some examples of regular expressions are listed in the following table.

<i>Regular Expression</i>	<i>Description</i>
<code>.*</code>	Zero or more of any number of characters in a row.
<code>.+</code>	One or more of any number of characters in a row.
<code>\d</code>	A metacharacter that specifies any numeric digit (0-9).
<code>\w</code>	A metacharacter that specifies any word character (a-z, A-Z, 0-9, including underscore).
<code>[a-zA-Z0-9_]</code>	A character class that equals the <code>\w</code> regular expression metacharacter (a-z, A-Z, 0-9, including underscore).
<code>^\w{8-16}\$</code>	Matches any word or combination of alphanumeric characters, 8-16 characters in length.
<code>^\d{5}\$</code>	A 5-digit number (i.e., a ZIP code).
<code>^http:\w.*</code>	An HTTP web address (URL).
<code>^http(s?):\w.*</code>	An HTTP or HTTPS web address (URL).

Expressions

Expressions are a group of characters. They are formed by combining variables and constants with operators. They are used in `if` and `while` statements. Performing arithmetic comparisons, string comparisons, and testing files are the main functions of expressions. If an expression contains the `<`, `>`, `&`, or `|` symbols, parentheses are required.

How to Search Text Using Regular Expressions

Follow these general procedures to search text using regular expressions.

Search Using Grep

To search using the regular expression tool:

1. Log in as a user.
2. Search using the regular expression tool.
 - To search through a filesystem using the regular expression tool, enter `ls -l | grep -{options} {regular expression with notational elements}`.
 - To search through file content using the regular expression tool, enter `grep -{options} {regular expression with notational elements} {file name}`.

TOPIC D Apply Filters to Text Streams

In the previous topic, you used regular expressions to locate text in files. In the course of your work, it will prove helpful if you can break the entire text into logical groups that match certain patterns.

This will be useful when you want to make changes that affect some, but not all, data. In this topic, you will apply filters to text streams to break them into sections that match specific criteria.

Imagine a scenario where a merger takes place between two organizations. Merging and sorting select policy documents and employee databases can be a tedious task for an administrator.

Applying filters makes the task of merging these documents and sorting the employee database an easy one.

Filters

A **filter** is a program that accepts an input or output request, verifies what data matches the criterion specified in the request, and then processes it. Filters are shell scripts and are sometimes used to insert or remove headers.



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# cat newfile  
1  
9  
4  
7  
2  
7  
8  
0  
[root@localhost ~]# sort -n newfile  
0  
1  
2  
4  
7  
7  
8  
9  
[root@localhost ~]#
```

Figure 4-14: Sorted output of a file.

Text Streams

A **text stream** is a sequence of one or more lines of text that can be written to be read on a text-based display. While reading from or writing to a text stream, the program divides the data into lines by reading an NL, or newline, at the end of each line.

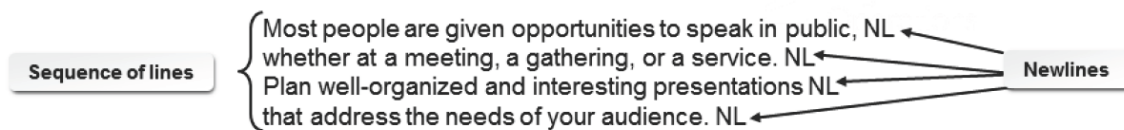


Figure 4-15: Sequence of lines that use the NL character.

Limitations of Text Streams

The position of the text stream should not be changed while reading from or writing to it.

Depending on the kind of text file being used, text streams may not support all character values.

The cut Command Options

The cut command cuts out the selected columns or fields. Common cut command options and their uses are given in the following table.

<i>If You Need To</i>	<i>Use This cut Command Option</i>
Delimit one field from another field.	-d={delimiter}
Suppress a line if the delimiter is not found.	-s
Specify the field number. For example, f2 indicates the second field.	-f={column number / column list}

Delimiter

A delimiter can be a tab, space, colon, semicolon, period, or comma used to separate one field from another.

The sort Command Options

The sort command arranges the lines in a file. Common sort command options and their uses are given in the following table.

<i>If You Need To</i>	<i>Use This sort Command Option</i>
Specify field values. For example, -k2 indicates the second field.	-k={column number}
Compare and sort lines based on the string numerical value.	-n
Sort fields in descending order. Note: By default the fields are sorted in ascending order.	-r
Separate one field from another.	-t={delimiter}

Textutil Commands

Textutil commands are used to modify an output. Some common textutil commands and their description are given in the following table.

<i>If You Need To</i>	<i>Use This Textutil Command</i>
Convert tabs in a file to appropriate number of spaces.	expand {file name}

<i>If You Need To</i>	<i>Use This Textutil Command</i>
Format text to a specified width by filling empty lines for the specified file name.	<code>fmt {file name}</code>
Display the first 10 lines of a file.	<code>head {file name}</code>
Count the number of lines in a file.	<code>nl {file name}</code>
Dump the specified files in octal format.	<code>od {file name}</code>
Merge lines of one or more files.	<code>paste {one or more files}</code>
Convert a text file to print.	<code>pr {file name}</code>
Split a file into equally sized pieces.	<code>split {file name}</code>
Display files in reverse to the standard output.	<code>tac {file name}</code>
Display the last 10 lines of a file.	<code>tail {file name}</code>
Translate characters from one format to another and to the standard output.	<code>tr {one format} {another format}</code>
Convert white spaces to appropriate number of tabs for the specified file name.	<code>unexpand {file name}</code>
Delete duplicate adjacent lines from a sorted file.	<code>uniq {file name}</code>
Print the byte, word, and line counts of the specified file name.	<code>wc {file name}</code>

How to Apply Filters to Text Streams

Follow these general procedures to apply filters to text streams.

Modify Output Using the join Command

To modify the output of text files using the join command:

1. If necessary, navigate to the relevant directory.
2. To join the two files by their first column content followed by the rest of the columns, provided the first columns of the files are identical on the standard output, enter `join {file name 1} {file name 2}`.
3. If necessary, to join the two files and redirect the output to a file, enter `join {file name 1} {file name 2} > {file name}`.



Note: While using the redirect command, you may generate additional files. For example, when you use multiple textutil commands to generate the desired output, you may redirect the content from one file to another, thereby generating additional files. You may want to delete these unwanted files to avoid confusion.

Send Output Streams of the cut Command Through the sort Command

To modify output by directing the output stream of the cut command through the sort command:

1. If necessary, navigate to the relevant directory.
2. To send the output streams of the cut command through the sort command, enter `cut {options}{delimiter} - {field number}{file name} | sort {options}`.



TOPIC E Link Files

You know how to locate files within the Linux system. Creating a link or shortcut to those files will enable you to locate them easily. In this topic, you will link files in Linux.

Linking files within Linux will help you track frequently used files without having to navigate through the file structure to search for them each time. You can help users who are not familiar with Linux to access related files by linking the files.

Inodes

An *index node (inode)* is a computer's reference for a file. The *index node table*, or *inode table*, is a data structure that contains information about individual files in a filesystem. Inode is an entry in the table that contains information about the device where the inode resides, the file type, the mode of file, and the UID and GID of the owner. It also contains information about the number of links to the file, the number of bytes in the file, the time of access and modifications, the time when the inode itself was last modified, and the addresses of the file's blocks on the hard disk. The `ls -li` command is used to locate the inode number of a file.

	Note: In the debugfs utility interface, you can use the <code>ffi</code> command to find free inodes.
	Note: Based on permissions, file modes can be writable, readable, or executable. The <code>chmod</code> command allows you to modify permissions for a file.

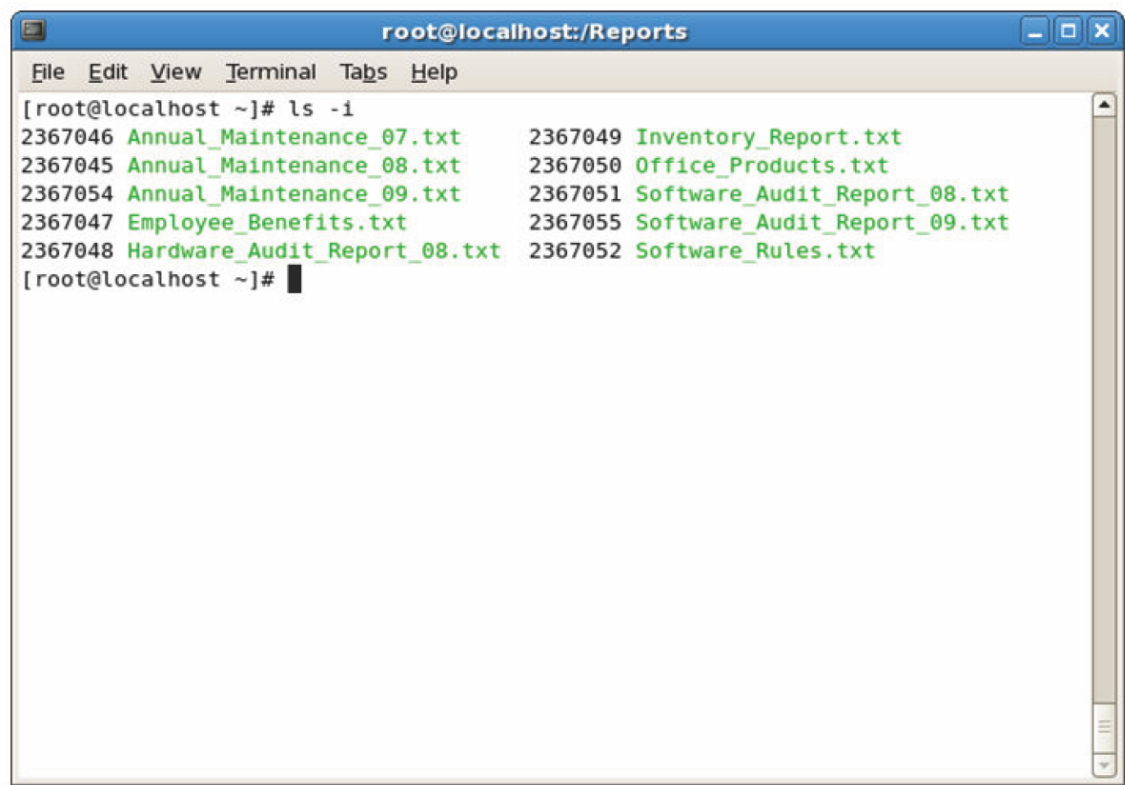


Figure 4-16: Files listed with inode numbers.

The In Command

The *ln* command is used to create a link to a file. A link allows a file name in one directory to point to a file in another directory. A link does not contain data of its own, only a reference to another file. Any changes to the link will reflect in the original file.

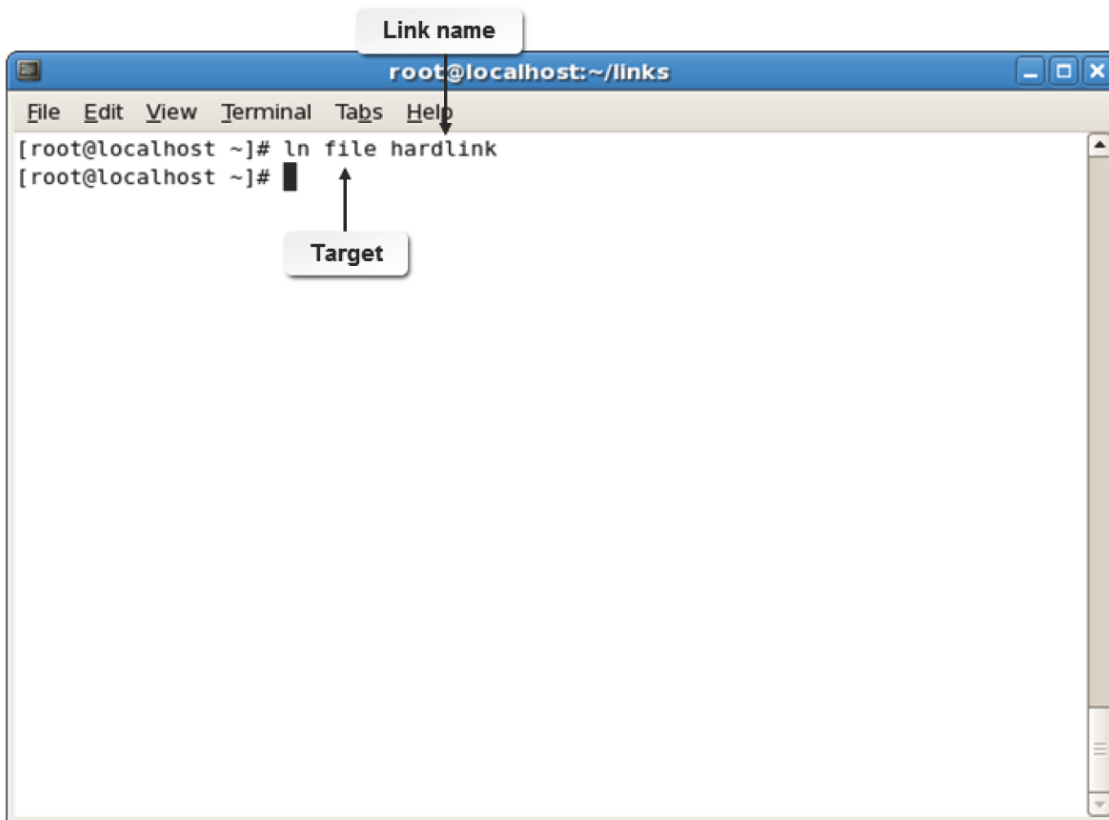


Figure 4-17: A link created using the `ln` command.

Syntax

The syntax of the `ln` command is `ln [option] {target} {link name}`.

In Command Options

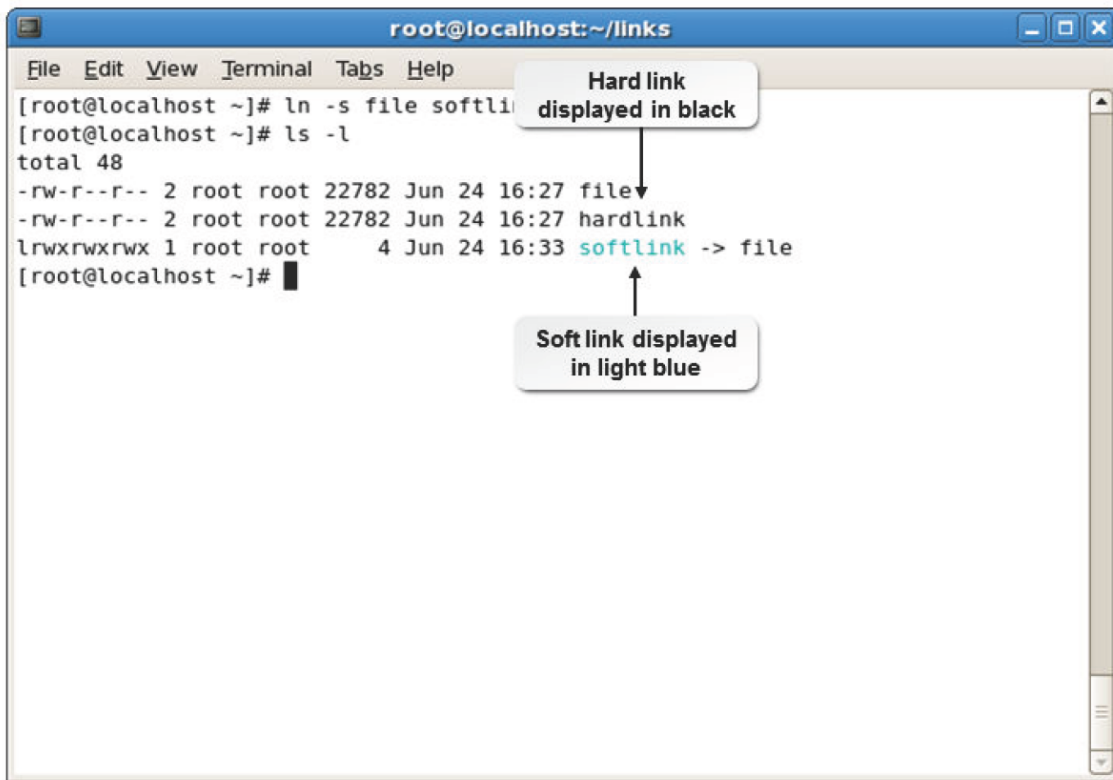
The `ln` command has various options. Some of the frequently used options are given in the following table.

Option	Used To
--backup	Back up existing destination files.
-f	Remove existing destination files.
-s	Make symbolic links instead of hard links.
-i	Prompt to remove destination files.
-v	Print the name of a file before linking.

Types of Links

Using the `ln` command, you can create two types of links: hard and symbolic (soft).

Link	Description
Hard	<p>A hard link is a reference to another file; it allows the file's data to have more than one name in different locations in the same filesystem.</p> <p>Applications treat a hard link as a real file. If the original file is deleted after a hard link is created, all its contents will still be available in the linked file. Hard links cannot be created between two directories, nor can they be created between two files in different filesystems.</p>
Symbolic	<p>A symbolic link is a reference to a file or directory that allows you to access mounted filesystems from a different directory. Unlike hard links, symbolic links can be created between two filesystems. If the original file is deleted after a symbolic link is created, then the original content is lost.</p> <p>A symbolic link is also known as a soft link.</p>



A terminal window titled 'root@localhost:~/links' showing the following commands and output:

```
[root@localhost ~]# ln -s file softlink
[root@localhost ~]# ls -l
total 48
-rw-r--r-- 2 root root 22782 Jun 24 16:27 file
-rw-r--r-- 2 root root 22782 Jun 24 16:27 hardlink
lrwxrwxrwx 1 root root 4 Jun 24 16:33 softlink -> file
[root@localhost ~]#
```

Annotations:

- A callout box labeled "Hard link displayed in black" points to the 'hardlink' entry in the output.
- A callout box labeled "Soft link displayed in light blue" points to the 'softlink' entry in the output.

Figure 4-18: A hard link and symbolic link created for the same file.



Note: Hard and symbolic links are a feature of the filesystem and are common in filesystems of most UNIX and UNIX-like operating systems such as Linux. The ext2, ext3, ext4, and XFS filesystems all support hard and symbolic links.

How to Link Files

Follow these general procedures to link files.

View the Inode Number of a File or Directory

To view the inode number of a file or directory:

1. Log in as a user.
2. To view the inode details, enter `ls -li {file or directory name}`.

Link Files

To link files:

1. Log in as a user.
2. Create file links.
 - To create a hard link, enter `ln {source file}{destination file}`.
 - To create a soft link, enter `ln -s {source file}{destination file}`.
3. To view the inodes of the file, enter `ls -li {file name}`.

Copy Files Through Links

To copy a file through a link:

1. Log in to the CLI as **root** and navigate to the relevant directory.
2. To create a symbolic link, enter `ln -s {file/directory name}{link name}`.



Note: A symbolic link is indicated by the name of the link (often displayed in light blue) and an arrow pointing to the source file.

3. To copy the file through a symbolic link, enter `cp {link name}{target name}`.
4. If necessary, to check if the file has been copied, enter `ls -l`.

Use Linked Files to Support System Administration Tasks

To create links to files that support system administration tasks:

1. Log in to the CLI as **root** and navigate to the relevant directory.
2. To create a symbolic link for a system task, enter `ln -s {system file}{link name with path}`.
3. If necessary, to execute the system file, enter `{link name}`.

TOPIC F Back Up and Restore Files

In the previous topics, you created, edited, located, and linked files. It is essential that you also know how to back up and restore these files when the need arises. In this topic, you will back up and restore files.

Learning how to back up and restore files will save you countless hours of repairing your system after a system failure. Backing up and restoring files allow you to keep an additional copy of files on your system because they existed at a specific point in time. If you ever have a system failure, these files can be used to restore your system.

Archiving

Archiving is a method of storing data by copying data from a system disk drive into a backup device. This is done to preserve a record of the data for future reference or to create data dumps. In the event of a network disruption resulting in data loss, the data can be retrieved from archives.

Built-in tool for archiving operations

```

root@localhost:~
File Edit View Termina Tabs Help
[root@localhost ~]# dump -0a -f /dev/st0 /usr/src
DUMP: Date of this level 0 dump: Sun Jun  6 17:01:12 2010
DUMP: Dumping /dev/mapper/VolGroup00-LogVol00 (/ (dir usr/src)) to /dev/st0
DUMP: Label: none
DUMP: Writing 10 Kilobyte records
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 36684 blocks.
DUMP: Volume 1 started with block 1 at: Sun Jun  6 17:01:13 2010
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing /dev/st0
DUMP: Volume 1 completed at: Sun Jun  6 17:01:24 2010
DUMP: Volume 1 70690 blocks (69.03MB)
DUMP: Volume 1 took 0:00:11
DUMP: Volume 1 transfer rate: 6426 kB/s
DUMP: 70690 blocks (69.03MB) on 1 volume(s)
DUMP: finished in 11 seconds, throughput 6426 kBytes/sec
DUMP: Date of this level 0 dump: Sun Jun  6 17:01:12 2010
DUMP: Date this dump completed: Sun Jun  6 17:01:24 2010
DUMP: Average transfer rate: 6426 kB/s
DUMP: DUMP IS DONE
[root@localhost ~]#

```

Figure 4-19: Archiving files using the dump command.

The cpio Command

The **cpio** command copies files to and from archives. It is included in standard Linux distributions.

The cpio command has three operating modes.

Operating Mode	Command	Description
Copy-out	cpio -o	In this mode, the command copies files into an archive. It reads the standard input to obtain a list of file names and then copies those files to the standard output.
Copy-in	cpio -i	In this mode, the command copies files from an archive. It extracts files from the standard input.
Copy-pass	cpio -p	In this mode, the command copies files from one directory tree to another. It reads the standard input to obtain the list of file names that are created and copied into the destination directory.

Restoring Files with the cpio Command

The main reason you need to back up data is so that you can retrieve the data if the file gets corrupted or deleted. If you use the cpio command to move files to another location, you will need to get the files out of the archive file so that you can use them. The format of the copy-in option is

cpio -icdv *[archive_file name]*.

The dd Command

The **dd** command copies and converts files to enable them to be transferred from one type of media to another. The dd command has various options.

<i>Option</i>	<i>Used To</i>
if={file name}	Specify the file from which data will be read.
of={file name}	Specify the file to which data will be written.
bs={number of bytes per block}	Specify the number of bytes at which data is read from an input file and written to an output file.
count={number of blocks}	Specify the number of blocks to be written to the output file from the input file.



Note: A selected input file is copied to a selected output file. If no files are selected, the standard input and the standard output are used.

Syntax

The syntax of the dd command is dd *[operand]...* or dd *[option]*.

The dump Command

The **dump** command dumps all files in a filesystem into a tape or another file. It can also be used to dump files modified after a specified date. The dump command has various options.

Some of the common dump command options are provided in the following table.

<i>Option</i>	<i>Used To</i>
-0	Make a full backup.
-1 to 9	Make incremental or partial backups.
-b {maximum block size}	Specify the number of kilobytes per dump record.
-f {location of the target file}	Specify the target location.
-z{compression level}	Specify the compression level in the range 1 to 9.

Syntax

The syntax of the dump command is dump *{-level #} -f {file} {filesystem/file/ directory}*.

The tar Command

The **tar** command allows you to create archives of data. You can use the command on previously created archives to extract files, store additional files, update files, and list files that were already stored. Files archives made with tar frequently have the .tar file extension. The tar command can also direct its output to available devices, files, or other programs using pipes.



Note: tar is derived from Tape ARchive.



Note: Archives made with tar are frequently compressed with gzip (resulting in the file extension .tar.gz) or bzip2 (resulting in the file extension .tar.bz2).

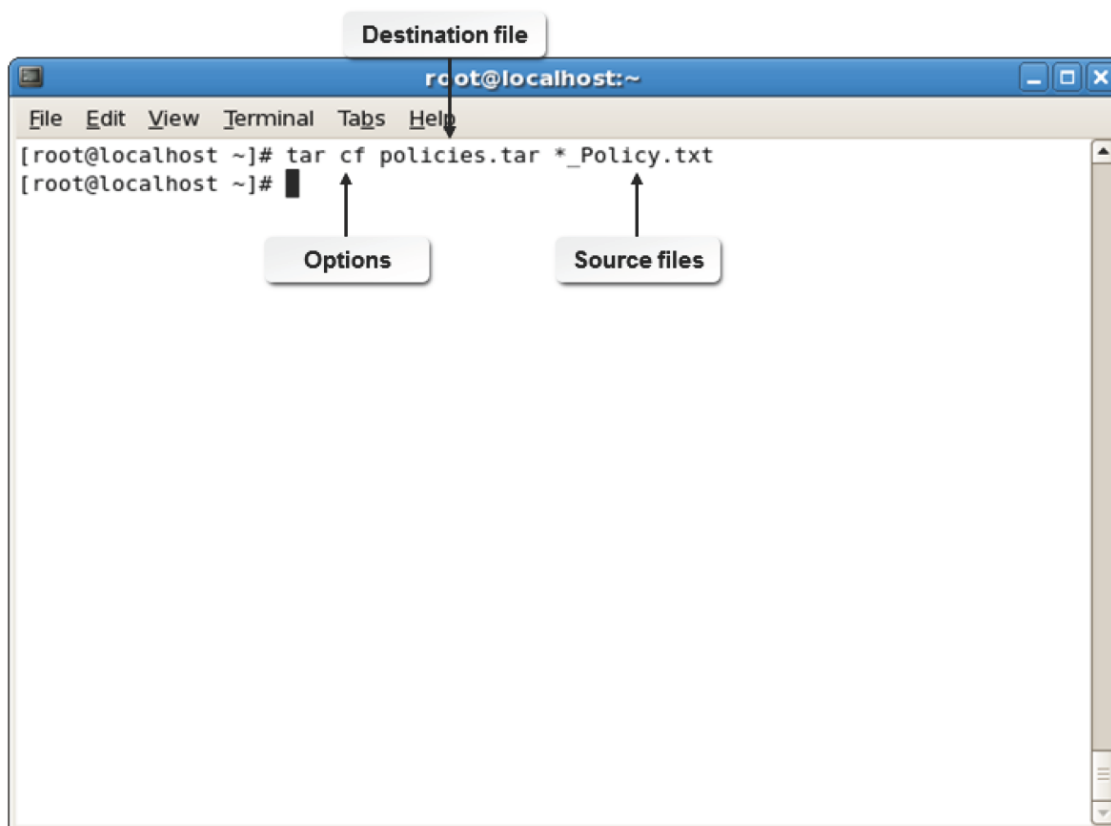


Figure 4-20: Archiving files using the tar command.

Syntax

The syntax of the tar command is `tar [options] {files | directories}`.



Note: Note that for historical compatibility reasons, the tar command is one of the few commands that may accept some of its command-line options without a hyphen when placed in the first position of the command options. For example, `tar cf filename.tar files*` is equivalent (and more frequently used than) `tar -cf filename.tar files*`.

Restoring Files with the tar Command

The command `tar -xvf` will restore the entire contents of the source file or directory structure. To restore a portion of a **tar** file, use the path and name of the file you wish to extract. You must use the exact path and name that was used when you created the tar file. You can also make restores interactive by using the command `tar -wxvf [destination] [source]`.

The gzip Command

GNU zip (gzip) is a compression utility that reduces the size of selected files. Files compressed with gzip frequently have the .gz file extension. The gzip command has several options. These command options are described in the following table.

Option	Description
-d	Decompresses the file.
-f	Forces compression or decompression of a file even if it has multiple links or if the file exists.
-h	Displays a help screen.
-L	Displays the gzip license.
-n	Omits saving the original file name and time stamp.
-N	Saves the original file name and time stamp.

<i>Option</i>	<i>Description</i>
-q	Suppresses all warnings.
-r	Descends into the directory and compresses files.
-v	Displays the name and percentage reduction of the compressed or decompressed file.
-t	Checks the compressed file for integrity.

Syntax

The syntax of the gzip command is `gzip [options] {file name}`.

File Compression Utilities

File compression utilities, such as gzip, attempt to compress only regular files and ignore symbolic links. Compressed files can be restored to their original form using `gzip -d`, `gunzip`, or `zcat`. If the original file name saved in the compressed file is not suitable for its filesystem, a new name is provided from the original one.

The xz Command

xz is a data compression utility, similar to gzip, that reduces the size of selected files and manages files in the .xz file format. The xz command has several options.

These command options are described in the following table.

<i>Option</i>	<i>Description</i>
-d	Decompresses the file.
-f	Forces compression or decompression of a file even if it has multiple links or if the file exists.
-h	Displays a help screen.
-q	Suppresses all warnings.
-v	Displays the name and percentage reduction of the compressed or decompressed file.
-t	Checks the compressed file for integrity.

Syntax

The syntax of the xz command is `xz [options] {file name}`.

File Compression Utilities


File compression utilities, such as xz, attempt to compress only regular files and ignore symbolic links. Compressed files can be restored to their original form using `xz-d`, `unxz`, or `xzcat`. If the original file name saved in the compressed file is not suitable for its filesystem, a new name is provided from the original one.

The bzip2 Utilities

There are many file archiving utilities in Linux that enable you to compress, decompress, and run other text-processing utilities on files. The **bzip2** utility and its related commands manage file compression using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding, providing considerably better compression than other common file compression tools. Files compressed with bzip2 frequently have the .bz2 file extension. The bzip2-related commands are described in the following table.

<i>Utility</i>	<i>Description</i>
----------------	--------------------

<i>Utility</i>	<i>Description</i>
bzip2	Compresses files at a faster rate than the gzip command. The syntax of this command is <code>bzip2 {file name}</code> .
bunzip2	Decompresses files that are compressed using the bzip2 command. The syntax of this command is <code>bunzip2 {file name}</code> .
bzcat	Decompresses files that are compressed using the bzip2 command to the standard output. The syntax of this command is <code>bzcat {file name}</code> .
bzdiff	Runs the diff command on compressed files. The syntax of this command is <code>bzdiff {file name}</code> .
bzip2recover	Recovers data from damaged bzip2 files. The syntax of this command is <code>bzip2recover {file name}</code> .
bzless	Runs the less command on compressed files. The syntax of this command is <code>bzless {file name}</code> .
bzmore	Runs the more command on compressed files. The syntax of this command is <code>bzmore {file name}</code> .


	Note: bzip is an older version of the compression utility that has been replaced by bzip2.
---	---

The unzip Command

The unzip command is used to list, test, and extract compressed files in a ZIP archive. The unzip command comprises various options. A few of these options are described in the following table.

Most Linux distributions also include a zip command which may be used to compress files in the ZIP archive format.

<i>Option</i>	<i>Description</i>
-c	Extracts files to the standard output.
-t	Tests files before extraction.
-f	Extracts new files and freshens existing files.
-Z	Displays the archive comment.
-v	Extracts and lists files in a verbose manner.

	Note: The unzip command extracts all files from the specified ZIP archive into the current working directory.
---	--

Syntax

The syntax of the unzip command is `unzip [options] {file name} -d [directory]`.

Guidelines to Determine a Backup Strategy

Follow these guidelines to determine a backup strategy.

Determining a Backup Strategy

An effective backup strategy provides a quick and effortless recovery, minimizing the loss of data in the event of an unexpected crisis.

To choose an effective backup strategy, follow these guidelines:

- Determine the scope of the backup operation to be performed:
 - Do you need to back up data on a single computer?
 - Do you need to back up data on multiple computers?
 - Are the computers situated in a single location?
 - Are the computers spread across different locations?
- Make sure that you have all the necessary information about the data to be backed up:
 - Identify the amount of data that needs to be backed up.
 - Does the data reside on a single system?
 - Is the data distributed among several servers?
 - Can the data be easily replaced?
- Determine the most suitable time for performing backup operations so that users can continue working.
- Ensure that users are informed well in advance about scheduled backups.
- Review the storage space:
 - Do you have an adequate number of storage tapes?
 - Determine the reliability of the backup media.
 - Determine whether the previous backup versions can be erased or not. It is advisable to retain previous backup versions.
- Depending on the scope of the backup operation to be performed, determine if you have the required number of human resources to perform the backup operation.
- Test the backup operations performed and verify the integrity of the backed up files.

motd

The **message of the day (motd)** file is displayed to all users on a daily basis and can be used to inform users about scheduled backups. It requires less disk space than email messages. The contents of the **/etc/motd** file are displayed after a user successfully logs in.

The Hanoi Sequence

While performing incremental or partial backups, the Hanoi sequence helps minimize the number of tapes used. Backup procedures have several levels ranging from 0-9. Level 0 indicates a complete backup and ensures that the entire filesystem is copied. A level number greater than 0 indicates that all new files and files modified since the last backup of the same or lower level will be copied. This is known as an incremental backup.

It is practical to always start with a level 0 backup. A level 0 backup should be performed at regular intervals, preferably once a month or once every two months. Data should be stored in a set of fresh tapes each time a level 0 backup is performed. These tapes should be stored forever. After performing a level 0 backup, dumps of active filesystems need to be made on a daily basis. A modified "Tower of Hanoi" algorithm is used for this purpose. The sequence of dump levels followed in this method is 3 2 5 4 7 6 9 8 9 9.

Every week, a level 1 dump needs to be taken and the daily Hanoi sequence repeats beginning with a dump level of 3.

The `/etc/issue` and `/etc/issue.net` Files

The `/etc/issue.net` file is the login banner that users see when they make a network connection with the system. For example, when you use a command line tool to connect with a system, the content in the `/etc/issue.net` file is displayed. It includes all the welcome information text displayed whenever a new session is opened. The `/etc/issue` and the `/etc/issue.net` files constitute the login banner that is displayed to local users. The `/etc/issue` file can be customized.

The restore Command

The `restore` command enables you to restore files or filesystems from backups made using the dump command. This command can be used across networks to restore data.

The following table describes common restore command options.

<i>Option</i>	<i>Enables You To</i>
-C	Compare the backup file with the source file.
-i	Run the restore command in restore mode to restore backups partially.
-r	Perform a complete recovery of the backed up files.
-f <i>{/location of the backup file}</i>	Specify the location of the backup file.

Volume Number

While making backups of large files on removable storage devices, such as tape drives, the total size of the files will be split into smaller volumes and stored in multiple tape drives with each tape drive identified with a specific volume number. When you want to restore the backup made on multiple volumes, specify the volume number starting from the last volume number to the first volume number. The hard disk, because it is a single volume, will always have the volume number 1.

How to Back Up and Restore Files

Follow these general procedures to back up and restore files.

Back Up Files Using the tar and gzip Commands

To back up files using the tar and gzip commands:

1. Determine the files you want to back up and put them in a directory.
2. To group all the files using the tar command, enter `tar cvf {target archive file}.tar {source file(s)}`.
3. To compress the tar file using the gzip command, enter `gzip {archive file}.tar`.
4. Save the archive file in another location (FTP site, CD, DVD, tape, USB drive, and so on).

Restore Files from a Backup Using the gzip and tar Commands

To restore files from a backup using the tar and gzip commands:

1. Copy the files from your backup location (FTP site, CD, DVD, tape, floppy disk, and so on).

2. To uncompress the files using the gzip or gunzip command, enter `gzip -d {archive file}.tar.gz`.
3. To untar the files using the tar command, enter `tar xvf {archive file}.tar`.
4. If needed, move the individual saved files back to their respective locations.

TOPIC G Manage Databases Using MariaDB

Previously, you worked with files in a Linux filesystem. In addition to storing data in text files, you may need to store data in a format that will allow you to easily retrieve it when required; a database will serve this purpose. In this topic, you will manage databases with MariaDB.

A text file can store volumes of data, but retrieving it will be a problem because you may have to manually locate the information you are looking for. Databases, such as MariaDB and MySQL, allow you to store data in an organized manner, which enables efficient retrieval of specific data.

This will save you time and effort.

Databases

A **database** is an organized collection of information. It is used to facilitate easy storage and retrieval of data. In a database, data may be grouped into a series of records, which can be further organized into smaller segments of data called **fields**. A table is the basic storage unit of a database and it consists of rows and columns. The model of a database decides the way data is organized in it.

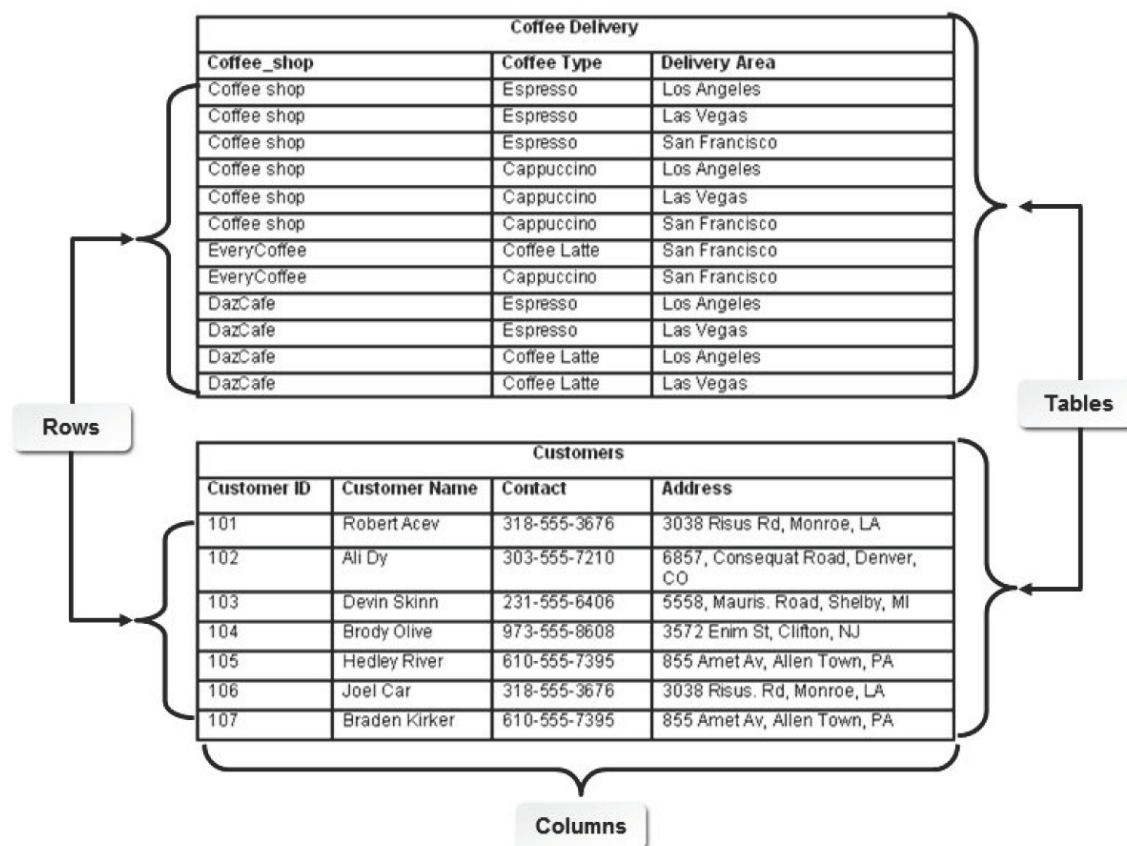


Figure 4-21: A database consisting of two tables.

Relational Databases

A **relational database** stores logically related data consistently in the form of related tables. These tables are linked through common fields or columns. The data stored is independent of files and is managed by a central

database engine that processes queries and manipulates data. Every row of data contains an identification key that identifies data uniquely and helps in reducing redundancy.

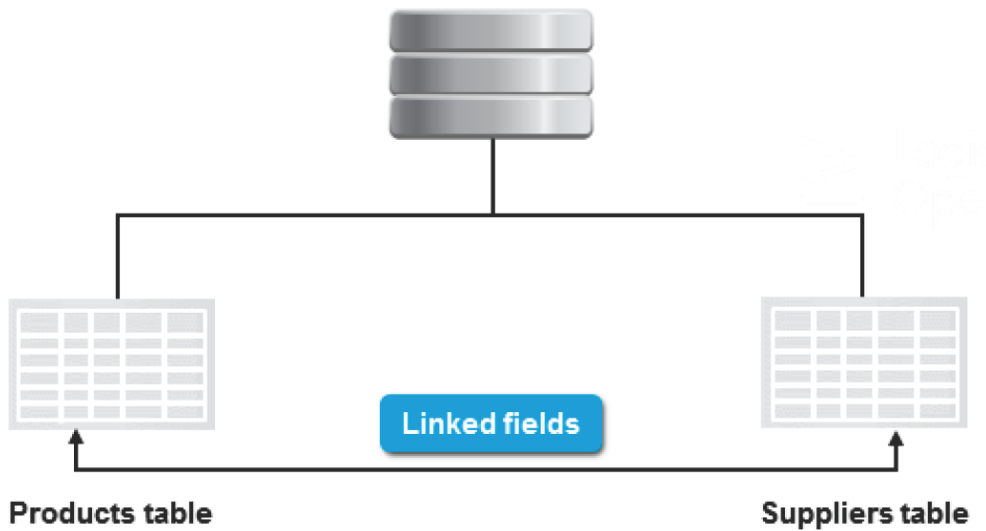


Figure 4-22: An example of a relational database.

Flat File Database

A flat file database stores data as flat files, which are static plain text documents. Flat files contain data that is structurally unrelated. The data in flat file databases cannot be retrieved or modified easily.

MariaDB

MariaDB is an open source **Relational Database Management System (RDBMS)** used for managing data. It is an enhanced, drop-in replacement for **MySQL**. It enables you to store, retrieve, manage, organize, and share data optimally. Using MariaDB, you can store data in one form and view it in various forms by analyzing and extracting only the relevant data from the database.

MariaDB also enables a group of networked computers to work together to enhance flexibility, efficiency, performance, scalability, and availability of the database, by eliminating time-consuming, error-prone tasks.

```
[root@localhost ~]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.41-MariaDB MariaDB Server

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> _
```

Figure 4-23: The welcome screen of the MariaDB client interface.



Note: MariaDB was created as an alternative to MySQL because of the commercial and uncertain direction for MySQL. MariaDB is intended to be fully compatible with MySQL, and a truly open source database system. Nearly all of the details and commands in

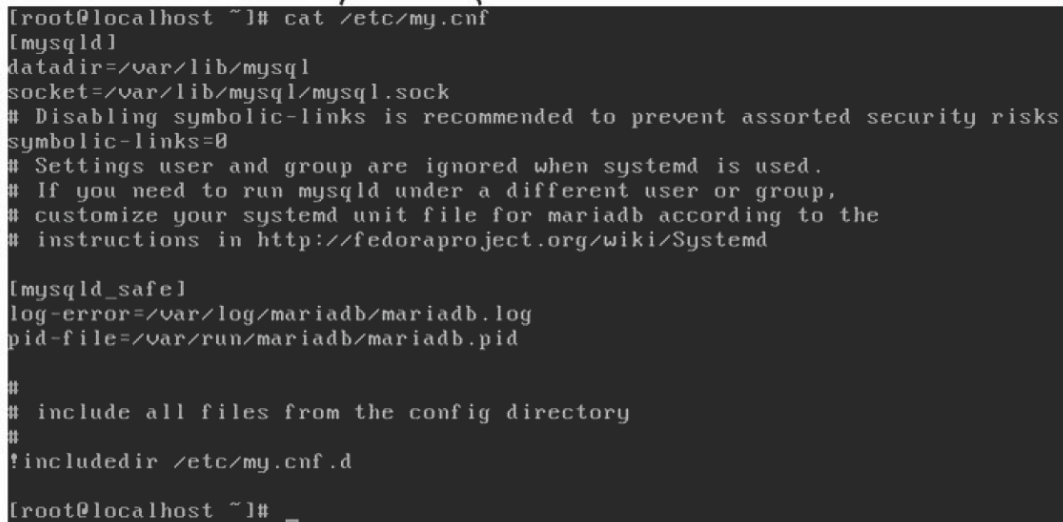
this topic will work for MySQL as well, and you will note that some of the MariaDB commands still have *mysql* in their names for full compatibility. For example, you access the MariaDB database monitor application via the command **mysql**.

The MariaDB Configuration File

The main configuration file of MariaDB, **my.cnf**, is located in the **/etc/** directory. You can set the global options for the MariaDB application in this file. The default options are usually sufficient; however, if you need to integrate MariaDB with other applications, you may need to modify this file.

This file allows you to set simple options, such as path to data directory; user name; and log file directory, as well as advanced options, such as `table_cache` and `key_buffer`, which can be set for the MariaDB daemon.

Main MariaDB configuration file



```
[root@localhost ~]# cat /etc/my.cnf
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d

[root@localhost ~]# _
```

Figure 4-24: The contents of the MariaDB configuration file.

The MariaDB Service

The [MariaDB service](#) manages the MariaDB database server. It allows you to manage the MariaDB server service. After installing MariaDB, you need to manually start this service. You can start, stop, restart, or view the status of the MariaDB server. This service needs to be running for clients to connect to and use MariaDB.

MariaDB daemon

```
[root@localhost ~]# systemctl start mariadb.service
[root@localhost ~]# systemctl status mariadb.service
mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled)
   Active: active (running) since Sun 2015-02-22 08:27:58 EST; 4s ago
     Process: 14243 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
     Process: 14215 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
    Main PID: 14242 (mysqld_safe)
      CGroup: /system.slice/mariadb.service
              └─14242 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
                └─14400 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql...
```

Feb 22 08:27:56 localhost.localdomain systemd[1]: Starting MariaDB database s...

Feb 22 08:27:56 localhost.localdomain mysqld_safe[14242]: 150222 08:27:56 mys...

Feb 22 08:27:56 localhost.localdomain mysqld_safe[14242]: 150222 08:27:56 mys...

Feb 22 08:27:58 localhost.localdomain systemd[1]: Started MariaDB database se...

Hint: Some lines were ellipsized, use -l to show in full.

```
[root@localhost ~]# _
```

Figure 4-25: Starting the MariaDB service.

MariaDB Commands

Clients use MariaDB commands to interact with the MariaDB server. These are the same commands that are used with the MySQL server.

<i>MariaDB/MySQL Command</i>	<i>Allows You To</i>
SELECT	Retrieve all records or records that match specific criteria.
CREATE	Create objects, such as a database, a table, or indexes, inside an RDBMS.
ALTER	Modify the existing table or database.
UPDATE	Update the values in a table.
USE	Make a particular database as the current database so that you can modify the objects of that database.
SHOW	Display the tables that are available in the current database.
INSERT	Enter values into a row or record of the table.
DESCRIBE	Display the structure of the table.
DELETE	Delete a row or rows from the table.



Note: The INSERT, UPDATE, and DELETE commands are also referred to as data manipulation commands because they allow you to modify the data in the table.



Note: These commands when given with their full syntax are also referred to as SQL Statements.

Optional Clauses in the select Statement

Optional clauses can be used with the SELECT statement. These clauses must be used in the order of precedence, and each of these clauses has a specific purpose.

Optional Clause	Purpose
WHERE	A condition used to specify that only certain rows can be retrieved from a table.
GROUP BY	A column identifier used to organize data into groups.
HAVING	A condition that works in conjunction with GROUP BY, specifying the groups to be included in the results.
ORDER BY	A condition that sorts query results by one or more columns.

Common SQL Queries and Actions

SQL queries can be used to perform various actions in a database. The following table illustrates various actions and their corresponding SQL queries.

Action	SQL Query
Extract all fields from a table.	SELECT * FROM <i>table_name</i> ;
Extract some fields from a table.	SELECT <i>field_name_1, field_name_2,..., field_name_n</i> FROM <i>table_name</i> ;
Extract all rows from a field.	SELECT <i>field_name</i> FROM <i>table_name</i> ;
Extract a row based on some condition.	SELECT <i>field_name</i> FROM <i>table_name</i> WHERE <i>condition</i> ;

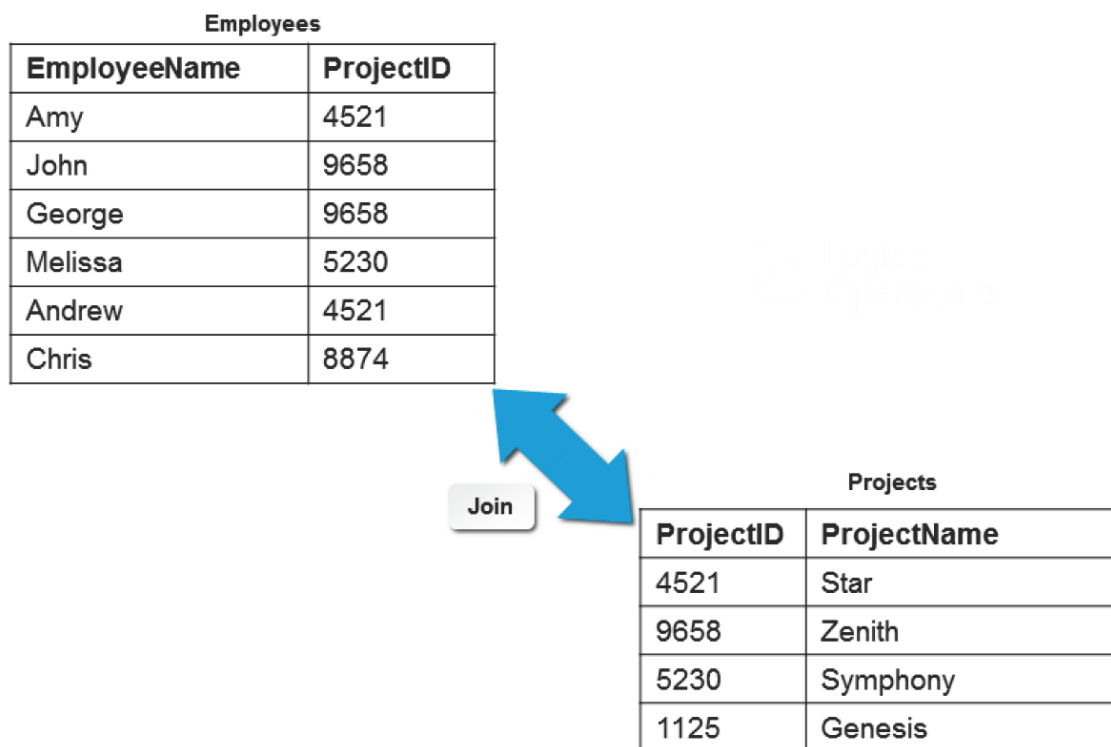
Range Operators Used with the select Command

The following table provides you with a list of range operators used with the WHERE clause of the SELECT command.

Condition	Use	Example
IN	Determines if a certain value is equal to any value in a specific list of values.	SELECT * FROM emp WHERE dept_id IN (10,20,30);
ANY, SOME	Compares a certain value to a specific list of values. Must be combined with one of the simple comparison operators.	SELECT * FROM emp WHERE dept_id= ANY (SELECT dept_id FROM dept WHERE dept_name='Sales' OR dept_name='Systems');
ALL	Compares a value to every value in a list. Must be combined with one of the simple comparison operators.	SELECT * FROM emp WHERE salary >= ALL (select avg(basic) FROM emp GROUP BY dept_id);
BETWEEN a AND b	Determines if a value is greater than or equal to a but less than or equal to b.	SELECT * FROM emp WHERE dept_id BETWEEN 10 AND 50;

Joins

A [join](#) is a query that is used to combine values in two or more tables in a relational database. It results in a temporary table called the joined table. A join connects tables by using their key fields.



```
select * from Projects left outer join Employees on Employees.ProjectID=Projects.ProjectID;
```

Figure 4-26: A join connects tables by using their key fields.

Let's say you have two tables: **Employees** and **Projects**. The **Employees** table has the **EmployeeName** and **ProjectID** fields, and the **Projects** table has the **ProjectID** and **ProjectName** fields. All the **ProjectIDs** appearing in the **Employees** table are not listed in the **Projects** table, and all the **ProjectIDs** appearing in the **Projects** table are not listed in the **Employees** table. To analyze which of the employees are assigned to a project and which of the projects have no employees assigned, you need to combine the information contained in these tables by using a join query.

Query output

```
mysql> select * from Projects left outer join Employees
-> on Employees.ProjectID=Projects.ProjectID;
+-----+-----+-----+-----+
| ProjectID | ProjectName | EmployeeName | ProjectID |
+-----+-----+-----+-----+
| 4521      | Star        | Amy          | 4521      |
| 4521      | Star        | Andrew       | 4521      |
| 9658      | Zenith      | John         | 9658      |
| 9658      | Zenith      | George       | 9658      |
| 5230      | Symphony    | Melissa      | 5230      |
| 1125      | Genesis     | NULL         | NULL      |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Figure 4-27: The result of a join query.

Types of Joins

Joins are broadly classified into two categories, inner joins and outer joins.

Join Type	Description
INNER JOIN	Allows only matching records to exist in the result set of the two joined tables.

<i>Join Type</i>	<i>Description</i>
OUTER JOIN	Selects all the values from one table and only those values from the second table that have matching values in the joined field.

How to Manage Databases Using MariaDB

Follow these general procedures to manage databases using MariaDB.

Install MariaDB

To install MariaDB:

1. Log in as **root** in the CLI.
2. To install the **mysqld** service of the MariaDB, on the terminal, enter `yum install mariadb-server`.
3. If necessary, enter Y to accept the installation.
4. Verify that the installation was successful by viewing the message displayed.

Start the MariaDB Daemon Service and Set it to Run Automatically

To start the MariaDB daemon service and set it to run automatically:

1. To start the **mysqld** service, on the terminal, enter `service mariadb start` or `systemctl restart mariadb.service`.
2. To automatically start the **mysqld** service at the system startup, enter `chkconfig mariadb on` or `systemctl enable mariadb.service`.
3. If necessary, to view the status of the **mysqld** service, enter `service mysqld status` To stop the MySQL server or restart the server, you can use the `service mysqld stop` and `service mysqld restart` commands, respectively.

Locate the MariaDB Configuration File

To locate the MariaDB configuration file:

1. Change to the `/` directory.
2. To locate the path of the MariaDB configuration file, enter `locate my.cnf`.
3. If necessary, open the **my.cnf** file in any text editor and modify the settings in the **my.cnf** file.
To apply the changes, restart the server.

Use MariaDB Commands

To use MariaDB commands:

1. To connect to the MariaDB server, type `mysql`.
2. To use the MariaDB commands, at the **mysql** prompt, enter the required commands in the format `{commands};`.
3. To exit from the **mysql** prompt, type `quit`.

Execute an Existing SQL Script

To execute an existing SQL script:

1. Log in as **root**.
2. Copy the **{filename}.sql** file to the **/var/lib/mysql** directory.
3. To connect to the MySQL server, type **mysql**.
4. To make it the current database, at the **mysql** prompt, type **USE {database name}**.
5. To execute the SQL commands, type **SOURCE {file name}.sql**;
6. If necessary, to verify that the script was successfully executed, type suitable sql commands.

Add Rows to a Table

To add rows to a table:

1. To display a description of the table, enter **DESC {table name}**;
2. Identify the values for fields that need to be added in the table.
3. To insert a row of data, enter **INSERT INTO {table name }({first field name},{second field name},...,[last field name]) VALUES ({first field value},{second field value}...,[last field value])**;

Modify Data

To modify data:

1. To display a description of a table, enter **DESC {table_name}**;
2. To modify data in the specified fields, enter **UPDATE {table_name} SET {first_field} = [value1], {second_field} = [value2] WHERE {condition}**;
3. Enter the appropriate command to delete data from the table.
 - To delete a row based on the specified condition, enter **DELETE FROM {table_name} WHERE {condition}**;
 - To delete all the rows of a table, enter **DELETE FROM {table_name}**;

ACTIVITY 4-1 Managing Files in Linux Review Scenario

Answer the following review questions.

1. Which tools will you use to search for files on your Linux system? Why?
2. Which text editor in Linux do you prefer? Why?

Summary

In this lesson, you located files, linked related files, created and edited files using a text editor, backed up and restored files, and explored MariaDB. This will help you customize the Linux system to your needs.

5 Managing Linux Permissions and Ownership

Lesson Time: 1 hour, 30 minutes

Lesson Introduction

While working with Linux® files, you may need to modify the permissions and ownership of these files. In this lesson, you will work with Linux permissions and ownership.

In Linux, changing permissions and ownership of files will enable you to restrict or assign those files to certain users. This will increase the overall security of your system. Novice users with high privileges can cause serious damage to a Linux system.

Lesson Objectives

In this lesson, you will work with Linux permissions and ownership. You will:

- Modify permissions on files and directories.
- Modify default permissions applied to files and directories.
- Modify ownership of files and directories.
- Set special permissions.

TOPIC A Modify File and Directory Permissions

Now that you can work with Linux files, you can begin to alter their permissions to restrict who can access or edit various files. In this topic, you will modify file and directory permissions.

Systems in your workplace may hold files and other data that should not be made accessible to all users. To prevent accidental modification or deletion of important information, certain files should be accessible only to the root user or to the owner of the file. As a system administrator, it is your responsibility to modify file and directory permissions that will enable you to restrict or allow user access to system critical and generic files.

Permissions

Permissions are access rights assigned to users, which enable them to access or modify files and directories. Permissions can be set at different levels and for different access categories. The `ls -l` command can be used to view the permissions of a file.



Figure 5-1: Permissions set for a few users.

The `ls -l` command gives you a long list of the files and directories in your current working directory. Each item in the list contains seven columns. The contents of the columns are described in the following table.

Column Number	Description
1	Permission string. This identifies if the item is a file or directory, the user, group, and other permission assignment, and the access method.
2	Number of links. Files generally have a link count of 1. For directories, the link count is the number of directories under it plus 2; 1 for the directory itself and 1 for the parent. Links are similar to Windows shortcuts; they point to the location where the file exists and allow you to access and view the file.
3	Displays the owner of the file or directory.
4	Displays the group to which the owner of the file belongs. All members of this group have the group permission listed in the permission string. The administrator adds users to a group so that permissions can be assigned to the group instead of to each user.
5	Lists the size (in bytes) of the file or directory.
6	Displays the date and time the file was created or last modified.
7	Displays the file or directory name.



Note: Use the `ls -ld [directory name]` command to list directory entries of the specified directory. The contents of the directory will not be displayed.

Access Categories

Access categories in Linux permissions decide how Linux interprets the permissions of a file. If a user's UID matches the permissions of the file, the user level permissions are applied. If the GID of the user matches the permissions, group permissions are granted. If neither of the permissions match, the general permissions for others are applied. The symbols for the access categories are listed in the following table.

Access Category	Description
u	Modifies permissions at user level.
g	Modifies permissions at group level.
o	Modifies permissions for other users.

Access Category	Description
a	Modifies permissions for all users globally.

Permission String

The output of the ls -l command shows the permission string for a file or directory. The permission string contains 11 characters.

- The first character indicates the type of file; **d** for directory and hyphen (-) for file.
- Characters at the second, third, and fourth positions denote permissions of the owner or user of the file or directory.
- Characters at the fifth, sixth, and seventh positions denote group permissions.
- Characters at the eighth, ninth, and tenth positions denote permissions for others.
- The final character indicates the access method for the file; period (.) for SELinux security context and plus (+) for any other combination of alternate access methods.

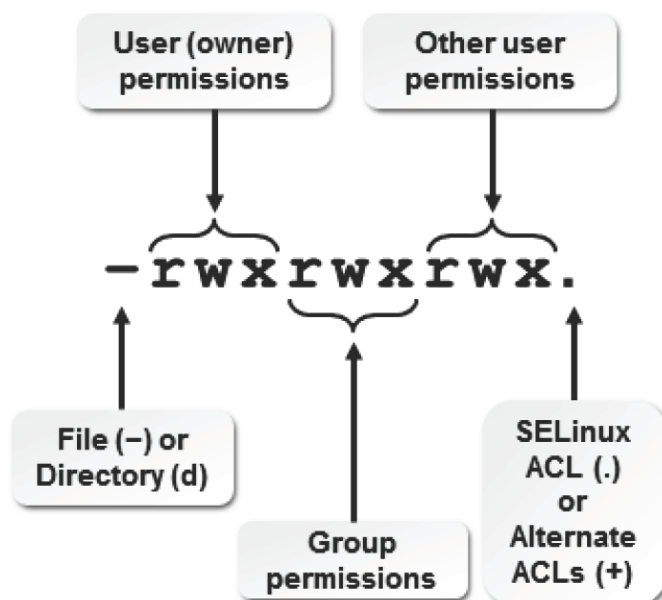


Figure 5-2: Details of permission string output from ls.

Permission Levels

Permissions are granted or denied by the owner of the file. The following table lists the levels of various permissions and their description.

Level of Permission	Description
User level r/w/x permission	Only the owner can read, write, and execute the file.
Group level r/w/x permission	Only the members of groups to which the file belongs to can read, write, and execute the file.
Other level r/w/x permission	All users can read, write, and execute the file.

File Owner

A **file owner** is the user who creates a file or directory. The file owner can set permissions to specify whether other users or groups have rights to read, write, or execute the file.

The chmod Command

The chmod command enables you to modify default permissions of a file or directory. Only the owner of the file or the system administrator can change the permissions of the file or directory.

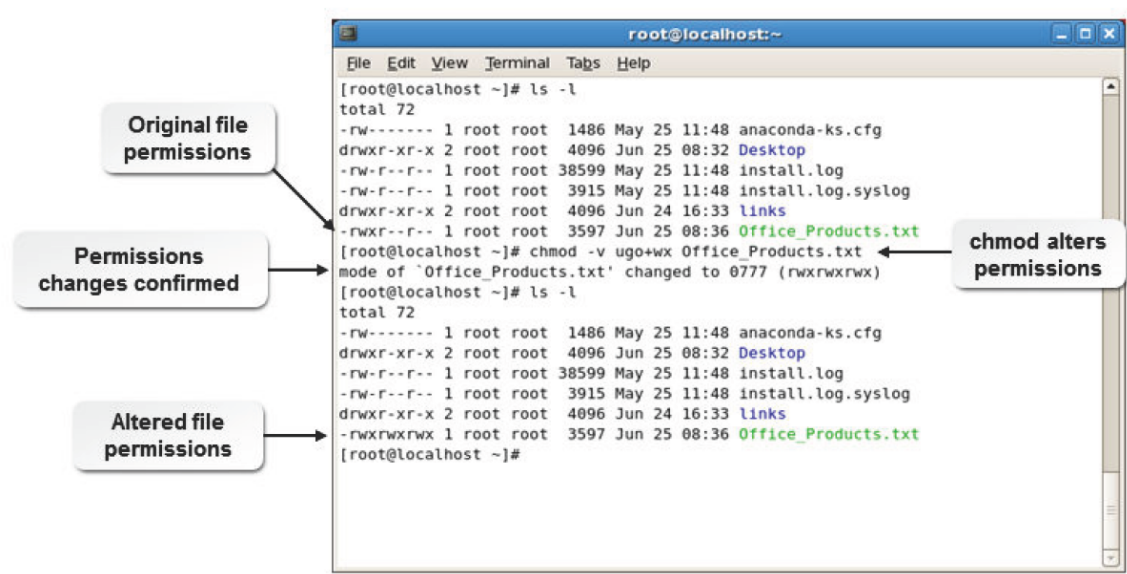


Figure 5-3: Modifying permissions using the chmod command.

Syntax

The syntax of the chmod command is `chmod [options] {mode} {file name}`.

chmod Command Options

The chmod command supports different options to modify permissions. One or more of these options may be used at a time.

Option	Description
-c	Reports changes that are made in permissions.
-f	Hides most error messages.
-v	Displays a diagnostic entry for every file processed.
-R	Modifies permissions of files and directories recursively.

chmod Modes

The chmod command supports two modes: the character mode and the numeric mode. The character mode allows you to set permissions using three components, namely, access categories such as u/g/o/a; operators such as +/=/-; and permission attributes such as r/w/x. The numeric mode is represented by three-digit numbers.

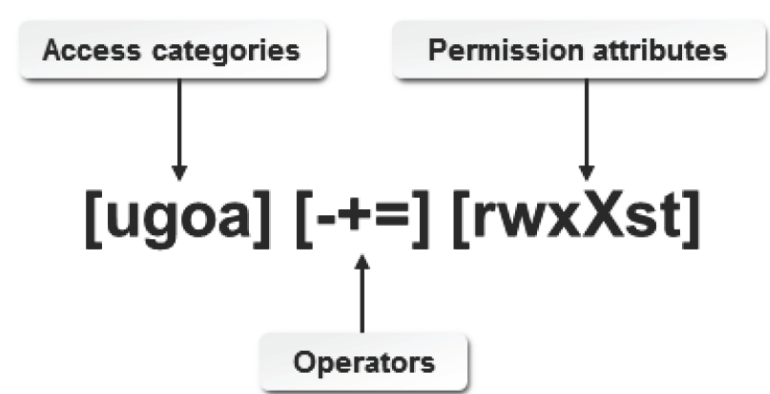


Figure 5-4: Components of the character mode.

Operators Associated with Permissions

Operators decide whether a permission is to be granted or removed. Common operators associated with Linux permissions are listed in the following table.

<i>Operator</i>	<i>Description</i>
+	Grants permissions.
-	Denies permissions.
=	Causes the permissions assigned to overwrite other existing permissions. Assigns permissions similar to those of the reference file.

Permission Attributes

Permission attributes define exactly what a user is allowed to do with a particular file. The three permission attributes are listed in the table.

<i>Permission Attribute</i>	<i>Allows You To</i>
r (read)	View file content.
w (write)	Modify file content.
x (execute)	Run a file (if it is an executable program and is combined with the read attribute).

Changing Permissions Using the Character Method

The permissions of a file or directory can be changed using the character method. The syntax of the chmod command when using this method is `chmod [options] {access categories} {operators}{permission levels} {file name or directory name}`.

Changing Permissions Using Octal Permission Numbers

Linux systems use octal (base-8) numbers to specify permissions. Each permission (r, w, and x) has an associated number.

<i>Octal Number</i>	<i>Attribute</i>	<i>Letter</i>
4	read	r
2	write	w
1	execute	x

By adding the octal numbers for the permissions you want to grant, you get the overall permission number to assign to a directory or file. Full permissions (read, write, and execute) are equivalent to 4 + 2 + 1, or 7. Read and write permissions are equivalent to 4 + 2, or 6. Complete permissions are expressed as a three-digit number, where each digit corresponds to the user, the group, and other permissions, respectively.

The syntax of the number method to change permissions is `chmod {number} {file name}`.

Commonly used octal permission numbers are listed in the table.

<i>Octal Permission</i>	<i>Permission Attribute Equivalent</i>
755	u=rwx,g=rx,o=rx

<i>Octal Permission</i>	<i>Permission Attribute Equivalent</i>
700	u=rwx,g=,o=
644	u=rw,g=r,o=r
600	u=rw,g=,o=

How to Modify File and Directory Permissions

Follow these general procedures to modify file and directory permissions.

View File or Directory Permissions

To view file or directory permissions:

1. Log in as a user.
2. View the permissions of a file or directory.
 - View the permissions of a file or directory, as necessary, from the command line.
 - To view the permissions of a file, enter `ls -l {file name}`.
 - To view the permissions of a directory, enter `ls -ld {directory name}`.
 - View the permissions of a file or folder using the Nautilus browser.
 - a. Right-click the file or folder and select **Properties**.
 - b. To view the permissions of the file or folder, select the **Permissions** tab.

Modify File or Directory Permissions

To modify file or directory permissions:

1. Log in as a user or root, depending on what type of files you want to modify.
2. Change the permissions of a file or directory in the CLI or the GUI.
 - To modify permissions in the CLI, enter `chmod [options] {file or directory name}`.
 - Modify permissions in the GUI.
 - a. Right-click the file or directory and select **Properties**.
 - b. Select the **Permissions** tab.
 - c. From the **File Access** or **Folder Access** drop-down list, select the desired permission for owner, owner groups, and other groups.

TOPIC B Modify Default Permissions

Previously, you modified file and directory permissions. Now, you need to know how to alter default permissions of files. In this topic, you will modify default permissions in Linux.

Modifying default file permissions will allow you to set higher or lower security levels for files created by users. This will allow high-level users to create files that are completely secure from other users.

Default File and Directory Permissions

In Linux, default permissions are assigned to newly created files and directories based on user privileges. For files created by the root user, the default permission is 644, which means that the root user has read and write permissions, while group users and others will have only read permission. For directories created by the root user, the default permission is 755, which means that the root user has read, write, and execute permissions, while group users and others will have only read and execute permissions. In the case of users with limited access rights, Linux assigns a permission of 664 for newly created files and 775 for newly created directories. These default permissions are determined by the user file creation mask, or umask. However, the default permissions may be altered by the root user.

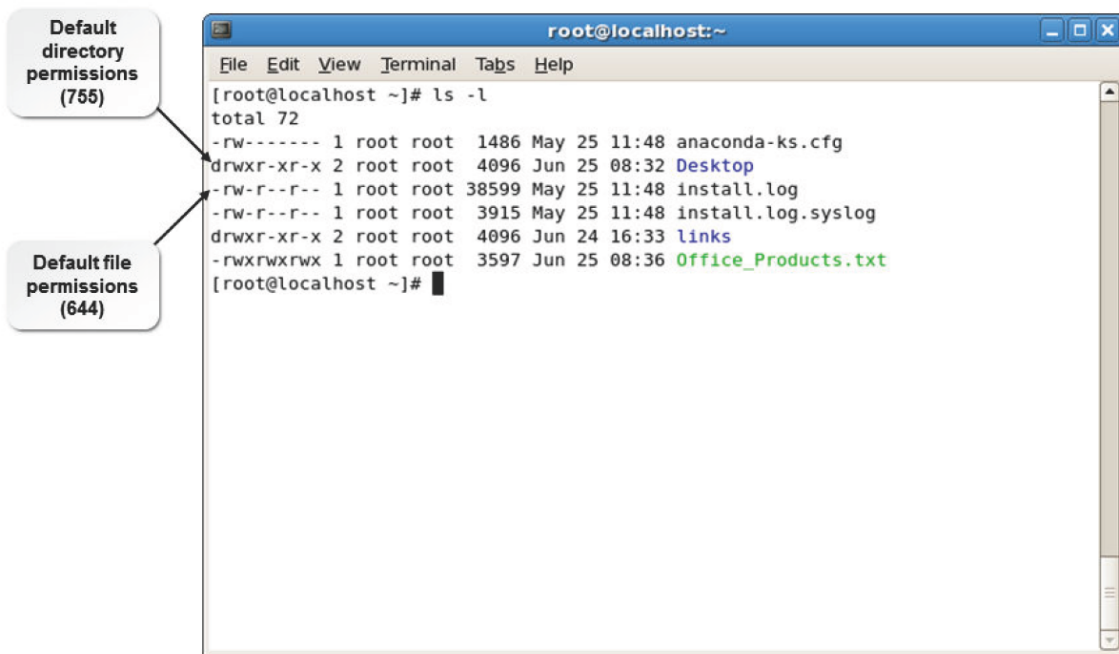


Figure 5-5: The `ls` command displaying the default file and directory permissions.

The umask Command

The `umask` command automatically alters the default permissions on newly created files and directories. The default permissions on newly created files and directories can be changed for security reasons.

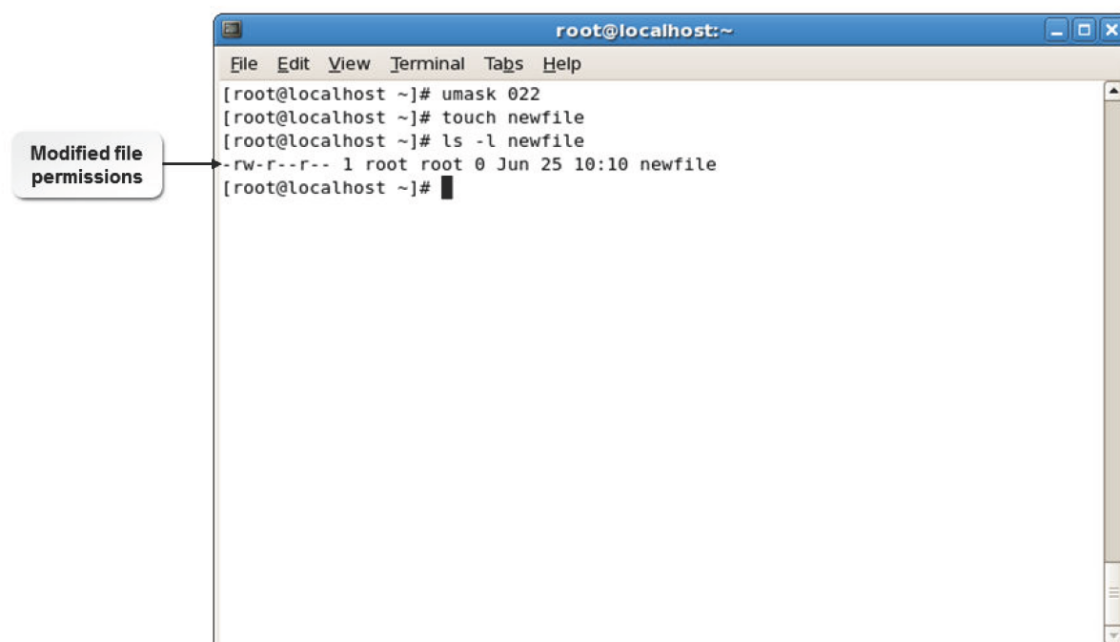


Figure 5-6: Default file permissions modified using the `umask` command.

Syntax

The syntax of the *umask* command is `umask {number}`.

Three-Digit and Four-Digit File Modes and Masks

When written in octal, numeric format, file permissions typically have three digits, each digit corresponding to the user, group, and others permissions. However, file permissions may also be written with four digits, with the new, leading digit signifying any advanced permissions to be defined (or 0, for none). For example, the base permissions for nonexecutable files in Linux are `rw-rw-rw-`, or `666`. This is equivalent to the octal format of `0666`. The permission mode `755` is equivalent to `0755`.

The Effect of *umask* on Files

By default, the base permissions for nonexecutable files in Linux are `rw-rw-rw-`, or `0666`. By entering `umask 0022`, the permissions assigned to all files, created from that moment until the system is restarted, will be `rw-r--r--` (`0644`). The numbers given with the `umask` command specify the permissions that need to be cleared from the default settings. The first digit allows you protect the file by setting advanced permissions. The other three digits allow you to set the normal permissions.

Number	Clears
0	No advanced permissions set by default.
0	Nothing from the users' default permissions, leaving <code>rw-</code> .
2	Write permission from the group, leaving just <code>r--</code> .
2	Write permission from others, leaving just <code>r--</code> .

These settings correspond to the default *umask* of the root user.

The Effect of *umask* on Directories

By default, the base permissions for directories in Linux are `rxwxrwxrwx` or `0777`. By entering `umask 0022`, the permissions assigned to all directories, created from that moment until the system is restarted, will be `rxwxr-xr-x` (`0755`). The numbers given with the `umask` command specify the permissions that need to be cleared from the default settings. The first digit allows you to protect the directory by setting advanced permissions. The other three digits allow you to set the normal permissions.

Number	Clears
0	No advanced permissions set by default.
0	Nothing from the users' default permissions, leaving <code>rxw</code> .
2	Write permission from the group, leaving just <code>r-x</code> .
2	Write permission from others, leaving just <code>r-x</code> .

These settings correspond to the default *umask* of the root user.

How to Modify Default Permissions

Follow these general procedures to modify default permissions.

Modify Default Permissions

To modify default permissions:

1. Log in as **root**.

2. To change the default umask value, enter `umask {umask number}`.
3. Create a new file to verify that files created from now on have different default permissions.

TOPIC C Modify File and Directory Ownership

With an understanding of default permissions, you can modify the owners of files and directories. A user or group may not want other users to access the files created by them. Also, users may require access to files created by other users. In this topic, you will modify file and directory ownership.

Imagine that you have been working on a project that is to be taken over by one of your colleagues.

Consequently, you will need to transfer the ownership of all the files you created for this project to your colleague. Modifying file and directory ownership will enable you to help your colleague make the transition to project owner.

The `chown` Command

The `chown` command can be used to change the owner, the group, or both for a file or directory.

The following table describes how to use this command.

<i>Command Syntax</i>	<i>Description</i>
<code>chown {user name} {file name}</code>	Changes the owner but not the group.
<code>chown {user name}:{group name} {file name}</code>	Changes the owner and the group.
<code>chown {user name} {file name}</code>	Changes the owner and the group. The group will be changed to the specified user's login group.
<code>chown :{group name} {file name}</code>	Changes the group but not the owner. This is the same as using the <code>chgrp</code> command.

Figure 5-7: File ownership changed using the chown command.

Recursively Changing Ownership

You can combine the chown command with the -R option to recursively change ownership through a directory structure. You can also use metacharacters to change ownership of groups of files at the same time.

Changing Group Ownership

The chgrp command is used to change the group ownership of a file or directory. The syntax of the command is `chgrp {group name} {filename}`.

How to Modify File and Directory Ownership

Follow these general procedures to modify file and directory ownership.

View File or Directory Ownership

To view file or directory ownership:

1. Log in as a user.
2. View the ownership of a file or directory in the desired mode.
 - View the ownership of a file or directory at the command prompt.
 - To view the ownership of a file, enter `ls -l {file name}`.
 - To view the ownership of a directory, enter `ls -ld {directory name}`.

Modify File or Directory Ownership

To modify file or directory ownership:

1. Log in as **root**.
2. Change the ownership of a file or directory.
 - Modify file or directory ownership at the command prompt.
 - To change the user ownership of the file or directory, enter `chown [command options] [username]: {group name} {file or directory name}`.
 - To change the group ownership of the file or directory, enter `chgrp [command options] {group name} {file or directory name}`.

TOPIC D Set Special Permissions and Attributes

Now that you have modified file and directory ownership, you can set special permissions for users.

There may be instances when you have to use special permissions to enable users to access files or directories. In this topic, you will set advanced permissions.

While it is desirable to allow only users with root or administrative permissions to execute certain commands, sometimes other users need to be able to issue them. Setting advanced permissions will allow other users to execute and maintain system utilities so that the Linux administrator does not have to do them for each user. This will save the administrator time and effort.

Special Permissions

Special permissions are used when normal permissions become inadequate, usually in the case of processes. With special permissions, less privileged users are allowed to execute a file that can usually be run only by the root user. **Set User ID (SUID)**, or `setuid`, is the permission that allows a user to have similar permissions as the owner of the file. **Set Group ID (SGID)**, or `setgid`, is the permission that allows a user to have similar permissions as the group owner of the file.

The SUID and SGID Permissions

The SUID and SGID commands are powerful tools that enable users to perform tasks without problems that could arise with users having the actual permissions of that user or group. However, these can be dangerous tools too.

While changing the permissions of a file to be either SUID or SGID, the following points should be considered:

- Use the lowest permissions needed to accomplish a task. It is recommended not to give a file the same SUID or SGID as the root user. A user with fewer privileges often can be configured to perform the task.
- Watch for back doors. If the user runs a program with the SUID set to root, then the user retains root as the effective UID when the user goes through the back door. The following can be used as back doors:
 - Programs that enable you to shell out.
 - Programs with multiple entrances and exits.

The `chattr` Command

The `chattr` command is used to change the attributes of a file on a Linux filesystem.

The following table lists the description for the options used in the syntax of the `chattr` command.

<i>Command Option</i>	<i>Used To</i>
-R	Recursively change the attributes of directories and their contents.
-V	Display the output of the <code>chattr</code> command and print the program version.
-v {version}	Set the version number of a file.
+i	Mark the file as read-only, or immutable.
-i	Remove the read-only, or immutable, attribute of the file.

Figure 5-8: File attribute changed by using the chattr command.

Syntax

The syntax of the chattr command is `chattr [-RV] [-v version] {[mode]}{file names}`.

The lsattr Command

The `lsattr` command is used to list the attributes of a file on a Linux filesystem.

The following table describes the options used in the syntax of the lsattr command.

Command Option	Used To
-R	Recursively list the attributes of directories and their contents.
-V	Display the program version.
-a	List all files in directories.
-d	List directories like files, instead of listing their contents.
-v	List the version number of the file.

Syntax

The syntax of the lsattr command is `lsattr [-RVadv] [file names]`.

Sticky Bits

A `sticky bit` is a permission bit that provides protection for files in a directory. It ensures that only the owner of a file can delete the file or directory. A sticky bit also forces a program or file to remain in memory so that it need not be reloaded when it is invoked again. A sticky bit on a file indicates to the operating system that the file will be executed frequently. Files with sticky bits are kept in the swap space or in the disk space that is set aside for virtual memory.

Figure 5-9: A file in the memory protected by a sticky bit.

The Immutable Flag

The **immutable flag** is an extended attribute of a file or directory that prevents it from being modified. The immutable flag is not set on all files. It is set only on those files, such as configuration files, that should not be modified. A single directory can have a mix of mutable and immutable files and subdirectories. Also, an immutable subdirectory can have mutable files.

Figure 5-10: The immutable flag prevents modification of the file.

The ACL

The **Access Control List (ACL)** is a list of permissions attached to an object. Usually, a file object in Linux is associated with three sets of permissions—read, write, and execute—for the three user groups: owner, group, and other. ACLs can be used for situations where the traditional file permission concept does not suffice. They allow the assignment of permissions to individual users or groups even if these do not correspond to the owner or the owning group.

Figure 5-11: The ACL of a file displayed using the getfacl command.

Commands Associated with the ACL

ACLs can be managed at filesystem level or at the file and directory level. To find out the ACL specifications of a file, you can use the `getfacl` command. To set the access control specifications for files and directories, you can use the `setfacl` command with its different options.

Special Permission Commands

Special permission commands can be used effectively to set special file or directory access rights for users. Some of the common commands to set these permissions are listed in the following table along with their syntax.

<i>Command Syntax</i>	<i>Used To</i>
<code>chmod u{operator}s {file name}</code>	Set the SUID for a file.
<code>chmod g{operator}s {directory name}</code>	Set the SGID for a directory.
<code>chmod o{operator}t {file name}</code>	Set the sticky bit for a file.
<code>umask {value}</code>	Set the default file creation mode.
<code>chattr {operator}i {file name or directory name}</code>	Set the immutable flag for a file or directory.

How to Set Special Permissions and Attributes

Follow these general procedures to set special permissions and attributes.

Set Special Permissions for Files and Directories

To set special permissions for files and directories:

1. Log in as a user.
2. Set special permissions for files and directories.
 - To add the SUID permission for a file, enter `chmod u+s {file name}`.
 - To add the SGID permission for a directory, enter `chmod g+s {directory name}`.
 - To add the sticky bit permission for the directory, enter `chmod +t {directory name}`.

3. If necessary, to verify the changes, enter `ls -l`.

Manage ACLs for Files and Directories

To manage ACLs for files and directories:

1. Log in as **root**.
2. Manage ACLs for files and directories:
 - To view the ACL for the specified file or directory, enter `getfacl {file name or directory name}`.
 - To set the ACL for the specified file or directory, enter `setfacl -m {g | u | o}:{user name or group name}:[r,w & x combination] {file name or directory name}`.
 - To inherit all the permissions for the newly created content in the specified directory name, enter `setfacl -m d:{g | u}:{user name or group name}:[r,w & x combination] {directory name}`.
 - To set the ACL for the specified file or directory, enter `setfacl -x {g | u}:{user name} {file name or directory name}`.

Change the Default Group Owner of New Files

To change the default group owner of new files:

1. Log in to the CLI as **root**.
2. To change the group ownership of the directory, enter `chgrp {group name} /{directory name}`.
3. If desired, to view the status of the group owner of the directory name, enter `ls -l`.
4. To set the SGID for the directory, enter `chmod g+s /{directory name}`.
5. If necessary, to view the status of the group permissions of the directory, enter `ls -l`.
6. Enter `cd /{directory name}`.
7. If necessary, enter `touch {file name}`.
8. If necessary, to verify the status of the group owner of the newly created file, enter `ls -l`.

ACTIVITY 5-1

Managing Linux Permissions and Ownership Review

Scenario

Answer the following review questions.

1. What methods do you think you might use to preserve confidentiality of information on Linux systems?
2. In what situation would you need to modify the default permissions?

Summary

In this lesson, you modified permissions and ownership of files and directories in a Linux system.

You also set process permissions to allow other users to execute a process generally run by an administrator, saving them both time and effort. Now, you will be able to efficiently control the security of your Linux system.

6 Printing Files

Lesson Time: 1 hour, 30 minutes

Lesson Introduction

In the last lesson, you worked with files in Linux®. Now, you may want to print those files containing essential information. In this lesson, you will work with printer hardware and software.

Like all standard operating systems, Linux allows its users to print files. You may have trouble viewing lengthy files continuously on a monitor and may consider printing them.

Linux includes effective printing utilities that allow you to print configuration files and any other text documents you create.

Lesson Objectives

In this lesson, you will print files. You will:

- Configure a local printer.
- Format text in a file and print it.
- Configure remote printing.

TOPIC A Configure a Local Printer

Previously, you worked with various files in Linux and now might want to print some of them.

Before you can print a file, you must configure a printer to work with the Linux operating system. In this topic, you will configure a local printer to work with your system.

Computers at homes and offices are regularly used for printing. Like other operating systems, Linux also supports printing. However, not all printers are compatible with the Linux operating system.

You must check for compatibility before selecting a printer to be used with your Linux system.

Even if a printer is compatible, you will need to know how to configure it with a system before it can be used for printing.

Printer Software

Printer software is a program that enables a printing device to print text or graphics on media. The printer software provided with a printer includes a driver and utilities. The printer driver allows users to choose settings for the printer. The printer utilities ensure that the printers are in operating condition.



Figure 6-1: Printer software comprises drivers and utilities.

PostScript®

PostScript is a **Page Description Language (PDL)** that tells a printer how to display text or graphics on a page. Laser printers primarily use PostScript for printing documents. The print quality is high because it resizes fonts and images without distortion. PostScript can work on different platforms and printers, and therefore can be used to share documents on the Internet.



Figure 6-2: PostScript helps a system communicate with a printer.

Linux-Compatible Printers

PostScript is the standard PDL supported by Linux. Therefore, most of the Linux-compatible printers are PostScript printers. PostScript printers support printing in Linux because PostScript Printer Definitions (PPDs) describe and provide access to printer-specific features. PPDs function as drivers for PostScript printers and provide a unified interface for the printer's capabilities.

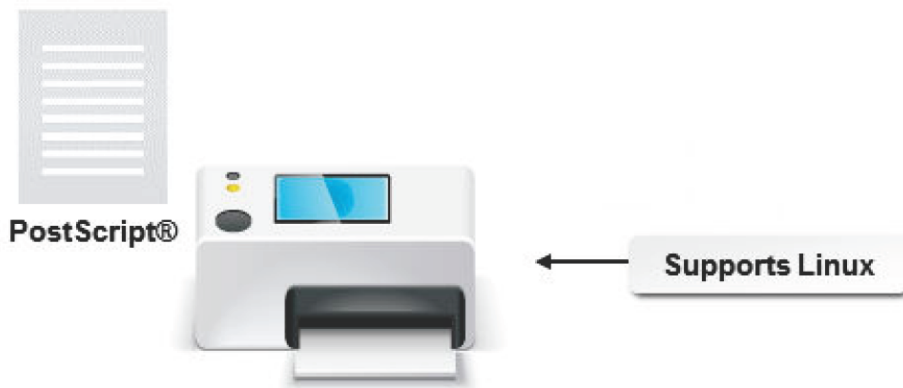


Figure 6-3: A PostScript printer.



Note: For information about specific distributions of Linux and installation of printer drivers, visit the website of the specific distribution.

Configuring a Local Printer

When Linux is installed, you can configure a printer to work with the operating system. Using the print system manager, you can add printers to the system. A local printer is attached directly to the Linux workstation via a parallel, serial, or USB port. A remote printer is attached to a UNIX or Linux machine elsewhere on the network.



Note: Line Printer Daemon (LPD) is a Linux system service for network printing.

CUPS

The **Common UNIX Printing System (CUPS)** is a systematic print management system for Linux; this printing system allows a computer to function as a print server. A system running CUPS is a host that can initiate print jobs from client systems. These jobs are then processed and sent to the appropriate printer. The main advantage of CUPS is that it can process different data formats on the same print server. CUPS is designed for scheduling print jobs, processing administrative commands, and providing printer status information to local and remote programs. The CUPS configuration is accessed by using a browser window. By making changes through the browser interface, you can edit the **cupsd.conf** file, which is located in the **/etc/cups** directory.

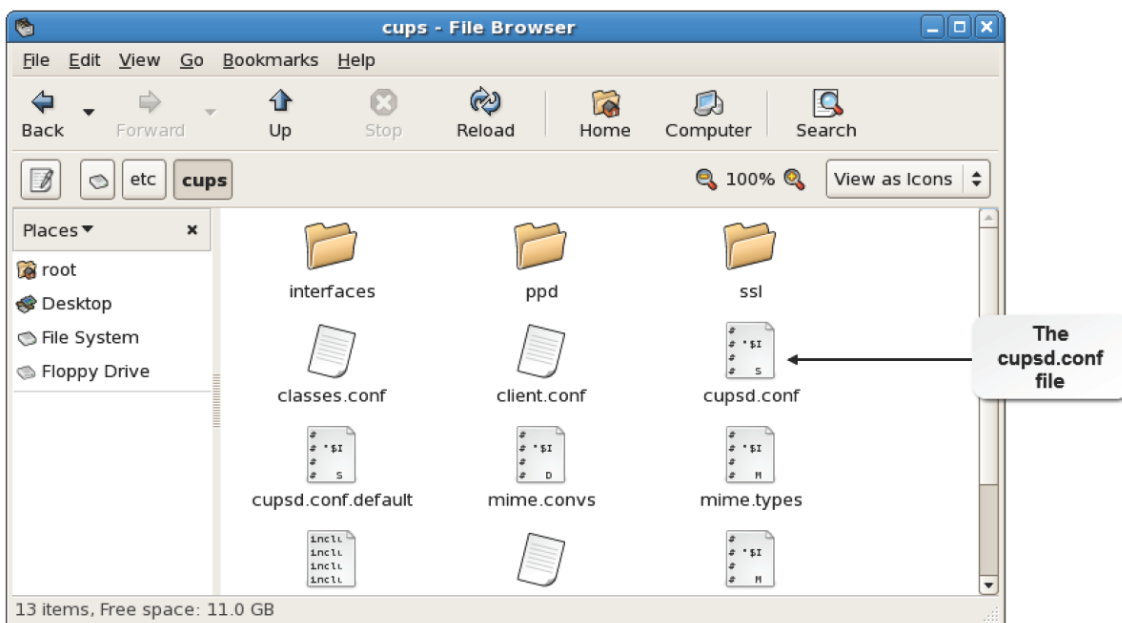


Figure 6-4: Accessing the cupsd.conf file located in the /etc/cups directory.

The Print Process

The print process enables you to print a document. Various steps are involved in this process. When a user issues a command to print a document in an application, the following steps take place:

1. The application invokes the printing client software.
2. The file passes from the printing client to the printer *spooler*.
3. The file then passes through several filters that convert the document from one format to another, before being finally sent to the printer.
4. The printer then prints the file.

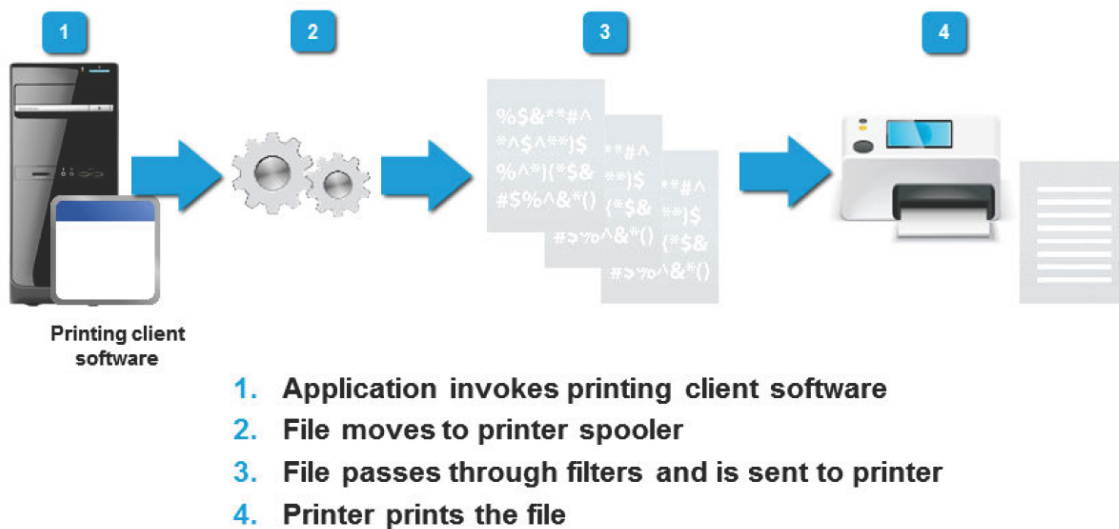


Figure 6-5: Process involved in printing a document.

Spooling

Spooling is the procedure by which print jobs are temporarily stored. If the printer is busy, print jobs are placed in a waiting line or print queue. These jobs are stored in a temporary storage space called a *spool*. Files in the queue are printed when the printer becomes free. This prevents programs from having to wait during a slow printing process.

Print Queues

A **print queue** is a temporary storage area that sorts incoming print jobs. Print queues are used by the print daemon so that applications that need to use the printer do not have to wait to issue the command for printing until the current print job is completed. A list of print jobs contains details of the file being printed currently and the files yet to be printed. Print queues allow multiple users to share a printer.

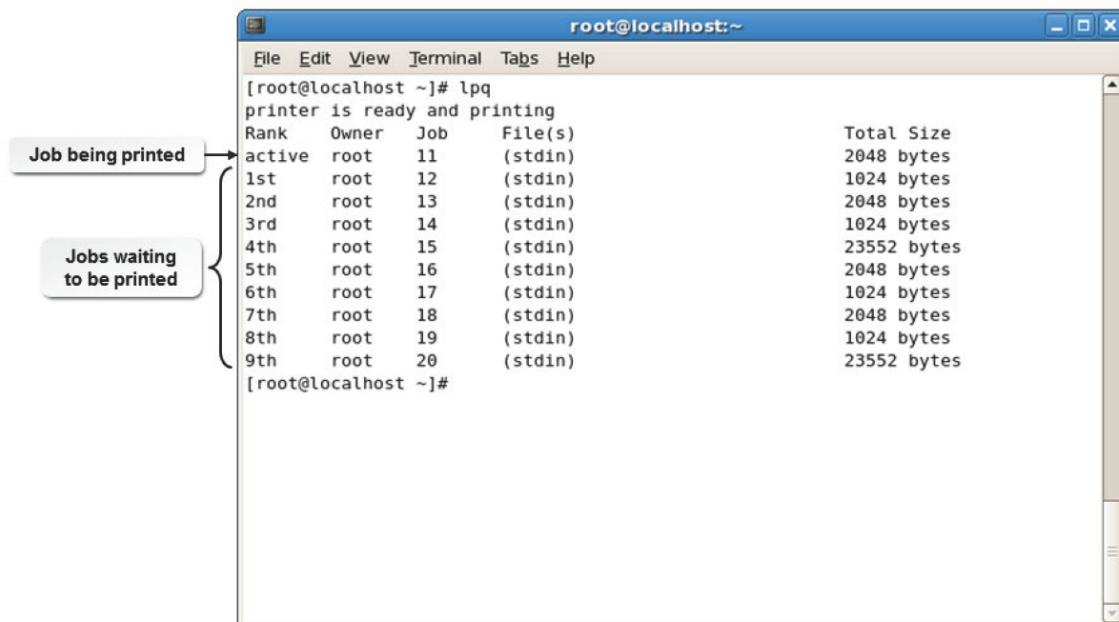


Figure 6-6: Print jobs in a queue.

How to Configure a Local Printer

Follow these general procedures to configure a local printer.

Add a Printer Using the Printer Configuration Dialog Box

To add a printer using the **Printer configuration** dialog box:

1. Connect the printer to the USB port. Switch on the printer and load paper.
2. Log in as **root** in the GUI.
3. To open the **Printer configuration** dialog box, select **Applications** → **System Tools** → **Settings** → **Printers**.
4. In the Printers Settings window, on the toolbar, select **Unlock** and then select **+**
5. In the **New Printer** dialog box, in the **Printer Name** text box, enter the name of the printer.
6. If necessary, in the **Description** text box, enter the description of the printer.
7. If necessary, in the **Location** text box, enter the location details of the printer.
8. To continue with the printer installation, select **Add**.
9. To install the printer, in the **Select Connection** list box, in the **Devices** list box, select the printer device and select **Forward**.
10. In the **Makes** list box, select the printer make and select **Forward**.
11. In the **Models** list box, select the corresponding model, and in the **Drivers** list box, select the corresponding driver and select **Forward**.
12. To add a new printer, on the confirmation page, select **Apply**.
13. To close the **Printer configuration** dialog box, from the menu, select **File** → **Quit**.

Add a Printer Using the CUPS Browser Interface

To add a printer using the CUPS browser interface:

1. Log in as **root** in the GUI.
2. In the terminal, enter `system-config-printer`.
3. In the **Printer Settings - localhost** dialog box, select **Add** and then select **Printer** from the options.
4. On the **New Printer** web page, in the **Add New Printer** section, provide the **Name**, **Location**, and **Description**, and then select **Continue**.
5. To configure a local printer, on the **Administration** tab, in the **Device for {printer name}** section, from the **Device** list, select **USB**. Select **Continue**.
6. In the **Make/Manufacturer for {printer name}** section, in the **Make** list box, select the manufacturer of the printer. Select **Continue**.
7. In the **Model/Driver for {printer name}** section, in the **Model** list box, select the model number of your printer and select **Add Printer**.
8. In the **Authentication Required** dialog box, enter the root user name and password. Select **OK**.
9. If necessary, on the **Set Printer Options** web page, modify the settings as required and close the **CUPS Print Settings** interface.

TOPIC B Print Files

In the last topic, you configured a compatible printer to work with a Linux system. With the printer installed, you are ready to print files. In this topic, you will print files in the Linux system.

Printing a file is a routine task. Learning to print in Linux will save you time when documenting system information and changes. While you can print from the GUI, learning to print from the command line will give you greater freedom when printing from another location.

Printer Commands

Linux comprises various commands that facilitate the printing process. Some of the commands are described in the following table.

<i>Command</i>	<i>Description</i>
lp	Submits files for printing or alters a pending print job. The syntax of this command is <code>lp [options] {file name}</code> .
lpr	Submits files for printing. Files entered on the command line are sent to the specified printer or to the print queue if the printer is busy. If no files are entered on the command line, the <code>lpr</code> command reads the print file from the standard input. The syntax of this command is <code>lpr [options] [file name]</code> .
lpq	Displays the current print queue status. The syntax of this command is <code>lpq [options] {print queue name}</code> .
lprm	Cancels print jobs in the queue. The syntax of this command is <code>lprm {print job id}</code> .
lpc	Allows you to start or stop a printer, enable or disable queues, manage jobs in the queue, and obtain a status report on the printers and queues. The syntax of this command is <code>lpc [parameter]</code> .
lpstat	Displays the CUPS status information. The syntax of this command is <code>lpstat [options]</code> .
cancel	Deletes a print job from the print queue. The syntax of this command is <code>cancel [options]</code> .

The lpr Command

The *lpr* command comprises various options that allow you to specify the nature of the print output.

The `lpr` command options are described in the following table.

<i>Option</i>	<i>Used To</i>
-E	Force encryption when connecting to the server.
-P {destination}	Print files with the destination printer specified.
-# {copies}	Set the number of copies to print from 1 to 100.
-C {name}	Set the job name.
-l	Specify that the print file is already formatted and sent without being filtered to the destination.
-o {option}	Set a job option.
-p	Specify that the print file needs to be formatted with a shaded header that includes the date, time, job name, and page number of the file.
-r	Specify that the print files should be deleted after printing.

The `lpc` Command

The `lpc` command comprises various options that allow you to manage print jobs.

The `pr` Command

The `pr` command formats files before they are printed. By default, the `pr` command sends its output to the terminal screen. It is also used in combination with commands that send output to a printer.

The `pr` command formats the file's header containing the page number, file name, date, and time.

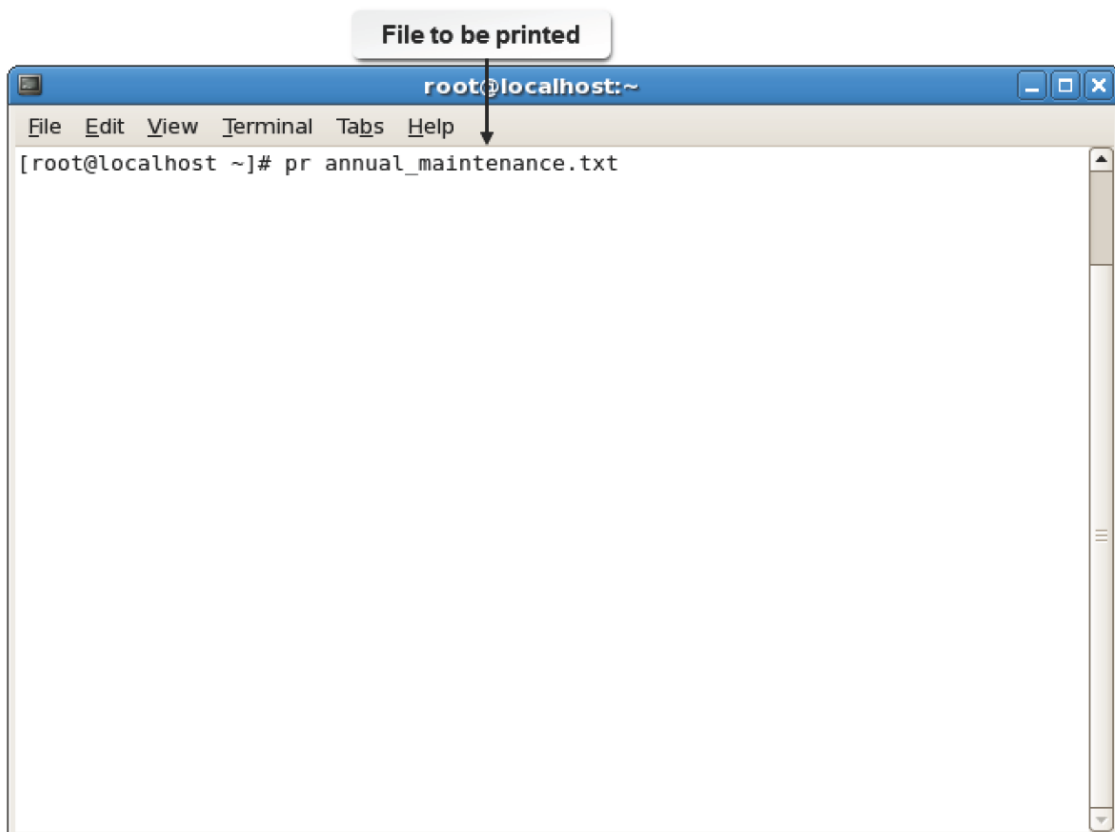


Figure 6-7: Printing a document using the `pr` command.

Syntax

The syntax of the `pr` command is `pr [options] {file name}`.

pr Command Options

There are many formatting options available for use with the pr command.

<i>Option</i>	<i>Description</i>
-{column}	Produces a multi-column output with the data arranged in columns.
-d	Produces a double-spaced output.
-m	Merges files.
-F	Ensures that the pages of the printout are separated by form feeds instead of new lines that include a 3-line page header on pages.
-l [#]	Changes the page length of the output, where # is the number of lines per page.
-h [new header text]	Allows you to change what is included in the header at the top of each page, where [new header text] replaces the file name in the default header contents.

Using pr with Piping

When you format your output with the pr command, you can send the formatted output to a file to be printed using the redirection operator (>) and pipe. For example, `pr -l45 sales > sales.out` creates the **sales.out** file, which is formatted and ready for printing. In addition, you can send the formatted file directly to the printer using the `lpr` command. For example, the `pr -l45 sales | lpr` command sends the formatted sales file directly to the default printer.

How to Print Files

Follow these general procedures to print files.

Print a File Without Text Formatting

To print a file without text formatting:

1. Change to the directory that contains the file you want to print.
2. Print the file using the `cat {file name} | lpr` command or using the `lpr {file name}` command.

Print a File with Text Formatting

To print a file with text formatting:

1. Change to the directory that contains the file you want to print.
2. To view the file with text formatting, enter `pr [options] {file name} | more`.
3. To print the file with text formatting, enter `pr [options] {file name} | lpr`.

Manage Print Jobs in the Print Queue in the CLI

To manage print jobs in the print queue in the CLI:

1. Log in as a user in the CLI.
2. Manage jobs in a queue.
 - To add a job to the print queue, enter `lpr {file name}`.

- To add a job to a specific print queue, enter `lpr -P {print queue name} {file name}`.
- To view all the print jobs in the queue, enter `lpq`.
- To view the print jobs in a specific print queue, enter `lpq -P {print queue name}`.
- To remove the desired print job, enter `lprm {print job id}`.

TOPIC C Configure Remote Printing

Now that you managed your local printer jobs and queues, you will now be able to apply those print techniques over a network. In this topic, you will connect and use remote printers.

Businesses and home users set up networks to transfer files. With this infrastructure in place, printing across networks is feasible. Being able to print with a remote printer allows you to minimize the number of printers required in your environment. You can have one printer for a larger number of users to share.

Print Servers

A **print server** is a computer that enables a network of users to access the central printer. The print server acts as a buffer, storing information to be printed until the printer is free. Print servers can be programmed to print jobs in the order in which they are received or in the order of priority.

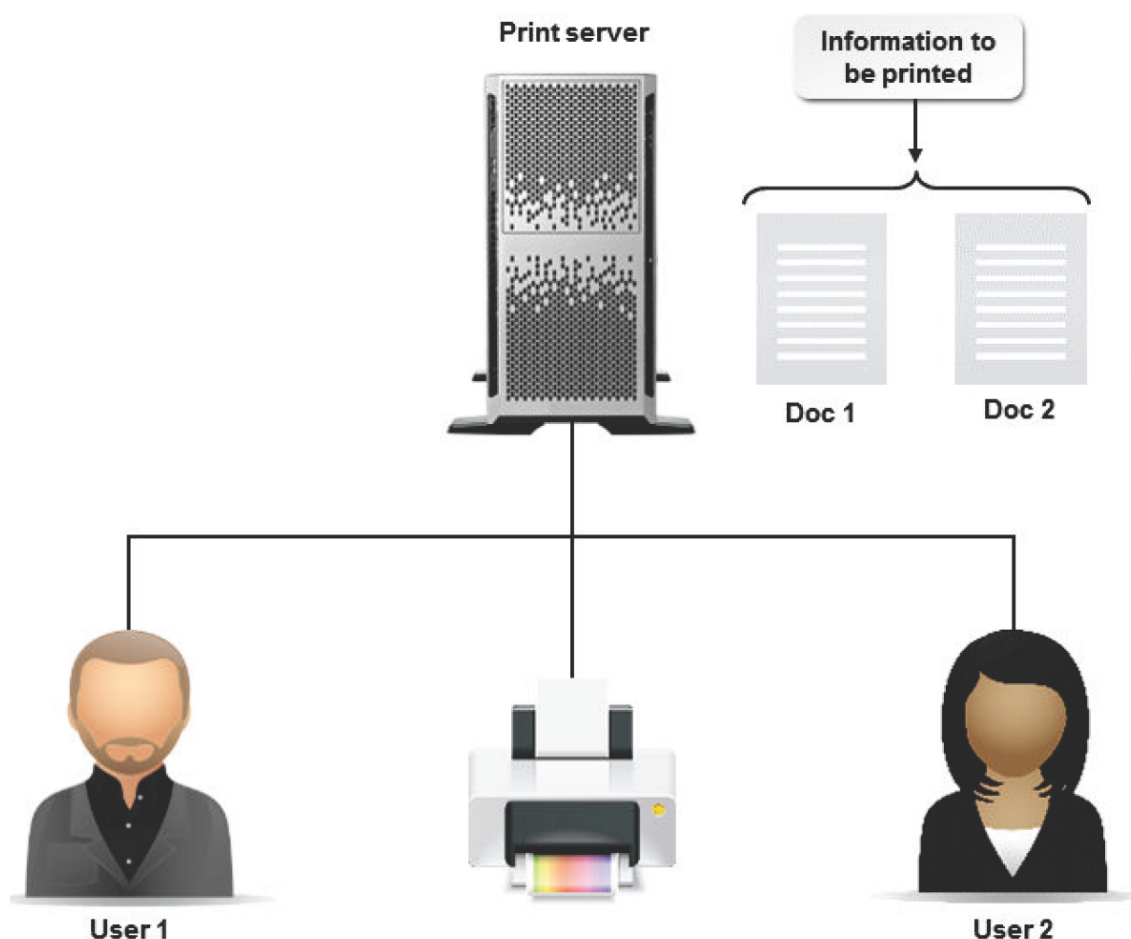


Figure 6-8: A print server manages print jobs over a network.

Remote Printing

In a network environment, users of a Linux system can print files using remote printers through the Linux print system. When you enter the `lpr` command, the file you specify is copied into the remote spool directory where it waits until the remote print server can print it.

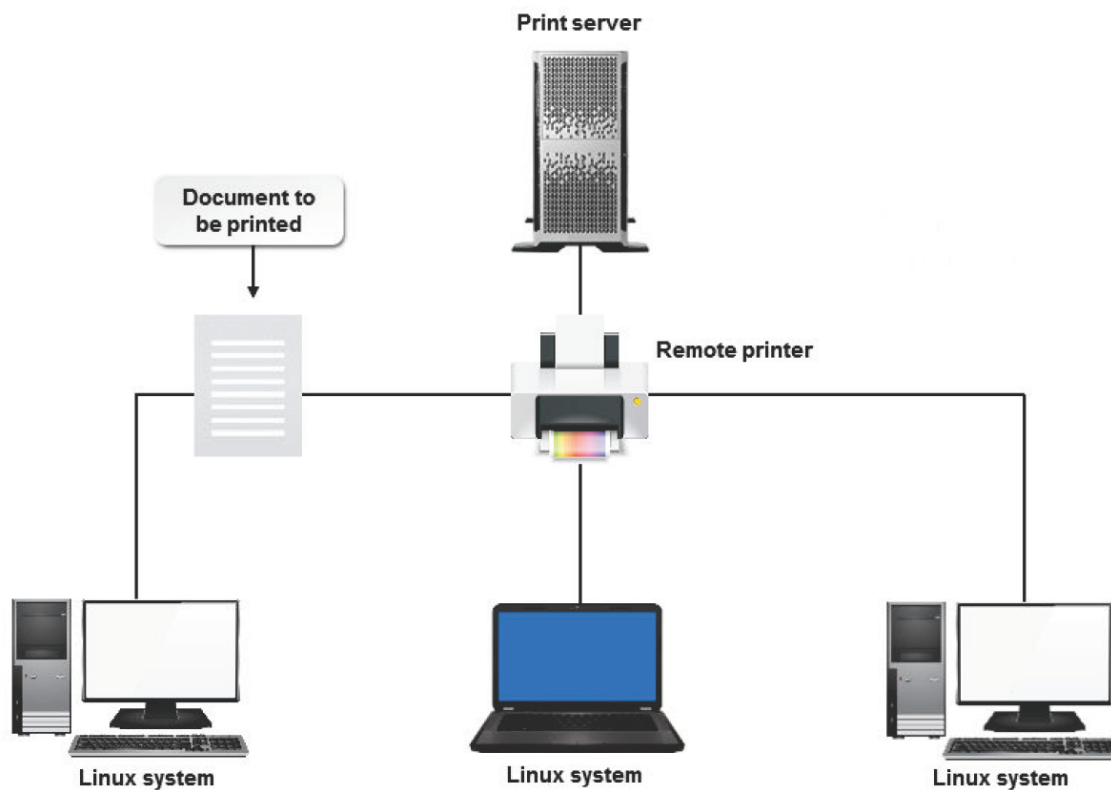


Figure 6-9: Printing on a network via the Linux print system.

Syntax

The syntax to print a document remotely is `lpr -P [printer name] [file name]`, where `[printer name]` is the name of the remote printer and `[file name]` is the name of the file you want to print.

Remote Printer Permissions

If you are setting up a remote printer, ensure that your system has the correct permissions to access the remote printer. The permissions are specified in either the `/etc/hosts.lpd` file or the `/etc/hosts.equiv` file, on the system to which the printer is attached. These files list the names of the remote systems that can use the local printer and can be modified to add or remove access to the printer.

Managing Remote Print Jobs

Just as you can manage local print jobs, such as removing jobs from a queue, holding jobs, or reordering jobs, you can also manage remote print jobs. You may not be able to do this as a regular user and may require someone with administrative access to the remote printer to do this for you.

However, you should be able to hold or delete your own jobs.

Samba

Samba is a suite of network sharing tools that help in the sharing of files and printers on a heterogeneous network, which consists of computers running on different operating systems.

Samba is an open source software application that provides enhanced interoperability with better performance and minimal maintenance. Using the Server Message Block (SMB) protocol, Samba enables Linux systems to communicate with computers running on other operating systems and share network resources such as printers.

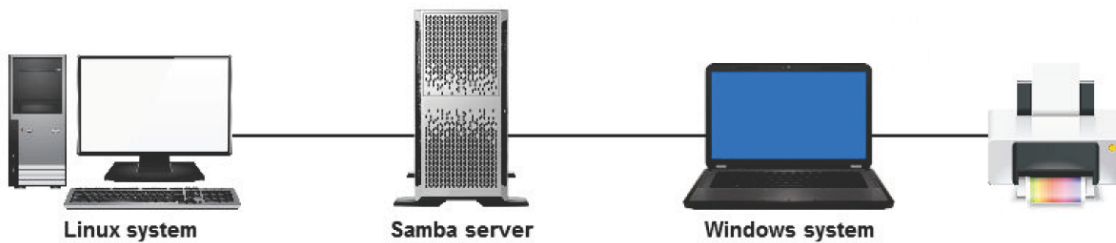


Figure 6-10: Samba allows a Linux system to access a printer connected to a Windows operating system.

Samba allows Linux to emulate some services that a Windows server provides. It allows a user to share resources between Linux and Windows machines. Samba also provides enhanced network security by allowing Active Directory (AD) support. AD helps provide authenticated user access and restricted access permissions. By using shared network resources, you can make printing in AD easy and secure.

SMB is a client-server protocol that is used to share and transfer files on a network. It allows a client on the network to send, print, or scan requests to the server. The server in turn, makes the device available to the client. SMB is most frequently used in computers that have Windows operating systems.

Configuring Samba

To configure Samba, you must edit the `/etc/samba/smb.conf` file. The following configuration

options must be set in the file: `load printers = yes` `printing = cups` `printcap name =`

`cups`. In the `[printers]` section, you should have something similar to the following:

```
[printers] comment = All Printers path = /var/spool/samba browseable = no public = yes guest ok = yes writeable = no printable = yes printer admin = root, @ntadmins
```

After modifying the file, the Samba service must be restarted for the changes to take effect.

The printers.conf File

The `printers.conf` file, which is stored in the `/etc/cups` directory, defines the set of local printers on a network as shared resources. The list of printers gets generated automatically using the `cupsd` daemon. It can be configured in such a way that it allows only the explicitly publicized printers. The file contains a set of directives that define the features of the printer being shared.

How to Configure Remote Printing

Follow these general procedures to configure remote printing.

Install Samba

To install Samba:

1. Log in as **root** in the GUI.
2. Mount the Linux installation CD-ROM that contains the latest version of Samba packages.
3. Navigate to the `/mnt/[cdrom or name of the media]/Packages` folder.
4. Select the `samba {package version}.rpm` and `perl - Convert -ASN1 - {package version}.rpm` files.
5. Right-click and select **Open with Software Installer**.
6. In the **Installing packages** window, verify that the two packages are listed and select **Apply** to start installing

the packages.

7. If necessary, to install the two packages, in the **Unable to verify {package name}** message box, select **Install anyway**.
8. To complete the installation, in the **Software installed successfully** message box, select **OK**.

Share a Printer Using Samba

To share a printer using Samba:

1. Open the **/etc/samba/smb.conf** file.
2. Define a share for a printer.
 - Specify the share name within brackets. *[share name]*.
 - Define the path to the spool file. *path = {path to the spool file}*.
 - Specify the printer name. *printer = {printer name}*.
 - Specify if the guest users are allowed to print. *public = {yes | no}*.
 - Specify the printing access for users. *printable = {yes |no}*.
 - If necessary, include a comment describing the shared printer. *comment = {Printer Description}*.
3. Save and close the file.
4. To verify that the **smb.conf** file is formed, enter `testparm /etc/samba/smb.conf`.
5. To start the service when the system boots, enter `chkconfig smb on`.
6. To restart the Samba service, enter `service smb restart`.
7. To view the IP address of the system to which the printer is connected, enter `ifconfig`.
8. Verify that the printer share you created is accessible from other Linux systems.

ACTIVITY 6-1

Printing Files Review

Scenario

Answer the following review questions.

1. Why would it be beneficial to have a local printer vs. a network printer?
2. What do you think the best way to manage print queues will be for your organization?

Summary

In this lesson, you worked with Linux printing services. You configured both local and remote printers, printed files using various printer commands, and managed print jobs and queues. You will now be able to set up printer connections and print backup records of files.

7 Managing Packages

Lesson Time: 2 hours

Lesson Introduction

You explored the basic Linux® environment, worked with files, and printed them. Now, you are ready to install packages to facilitate the distribution and installation of software. In this lesson, you will manage packages.

You will need to install software on your system to perform required tasks effectively. To do this, you need to learn about packages and package managers and how to install them on your system. Unless the packages are fully installed, the desired software will not function.

Lesson Objectives

In this lesson, you will manage packages. You will:

- Manage packages using the RPM package manager.
- Verify packages.
- Upgrade and refresh packages.
- Configure repositories.
- Manage packages using the YUM package manager.
- perform advanced package and application management.

TOPIC A Manage Packages Using RPM

Installing software on your system will increase the capabilities of your computer and enable you to perform a new set of tasks or perform a common set of tasks much faster. You need to know how to manage packages before you install software. In this topic, you will manage packages using the RPM package manager.

As a Linux professional, you will need to install software on systems. Software is a collection of packages. You can install the software only if you know how to add these packages on your system.

Even if one package is not installed correctly, the software will not work. Installation of these packages is facilitated by package managers. Therefore, it is necessary to know about packages and package managers.

Packages

A **package** is a collection of classes, functions, or procedures that can be imported as a unit.

Packages include all files required to run an application. Each package is compiled specifically for each Linux distribution and type of system. Packages are of different types, depending on the applications for which they are used.

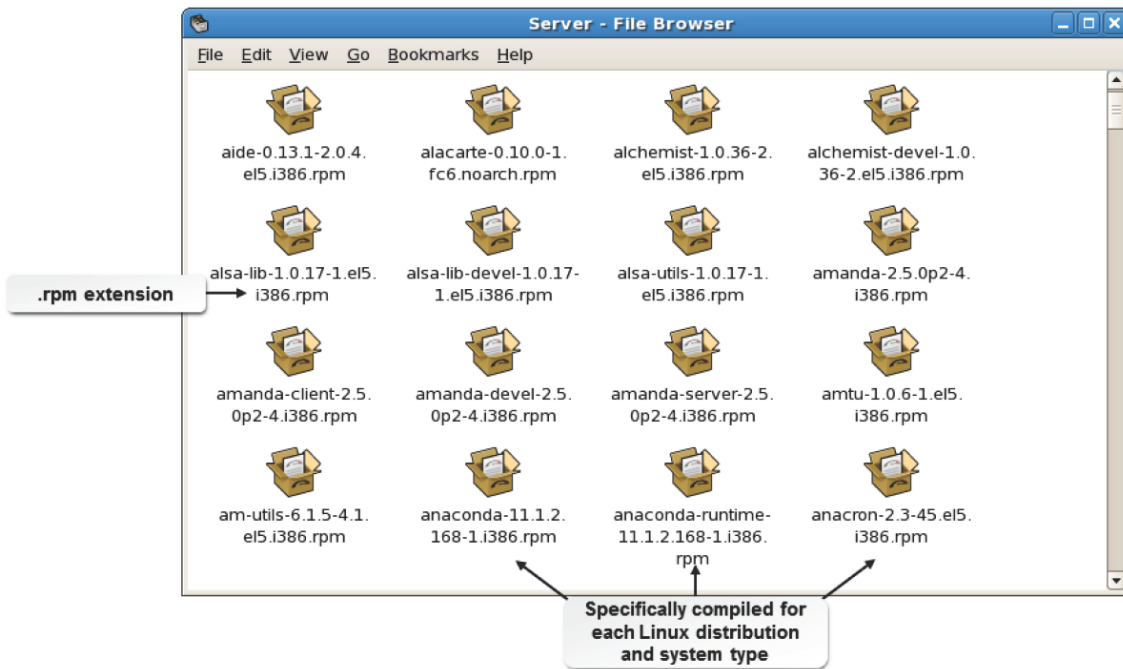


Figure 7-1: Packages on a Linux system.

Package Managers

A **package manager** is a tool that enables you to search for packages and upgrade or remove them.

It tracks the files that are provided with each package. Querying options are also provided by a package manager to list the installed packages and their characteristics. The naming convention followed by package managers for package files is **name-version-release.architecture.rpm**. RPM and YUM (Yellow dog Updater, Modified) are examples of package managers.



Figure 7-2: Installing a package using the RPM package manager.

Documenting Changes to Installed Packages

It is recommended that you document any change you make to installed packages. This will help you troubleshoot issues and track the versions that were previously installed.

Dependencies

Dependencies are the packages that a target package depends on for its functionality. Dependency chains can run on for many levels. For example, package A will be installed only after package B is installed. Similarly, package B will be installed only after package C is installed. Package managers can fetch the required packages in an automated manner, saving time and effort. Dependency management is a major function of package managers.



Note: RPM has several complementary utilities, such as up2date and yum, to manage dependencies.

```
root@localhost:~/rhelsource/Server
File Edit View Terminal Tabs Help
[root@localhost ~]# rpm -ivh httpd-manual-2.2.3-22.el5.i386.rpm
warning: httpd-manual-2.2.3-22.el5.i386.rpm: Header V3 DSA signature: NOKEY, key
ID 37017186
error: Failed dependencies:
→ httpd = 2.2.3-22.el5 is needed by httpd-manual-2.2.3-22.el5.i386
[root@localhost ~]#
```

Failed package installation

Figure 7-3: Package installation fails due to unavailability of dependency packages.

The RPM Package Manager

The **RPM Package Manager (RPM)**, developed by Red Hat®, is a tool for maintaining packages.

By providing a standard software packaging format, RPM enables easy administration and maintenance of Linux systems and servers. RPM provides a standard installation mechanism, information about installed packages, and a method for uninstalling and upgrading existing packages.

```
root@localhost:/rhelsource/Server
File Edit View Terminal Tabs Help
[root@localhost ~]# rpm | more
RPM version 4.4.2.3
Copyright (C) 1998-2002 - Red Hat, Inc.
This program may be freely redistributed under the terms of the GNU GPL

Usage: rpm [-aKfgpWHqV] [-aKfgpWHqVcdils] [-aKfgpWHqVcdilsaKfgpWHqV] [-aKfgpWHqVcdilsaKfgpWHqV] [-aKfgpWHqVcdilsaKfgpWHqV] [-aKfgpWHqVcdilsaKfgpWHqVK] [-aKfgpWHqVcdilsaKfgpWHqVK] [-aKfgpWHqVcdilsaKfgpWHqVKi] [-aKfgpWHqVcdilsaKfgpWHqVKiv] [-aKfgpWHqVcdilsaKfgpWHqVKiv] [-aKfgpWHqVcdilsaKfgpWHqVKiv?] [-a|--all] [-f|--file] [-g|--group]
[-p|--package] [-W|--ftswalk] [--pkgid] [--hrid] [--fileid]
[--specfile] [--triggeredby] [--whatrequires] [--whatprovides]
[--nomanifest] [-c|--configfiles] [-d|--docfiles] [--dump] [-l|--list]
[--queryformat=QUERYFORMAT] [-s|--state] [--nomd5] [--nofiles]
[--nodeps] [--noscript] [--comfollow] [--logical] [--nochdir]
[--nostat] [--physical] [--seedot] [--xdev] [--whiteout]
[--addsign] [-K|--checksig] [--delsign] [--import] [--resign]
[--nodigest] [--nosignature] [--initdb] [--rebuilddb] [--aid]
[--allfiles] [--allmatches] [--badreloc] [-e|--erase <package>+]
[--excludedocs] [--excludepath=<path>] [--fileconflicts] [--force]
[-F|--freshen <packagefile>+] [-h|--hash] [--ignorearch] [--ignoreos]
[--ignoresize] [-i|--install] [--justdb] [--nodeps] [--nomd5]
[--nocontexts] [--noorder] [--nosuggest] [--noscripts]
[--notriggers] [--oldpackage] [--percent] [--prefix=<dir>]
```

Figure 7-4: Various options of the RPM tool are displayed.



Note: RPM is distributed under the GNU General Public License (GPL) and can be used with many distributions of Linux and even with other UNIX implementations.

The RPMS Directory

The Red Hat distribution includes the **RPMS** directory containing packages. You can also find packages on the Internet and FTP sites. One website where you can find packages is [http:// rpmfind.net](http://rpmfind.net).

Installing Packages

When you install Linux, you may install all the packages it comes with. However, it is better to install only the packages you need. Later, when you need to install additional packages, you can use your CD or DVD (or whatever source you used) to obtain additional packages. You can check whether the updated versions of the package are released and install a recent version.

The /usr/lib/rpm/* Directory

The **/usr/lib/rpm/*** directory contains the RPM tools required to manage the RPM packages.

The **/var/lib/rpm/*** directory contains the RPM database of the installed packages. By default, the **rpmrc** file, which is the global RPM configuration file, is located in the **/usr/lib/rpm/*** directory.

The **rpmrc** file contains information of the RPM architecture compatibility. If you want the RPM settings to be applicable for a system-wide configuration, place the **rpmrc** file in the **/etc** directory.

If the **rpmrc** file is placed as **.rpmrc** in the home directory of any user, then the RPM settings will be applicable only for that specific user.

RPM Commands

Common RPM package management commands enable you to perform package management tasks.

Frequently used RPM package management commands are given in the following table.

Command	Enables You To
<code>rpm -i {package name}</code>	Install a package.
<code>rpm -F {package name}</code>	Reinstall a package.
<code>rpm -U {package name}</code>	Upgrade a package.
<code>rpm -e {package name}</code>	Remove a package.



Note: Appending `h` to some of the existing options of the `rpm` command will print 50 hashes as the package archive is unpacked. Appending `v` will give you a verbose status of the package management task that is performed. For example, `rpm -ivh {package name}` will indicate the status of the installation of the RPM package.

RPM Components

The RPM package manager contains a number of components. Using these components, you can maintain a list of packages that are installed on a system.

Component	Description
The RPM local database	Tracks packages that are installed on a system.
The RPM package	Contains many executables and scripts required to install packages.
YUM	Acts as the front-end package installer for RPM.
RPM package files	Contains source code for packages.

RPM Queries

An RPM query is a function that is used to query RPM for information on both installed and uninstalled packages. There are various options, which give distinct outputs, for the `rpm -q` command.

```
root@localhost:rhelsource/Server
File Edit View Terminal Tabs Help
[root@localhost ~]# rpm -qi samba
Name       : samba                               Relocations: (not relocatable)
Version    : 3.0.33                             Vendor: Red Hat, Inc.
Release    : 3.7.el5                             Build Date: Tue 09 Dec 2008 11:03:42
          AM EST
Install Date: Thu 25 Jun 2009 11:37:44 AM EDT      Build Host: hs20-bc1-7.build.
redhat.com
Group      : System Environment/Daemons          Source RPM: samba-3.0.33-3.7.el5.src
.rpm
Size       : 31830025                             License: GNU GPL Version 2
Signature  : DSA/SHA1, Mon 15 Dec 2008 06:05:28 PM EST, Key ID 5326810137017186
Packager   : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL        : http://www.samba.org/
Summary    : The Samba SMB server.
Description:

Samba is the suite of programs by which a lot of PC-related machines
share files, printers, and other information (such as lists of
available files and printers). The Windows NT, OS/2, and Linux
operating systems support this natively, and add-on packages can
enable the same thing for DOS, Windows, VMS, UNIX of all kinds, MVS,
and more. This package provides an SMB server that can be used to
provide network services to SMB (sometimes called "Lan Manager")
clients. Samba uses NetBIOS over TCP/IP (NetBT) protocols and does NOT
```

Figure 7-5: Basic details of the Samba package obtained using an RPM query.

The options for the RPM query are listed in the following.

Option	Enables You To
<code>rpm -qa</code>	List all the packages that are installed on your system.

Option	Enables You To
rpm -qc {package name}	List the configuration files of a specified package.
rpm -qi {package name}	Give the basic details of a package such as the installed date, size, signature, and summary.
rpm -ql {package name}	List the files in a package.
rpm -qR {package name}	List the package dependencies.
rpm -qa grep {package name}	Send the results of the rpm query command to the grep command to search the results for a specific package.
rpm -qf {file name}	Find which package provides a specific file.
rpm -qpl {package name}	List all the files in a package yet to be installed.
rpm -qp {package name}	List the packages that start with a particular alphabet or name. For example, in the syntax, if you substitute {package name} with mysql*, you will get a list of all the packages that start with mysql.

Syntax

The syntax of an RPM query is rpm -q {what_packages} {what_information}.

The rpm2cpio Utility

The [rpm2cpio](#) utility enables you to extract individual files from RPM packages.

How to Manage Packages Using RPM

Follow these general procedures to manage packages using RPM.

Install Packages

To install packages:

1. Download the **rpm** file that you want to install.
2. To install the package, use the rpm -ivh {package name} command.
3. If necessary, download the dependency packages.

Uninstall Packages

To uninstall packages:

1. To search for the rpm package that you want to uninstall, use the rpm -qi {package name} command.
2. To uninstall the rpm package, use the rpm -e {package name} command.

TOPIC B Verify Packages

In the last topic, you managed packages using the RPM package manager. As a Linux system administrator, you may have to verify and repair corrupt packages. In this topic, you will verify packages.

As a Linux administrator, you may face situations where due to accidental deletion of important files, the installed packages may stop working. Generally, in such situations, you may need to spend time to identify the missing files

and troubleshoot package installations. Because there are a number of packages on the Linux system, it becomes difficult to individually identify the corrupted packages.

You can use the rpm package management tools to easily locate such corrupted packages and save time.

RPM Verification

RPM verification compares the existing packages with the RPM package database and returns the missing or corrupt packages. Various options allow you to verify specific information in the package.

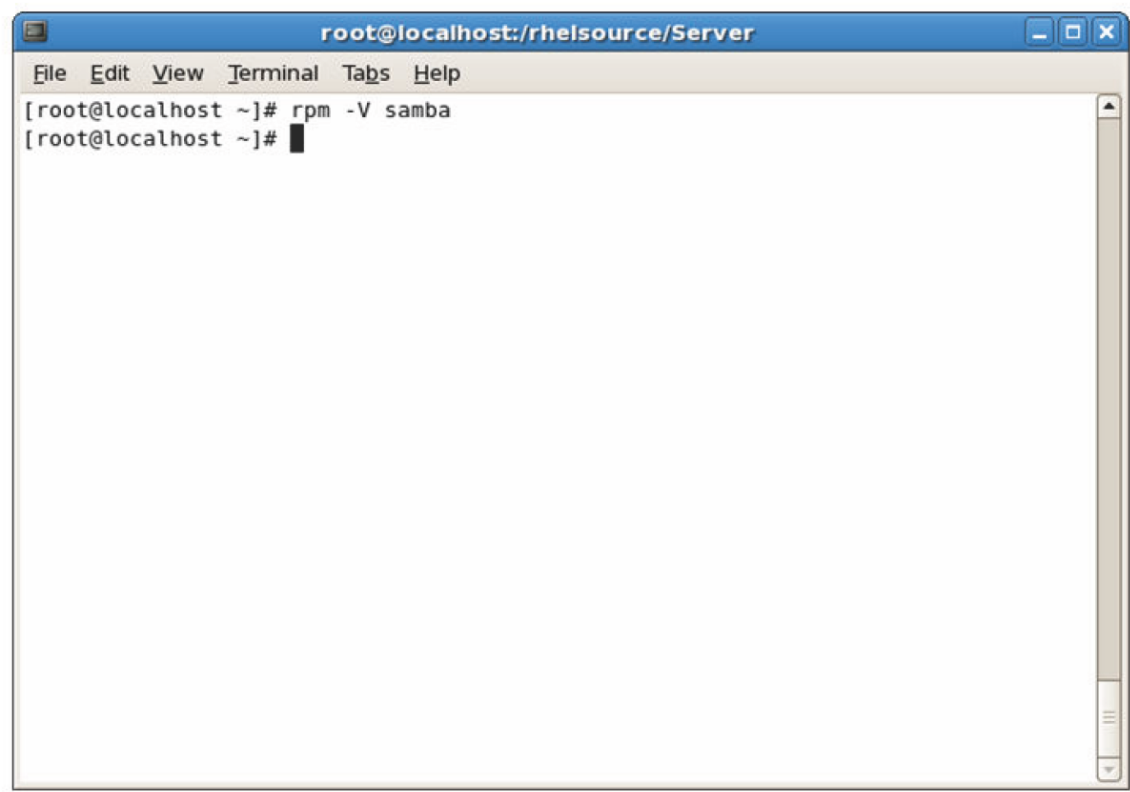


Figure 7-6: Verifying the Samba package.

RPM verification options are listed in the following table.


<i>Option</i>	<i>Enables You To</i>
rpm -Va	Verify all the installed packages.
rpm -V {package name}	Verify a specific package.
rpm -V -f {file name}{package name}	Verify a specific file in the package.

Syntax

The syntax for RPM verification is rpm -V {package name}.

Importance of Verifying Packages

The integrity of a package can also be verified via the RPM system. This verification process can be used to ensure that the package files are present in the correct directories without modifications and have the proper permissions. More specifically, RPM checks the size, MD5 checksum, permissions, type, owner, and group of each file in the package.



Note: MD5 checksum is like a file's fingerprint; it can be used to uniquely identify the file and verify its integrity.

Verification Error Codes

You can check either a specific package or all the packages. If everything is fine with the packages, no messages are displayed and you return to the command prompt. If a problem is detected, an eight-character string is displayed to alert you of the change. Based on this output, you can determine if the package needs to be reinstalled. In the following table, the verification error codes are listed in the same order in which they appear in case there is an error during package verification.

<i>Error Code</i>	<i>Description</i>
S	Change in file size since installation.
M	Different permission or file type mode.
5	MD5 checksum test failed.
D	Device attribute error.
L	Symbolic link errors.
U	Different user setting from the original.
G	Different group setting from the original.
T	Current file modification time does not match the original file modification time.

Any of the eight characters appearing in the output will indicate that the particular test has failed. A period (.) will be displayed if the test is successful. A question mark (?) will be displayed if a test is skipped by the command.

How to Verify Packages

Follow these general procedures to verify packages.

Verify Packages

To verify packages:

1. Log in as **root**.
2. Verify installed packages.
 - To verify all the rpm packages, use the `rpm -Va` command.
 - To verify an individual rpm package, use the `rpm -Vv {package name}` command.
 - To verify a specific file in the package, use the `rpm -V -f {file name} -p {package name}` command.

TOPIC C Upgrade Packages

Now that you know the methods for verifying packages, you can learn to upgrade packages. In this topic, you will upgrade packages by updating and refreshing installed packages.

One of the aspects of system administration is keeping the system's software up-to-date. Many applications are in active development and new releases are made available on a regular basis. These new releases may add functionality, fix bugs in older versions, or provide important security updates.

RPM has the ability to quickly and easily upgrade software packages. This will save you from having to uninstall and reinstall a package, to have the newest version. Upgrading packages allows you to have the newest features of

an application in the shortest amount of time.

Upgrade/Freshen Packages

Packages can be easily upgraded by using the upgrade or freshen option.

<i>Option</i>	<i>Description</i>
Upgrade	Checks package versions against the package versions installed already. If the package is found, the package will be upgraded. If the package is not found, the package will be installed. The syntax that is used to update packages is <code>rpm -U {package name}</code> .
Freshen	Checks package versions against the package versions installed already. If the package is found, the package will be updated. If the package is not found, the package will not be installed. The syntax that is used to freshen packages is <code>rpm -F {package name}</code> .

Freshen Packages

Entering `rpm -Fvh *.rpm` automatically upgrades only those packages that are already installed.

How to Upgrade and Refresh Packages

Follow these general procedures to upgrade and refresh packages.

Upgrade Packages

To upgrade packages:

1. Download the updated package.
2. Update the existing package using the `rpm -Uvh {package name}` command.
3. To verify that the package is updated, use the `rpm -qi {package name}` command.

Freshen Packages

To freshen packages:

1. Download the updated package.
2. To freshen the existing package, use the `rpm -Fvh {package name}` command.
3. To verify that the package is updated, use the `rpm -qi {package name}` command.

TOPIC D Configure Repositories

In the last topic, you upgraded and refreshed the installed packages. When managing a network, you may have to update systems with the latest packages. Ultimately, you need to know where to obtain these packages. In this topic, you will examine repositories and how to use them to update systems.

In Linux, there are a number of packages that keep evolving with new versions of the same package being available. As a Linux administrator, you need to ensure that the latest packages are installed on a system by updating only the specific packages instead of reinstalling the entire Linux OS every time. To ensure this, you must know where the packages are available and how they can be downloaded. Updating systems with the latest

packages will help users to make use of the newer features of the packages and be able to perform their tasks efficiently.

Repositories

A **repository** is a database that holds source code and compilations of Linux software and applications. There are two types of repositories; local and online. Software can be installed on a system only when repositories for the software are present on the system. The packages for the software are found in their respective repositories and are directly installed from them.

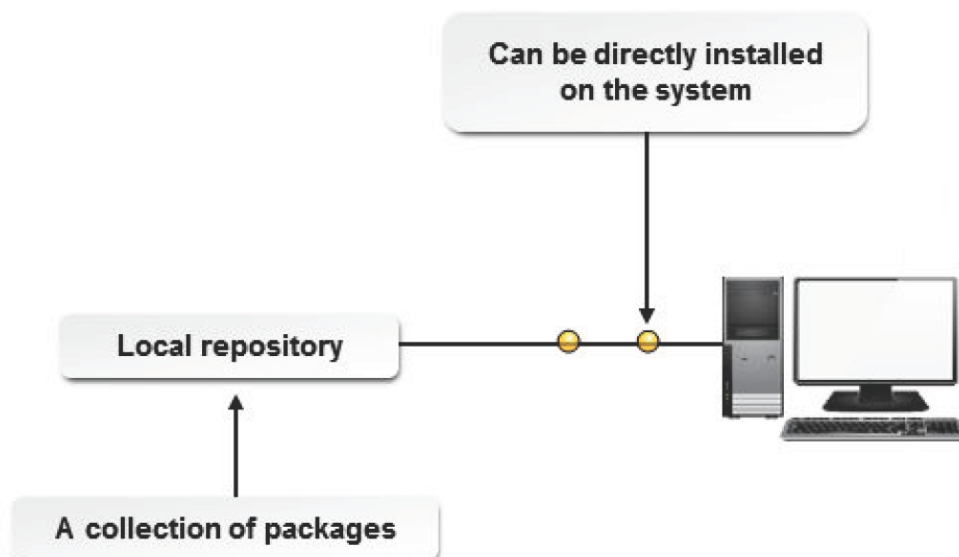


Figure 7-7: A local repository.

Types of Repositories

There are two types of repositories, **online repositories** and **local** or **private repositories**. Online repositories are found on the Internet. Packages can be directly downloaded from the Internet and installed on a system. Local or private repositories are stored on a system. The process of updating systems is greatly facilitated by repositories because the source files are readily available. Repositories make it easy for system administrators to update multiple systems simultaneously.

The **createrepo** command is a utility that generates a database of information stored in each RPM in a directory. This streamlines remote access of the packages held in repositories.

GPG

The GNU Privacy Guard (GPG), also known as GnuPG, is a complete and free software implementation of the OpenPGP standard used for encrypting and signing packages or files.

GnuPG is comparable to PGP for most applications. The encryption or signature is used to verify the authenticity of any file shared on the network.

How to Configure Repositories

Follow these general procedures to configure repositories.

Create a Private Repository

To create a private repository:

1. Log in as **root**.

2. To create a directory, on the terminal, enter `mkdir /{directory name}`.
3. Populate the directory with packages.
4. To create a private repository, enter `createrepo -v /{directory name}`.
5. If you add or remove any package from the directory, run the `createrepo` command again.

Configure Additional Repositories

To configure additional repositories:

1. Log in as **root**.
2. To navigate to the `/etc/yum.repos.d` directory, enter `cd /etc/yum.repos.d`.
3. To create a file, enter `vi {file name}`.
4. Switch to insert mode.
5. Type the required information.
 - To set the repository name, enter **`[repository name]`**.
 - To give a description of the repository, enter **`name=description of the repository`**.
 - To set the repository's base URL, enter **`baseurl = {URL of the repository}`**.
 - To control the status of the repository, enter **`enabled = { 0 | 1 }`**.
 - To control the GPG signature verification, enter **`gpgcheck = { 0 | 1 }`**.
6. Save and close the file.

TOPIC E Manage Packages Using YUM

Earlier in this lesson, you managed package installation on a single computer using RPM. As a system administrator, your task involves installing software on multiple systems simultaneously within a short period of time. In this topic, you will manage package installation using YUM.

As a system administrator, you will be dealing with multiple systems at the same time. It is necessary that you install packages on all systems simultaneously and in the shortest possible time. Knowledge about the YUM package manager will help you install and manage packages on multiple systems simultaneously.

The YUM Package Manager

The **Yellow dog Updater, Modified (YUM)** is a package manager that is used to update, install, and manage packages. YUM automatically detects and configures the dependencies for software packages and maintains a database of the installed software. YUM is widely used by system administrators because it is easy to work with. It supports both local and online repositories.

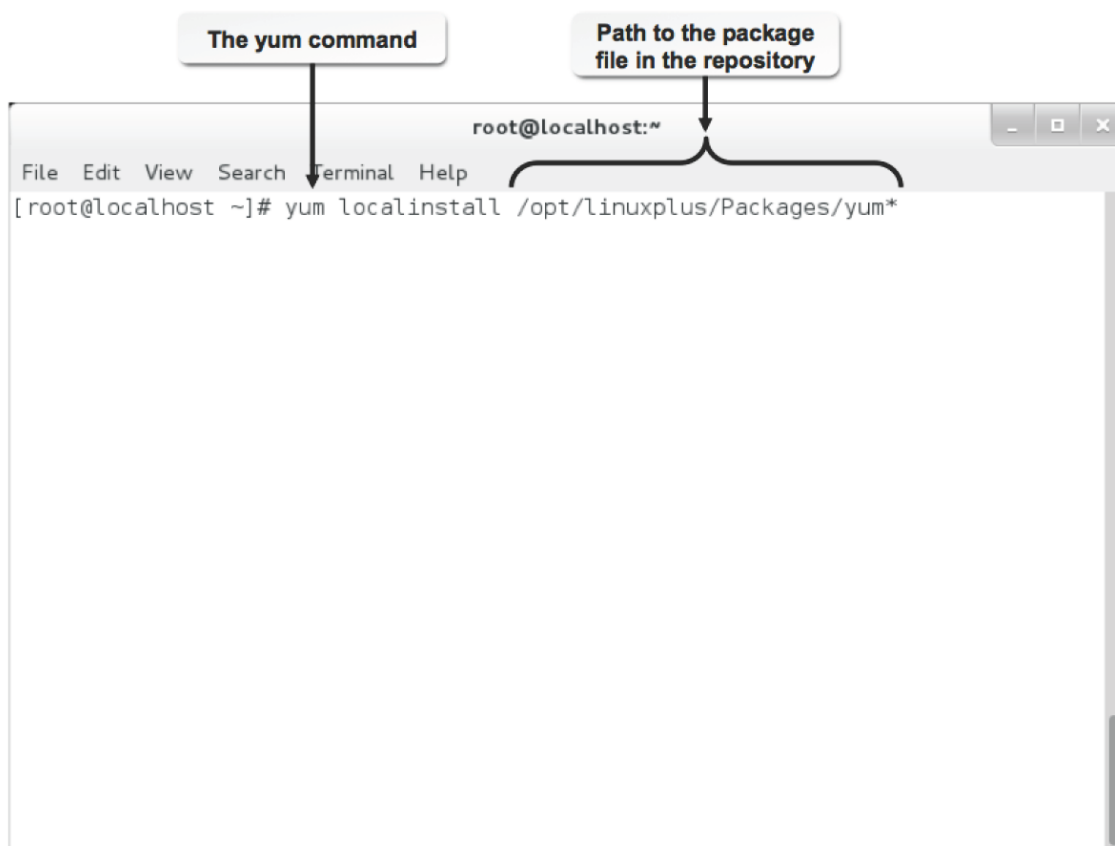


Figure 7-8: Installing a package from a local repository using the YUM package manager.

YUM Commands

YUM has various commands that can be used to maintain packages.

Command	Enables You To
install	Install a package.
update	Update packages. The command will update all packages when a package is not specified.
check-update	Check for available updates.
remove	Remove the specified packages.
list	Display the details of the specified package. When a package is not specified, it lists the status of all the packages on the system.
info	Display a brief description of the specified package.
localinstall	Install packages from a local repository.

Syntax

The syntax of the YUM command is `yum [options] {command} {package name}`.

The yumdownloader Utility

The yumdownloader utility allows you to download rpm packages from yum repositories. This utility is part of the yum-utils package.

The syntax of the yumdownloader utility is `yumdownloader [options] {package name}`.

How to Manage Packages Using YUM

Follow these general procedures to manage packages using YUM.

Manage Packages Using YUM

To manage packages using YUM:

1. Log in as **root**.
2. Manage packages using YUM.
 - To install packages using YUM, enter `yum install {package name}`.
 - To remove packages using YUM, enter `yum remove {package name}`.
 - To display a package's description, enter `yum info {package name}`.
 - To update the system with the specified package, enter `yum update {package name}`.

TOPIC F Advanced Package and Application Management

In the previous topic, you managed packages using the YUM package manager. You may also need to manage packages in other distributions of Linux, which have their own package managers. In this topic, you will perform advanced package and application management.

Linux supports different kinds of packages. As a Linux administrator, you need to be aware of the package managers that allow you to handle specific packages. The standard package managers have restrictions in the number of file types they can recognize, and they might also be available only in certain Linux distributions. While it may be possible that you can install the relevant package managers to handle the required file types, the number of file types make it practically impossible.

To overcome this, you need to be able to handle generic files, without relying on package managers.

As a Linux administrator, your ability to work with source files will give you the skill and flexibility to troubleshoot and manage packages in any Linux distribution. You will be in a position to install applications, regardless of the availability of specific package managers.

As a system administrator, you will also need to handle system software and also other applications that you install. Some of these applications may require similar services to ensure their proper working. It will be helpful to have these services easily accessible to applications with similar needs.

Managing shared libraries can help accomplish this with ease.

The Debian Archive Package Installation Process

The Debian Archive Package Installation process consists of three stages: check for dependency, unpack, and configure.


1. In the **check for dependency** stage, the package manager checks the specified Debian archive package for the numerous dependencies required.
2. In the **unpack** stage, the Debian archive package and its dependent packages are unpacked into the filesystem of the hard disk.
3. In the final **configure** stage, the unpacked files can be configured with the default or customized values to suit your requirements. In addition to this, you can choose to reconfigure the package and its dependencies later.



Figure 7-9: Stages in the Debian Archive Package Installation process.

Debian Package Management Tools

The Debian package management tools are a set of tools for package management of Debian-based Linux distributions. These tools are used to install, list, and remove packages conforming to the Debian software package standard for Debian, Ubuntu, and related Linux distributions.

	<p>Note: The Advanced Package Tool (APT) is a front end for the Debian package management tools, designed to make Debian more user friendly.</p>
---	---

The tools included in the suite are listed in the following table.

<i>Tool</i>	<i>Description</i>
dpkg	It is the main package management program. Its main purpose is to install and remove Debian packages.
dpkg-deb	It is the archive manipulation tool of the Debian binary package. It is used to extract the DEB package contents from a directory and display package information. It is also used to collect and remove information about Debian archives.
dpkg-reconfigure	It is a tool that allows you to upgrade the installed packages. It enables you to specify options similar to the original installation of the package. Additionally, you can select the front-end interface application for this tool.
dpkg-split	It splits packages into smaller parts. It is useful for splitting packages into sizes of 1.44 MB, so that they can fit into a series of floppy disks.
dselect	It is a menu-driven, front-end text-based interface of the dpkg package. Through this utility, you can install and remove packages.

Debian Archive Package Management Commands

Debian package management commands can be used to get information about Debian archive packages.

These commands have various options.

<i>Option</i>	<i>Used To</i>
dpkg -p [Debian package name]	View the version, dependencies, and integrity of the package.
dpkg -L [Debian package name]	List the contents of the package.
dpkg -l [Debian package name]	View the installation status.
dpkg --get-to-unpack	Find the packages that are yet to be installed.
dpkg -S [file or package name]	Find packages containing specific files or software.

The apt-get Command

The apt-get command is used to install or upgrade packages through the Internet or from the distribution CD on Debian, Ubuntu, or related Linux distribution. While installing or upgrading packages, the apt-get command accesses the website or the CD-ROM listed in the **/etc/apt/sources.list** file.

The apt-get command has various options.

<i>Option</i>	<i>Used To</i>
apt-get install {Debian package name}	Install packages.
apt-get remove {Debian package name}	Uninstall packages.
apt-get update	Update the list of new packages available.
apt-get upgrade [Debian package name]	Upgrade packages.

Syntax

The syntax of the apt-get command is `apt-get [options] {command}`.

The apt.conf File

The **apt.conf** file is the configuration file for the apt-get command. This file stores additional information such as the number of attempts to be made while downloading packages and the available cache memory.

The /var/lib/dpkg/* Directory

The **/var/lib/dpkg/*** directory contains the Debian database of the installed packages. By default, the **dpkg.cfg** file, which is the dpkg configuration file, is located in the **/etc/dpkg** directory. The dpkg configuration file may contain all the dpkg command line options.

aptitude

aptitude is a text based front-end tool for the APT package manager. It is used in the same manner as the apt-get command.

Alien

Alien is a program in Debian and Ubuntu Linux that converts packages in other Linux distribution file formats to Debian Package (dpkg) format. It supports conversion among packages such as Linux Standard Base, RPM, deb, Stampede, and Slackware.

For instance, the `alien --to-deb {file name}` command will convert an .rpm package into a .deb package.

How to Manage Packages Using the Debian Package Manager

Follow these general procedures to manage packets using the Debian Package Manager.

Install Packages Using the Debian Package Manager

To install packages using the Debian package manager:

1. Ensure that Debian Linux is installed on the system.
2. Log in to the CLI as **root**, or in the GUI, display the terminal window.
3. To install the Debian package, on the terminal, enter `dpkg -i {Debian package file}`.

Upgrade Packages Using the Debian Package Manager

To upgrade packages using the Debian package manager:

1. Log in to the CLI as **root**, or in the GUI, display the terminal window.
2. Upgrade a Debian package.
 - Enter `dpkg --update-avail {Debian package file}`.
 - Enter `dpkg-reconfigure {Debian package name}`.

Uninstall Packages Using the Debian Package Manager

To uninstall packages using the Debian package manager:

1. Log in to the CLI as **root**.
2. To uninstall packages, enter `dpkg -r {Debian package name}`.

Manage Packages Using the apt-get Command

To manage packages using the apt-get command:

1. Log in as **root**.
2. To search for files that match a specific pattern or a specific package, enter `apt-cache search {regular expression}{package name}`.
3. Manage packages using suitable commands.
 - To install a Debian package, enter `apt-get install {Debian package name}`.
 - If necessary, to update the list of new packages available, enter `apt-get update`.
 - To upgrade a Debian package, enter `apt-get upgrade [Debian package name]`.
 - To uninstall a Debian package, enter `apt-get remove {Debian package name}`.

Install Alien Packages Using the alien Command

To install Alien packages using the alien command:

1. Log in to the CLI as **root**.
2. To install packages belonging to other distributions, type `alien -i {package file name}`.

makefile

Makefile is a description file that contains the details of files, dependencies, and rules with which an executable application is built. It is used to configure, compile, and install the application or driver.

System built-in rules for maintaining, updating, and regenerating groups of programs are overridden by the contents of *makefile*.

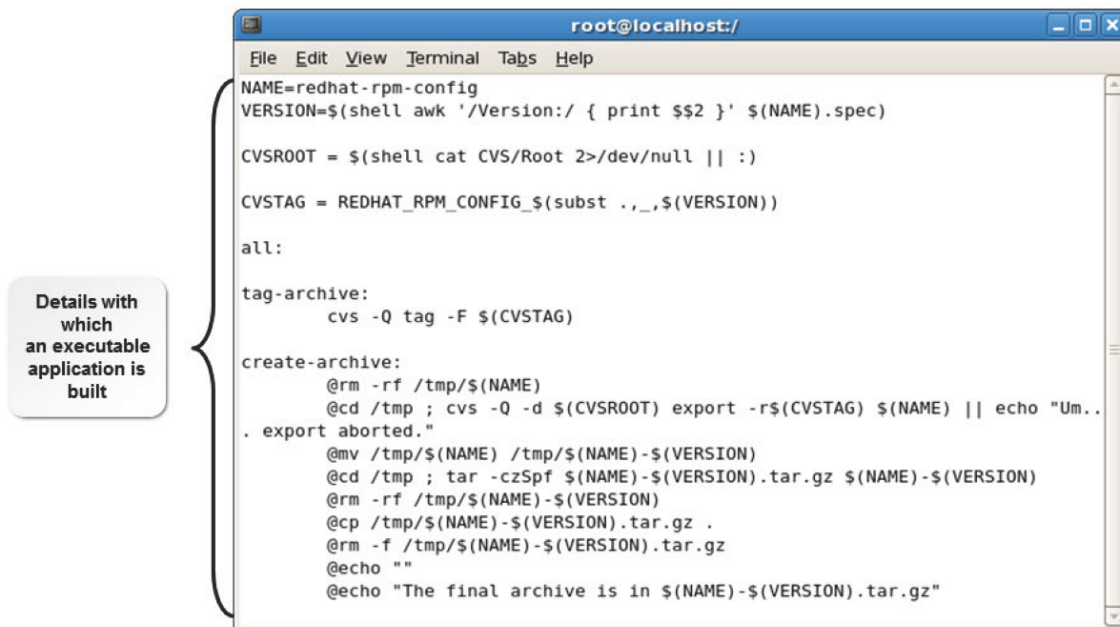


Figure 7-10: makefile is used to configure, compile, and install the application or driver.

makefile Commands

makefile allows you to build an application or a driver from its source. You need to issue certain commands in a sequence.

Command	Used To
./configure	Gather system information to compile an application.
make	Compile an application.
make install	Install the newly compiled program.
make uninstall	Uninstall a program.
make clean	Clean up after successfully compiling an application.
make test	Install a perl module. It is an optional command in the package management function.



Note: In the previous commands, make clean and make test are optional, while the first four commands are necessary in any makefile operation.

Autoconf

You can use autoconf to create shell scripts that automatically configure packages. The shell scripts created by autoconf can run independently.

Tarballs

Some applications, even though they are really just standard tar files, are referred to as tarballs.

Tarballs come in several different formats, as described in the following table.

Format	Description
.tar	The standard tar file without extra compression.
.tar.gz or .tgz	The standard gzip-compressed tar file.
.tar.bz2	A tar file that is further compressed using the bzip2 utility.

Format	Description
bin.tar, .bin.tar.gz, or .bin.tgz	A tar file containing binary files rather than source files.

Compiling from makefile

Most source files are available in the tarball format. To compile an application, you need to be at the command line or work through a GUI. First, change to the directory that is made by the package.

Here, you should find an INSTALL file. Read the contents of the INSTALL file. Most tarballs include one or more of the following files: INSTALL, COPYING, README, or CHANGES. The INSTALL file usually includes a generic process for installing tarballs. If a program needs to be compiled in a certain way, you can find the necessary information in either the INSTALL or README file. This is how it works in theory, but it does not always work, usually due to dependencies on other programs.

makefile and configure Script

Generally, application vendors provide either a makefile or a configure script to build an executable program. Both files come with default settings, and you can customize these files to suit your individual preferences. For example, you can specify the location where you need the files to be installed.

When you build a program using a makefile, you need to make the necessary changes to the file, and you also need to update the necessary header files; otherwise, the installation will be incomplete.

When a vendor provides you with a configure script, you can make the desired changes to the script and then run the script. This will generate the necessary files along with the makefile. Because it is not necessary to make any change to the makefile, you can proceed with the installation.

How to Install Software Using Source Files

Follow these general procedures to install software using source files

Compile Applications or Drivers

This is the most common way to compile an application from source:

1. Download the source code from the vendor.
2. Untar and ungzipped the files into a directory.
3. In the directory containing the files, issue the `./configure` command.
4. Next, issue the `make` command to compile the application.
5. To install the application, use the `make install` command.
6. Optionally, to clean up the temporary files used during the compile process, use the `make clean` command.

Shared Libraries

A **shared library** is a file that contains routines, which are used by various applications. Shared libraries are loaded into memory by the operating system when required. They are then shared with other applications. Shared libraries are loaded when an executable file that links to them is loaded.

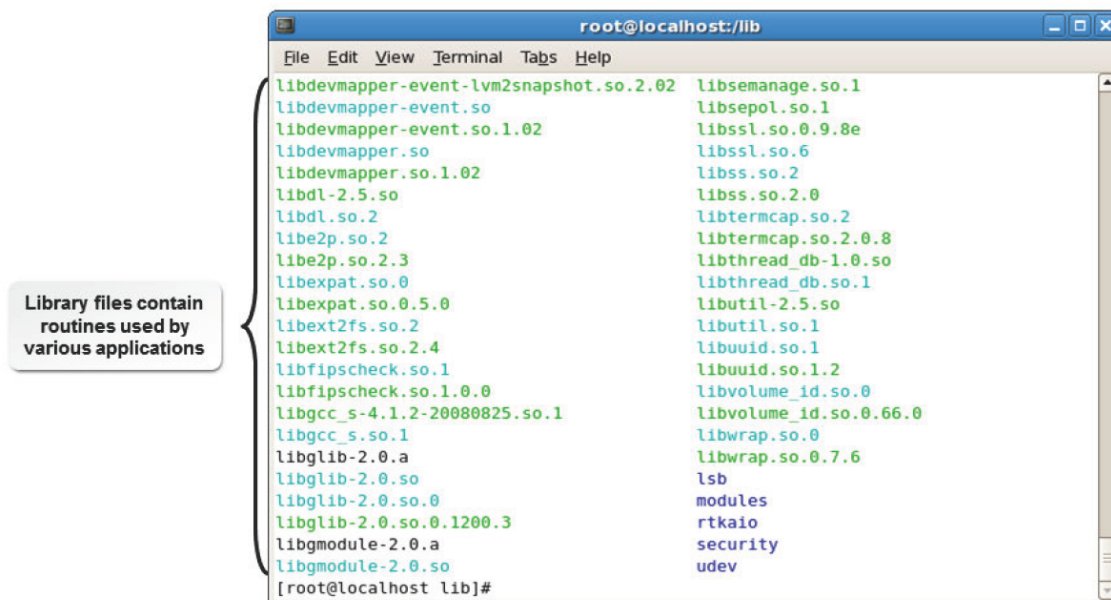


Figure 7-11: List of shared libraries.

Executable Programs

An executable program is a file in a format that a computer can directly execute. Executable files cannot be read by human beings because they are not plain text files and are compiled. An executable file is used to perform various functions or operations on a computer. Binary machine instructions that the computer knows how to execute can be included in an executable file. It can also contain a shell script. Executable files in Linux can have any name.

The ldconfig Command

When you add a new library file, the file details are passed on to the **/etc/ld.so.conf** file, which contains the details of the default system libraries. You need to run the ldconfig command to update these changes from the **/etc/ld.so.conf** file and load the shared libraries from the locations specified to the **/etc/ld.so.cache** file. The **/usr/lib** and **/lib** directories are the default system library file locations where the system libraries are kept. Some of the common ldconfig command options are given in the following table.

<i>If You Need To</i>	<i>Use This ldconfig Command Option</i>
Specify the configuration file where the library paths are stored.	-f {configuration file name}
Specify the cache file where the library file updates will be stored.	-C {cache file name}
Display the details of the library file.	-v
Update the library file information in the specified location on the command line instead of the default location.	-n //location]



Note: The **ld.so.conf** file contains the location details of the shared libraries.

How to Manage Shared Libraries

Follow these general procedures to manage shared libraries.

Install Library Files

To install library files:

1. Log in as **root**.

2. To identify the library files required for a package to be installed, enter `rpm -qpR {package file name}`.
3. If necessary, to update the locate command database, enter `updatedb`.
4. To verify that the library files are present on the system, enter `locate {library file name}`.
5. If the library files are not present, install the necessary packages by entering `rpm -ivh {package file name}`.

Determine the Location of Shared Libraries

To determine the location of shared libraries:

1. Log in as **root**.
2. To display the list of shared libraries required for an application, enter `ldd l{location of the executable file}`.

Set a Custom Path for Additional Library Files

To set a custom path for additional library files:

1. Log in as **root**.
2. To open the **profile** file, on the terminal, enter `vi /etc/profile`.
3. Specify the location of the other library files and export the library path variable.
 - a. To define the `LD_LIBRARY_PATH` variable, in a new line, enter `LD_LIBRARY_PATH=/usr/lib:/lib:/Location of the other library files`.
4. Save and close the file.
5. Log out and log in as **root** to update the changes.



Caution: While defining the `LD_LIBRARY_PATH` variable, you must specify the default path of the shared library along with the location of the newly added library files.

- b. To convert the `LD_LIBRARY_PATH` variable to an environment variable, enter `export LD_LIBRARY_PATH`.

ACTIVITY 7-1

Managing Packages Review

Scenario

Answer the following review questions.

1. What packages might you need to install in your organization? Which package management tools will you use most often?
2. Why do you think it is important to create your own repositories?

Summary

In this lesson, you managed packages using package managers and explored the various repositories from where you can download the packages. This will enable you to easily install software packages on Linux systems.

8 Managing Kernel Services

Lesson Time: 1 hour, 45 minutes

Lesson Introduction

In the last lesson, you managed packages to better modify the software running on your Linux® system, which might be different depending on the Linux distribution you are using.

But there are some aspects of Linux that are consistent, and those are typically handled by the kernel. The kernel, being the core of the Linux operating system, handles various crucial functions such as system initialization, process scheduling, and memory and hardware management. In this lesson, you will explore the role of kernel services and kernel service configuration.

As a Linux system administrator, you may need to configure, modify, and customize the kernel to meet user requirements. Even a minor misconfiguration may cause kernel malfunction, rendering the system ineffective. Therefore, an in-depth knowledge of kernel services is required to manage the kernel efficiently.

Lesson Objectives

In this lesson, you will manage kernel services. You will:

- Identify the role and functions of the Linux kernel.
- Customize kernel modules.
- Create an initrd image.
- Manage device drivers.
- Monitor processes and resources.

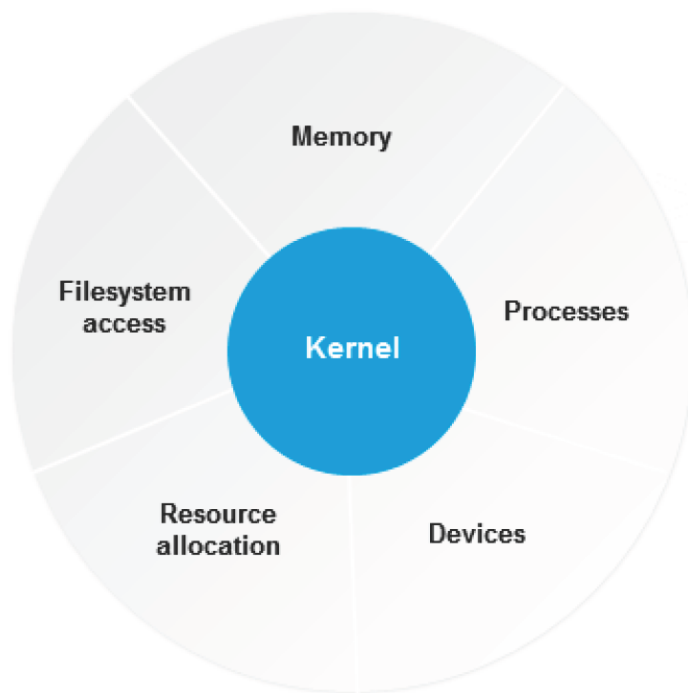
TOPIC A Explore the Linux Kernel

The first component that initializes in the Linux boot process is the kernel. It provides all the essential services that are required for running the computer and controls the rest of the processes that operate on the computer. In this topic, you will identify the role of the Linux kernel and its functions.

If a system crashes or stops performing, it actually means that the kernel or an operation critical to the working of the kernel has failed. As a Linux administrator, you need to understand the functionality of the kernel to be able to troubleshoot and provide solutions.

The Kernel

The **kernel** is the core of an operating system. All other components rely on it. It is loaded first and remains in the main memory. It contains system-level commands and other functions that are hidden from users. The kernel manages filesystem access, memory, processes, devices, and resource allocation on a system. The kernel also controls all the hardware devices plugged into the system.



Functions of the kernel

Figure 8-1: The kernel's role in an operating system.

The Linux Kernel

The [Linux kernel](#), which is the core constituent of the Linux operating system, manages all other resources on the system. It performs functions such as sharing resources and allocating memory, input and output operations, security settings, and user access. It controls the interaction between software applications and underlying system resources. The kernel initializes itself during the boot process and then starts running the other processes. By default, the kernel loads with a minimal set of functions required to run a system. The kernel's functionality can be expanded by installing kernel modules. The kernel is required to synchronize the operations of multiple processes and govern resources.

Kernel Versions and Modules

Linux kernel versions refer to the different editions of the Linux kernel. Kernel versions are identified by their kernel number, which consists of four parts. The format of the version number is *major_version_number.major_revision_number.minor_revision_number.fix_number*.

The version number can be viewed using the `uname -r` command. Common kernel modules include `input`, `ext4`, `CD-ROM`, `lp`, `udf`, and `jbd`.

Kernel Layers

The kernel performs various functions to control and manage the operations of a system. It is composed of various layers.

<i>Kernel Layer</i>	<i>Function</i>
System Call Interface (SCI)	Handles function calls sent from user applications to the kernel. A function call is basically a service request sent to the operating system's kernel for invoking system-level functions such as requests for processing time and memory allocation. This layer also enables the kernel to schedule and process function calls and manage multiple function calls simultaneously.

<i>Kernel Layer</i>	<i>Function</i>
<i>Process management</i>	Handles different processes by allocating separate execution space on the processor and ensuring that the running of one process does not interfere with other processes. The kernel implements sharing of the processor time for executing multiple processes through process scheduling.
<i>Memory management</i>	Manages the computer's memory, which is one of the complex tasks performed by the kernel. Like processor sharing, the system's memory also needs to be shared among different application services and resources. The kernel maps or allocates the available memory to applications or programs on request and frees the memory automatically when the execution of the programs is complete, so that it can be allocated to other programs.
<i>Filesystem management</i>	Manages the filesystem, which involves storing, organizing, and tracking files and data on a computer. The kernel also supports a virtual filesystem that provides an abstract view of the underlying data that is organized under complex structures, so that it appears to be a single structure.
<i>Device management</i>	Manages devices by controlling device access and interfacing between user applications and hardware devices of the computer. When the user application sends a system call, the kernel reads the request and passes it on to the drivers that manage the activities of that particular device. For this purpose, the kernel maintains a list of devices in the <i>ldev</i> directory.

Types of Kernels Available in Linux

Kernels can be classified as monolithic or modular.

<i>Kernel Type</i>	<i>Description</i>
<i>Monolithic</i>	In a monolithic kernel, all modules, such as device drivers or filesystems, are built-in. A monolithic kernel can interact faster with devices. But the major disadvantage is its huge size, which leads to higher usage of RAM.
<i>Modular</i>	In a modular kernel, only a minimal set of essential modules are built-in. The rest of the modules can be installed and the kernel can be rebuilt whenever necessary. A modular kernel is also known as a micro kernel or a dynamic kernel. Modular kernels are flexible and save memory usage because the kernel modules, which are loaded as required, are removed from the memory when the related devices are unmounted.

TOPIC B Customize Kernel Modules

In the previous topic, you familiarized yourself with the basic concepts of the Linux kernel. Kernel modules are functions that extend the capability of the kernel to support additional functionalities.

In this topic, you will customize kernel modules.

The Linux kernel, by default, loads with a minimum set of kernel modules. When you want the kernel to support some additional functionality, you have to install or load the necessary modules manually. Customizing the modules to suit user requirements will enable you to manage the kernel efficiently.

Kernel Modules

A **kernel module** is a system-level function that extends the functionality of the kernel. It can be dynamically loaded into the kernel or unloaded from the kernel when required. It enables the kernel to update or recompile itself without requiring the system to reboot. The kernel module file consists of a **.ko** extension. Modules built for a specific kernel version may not be compatible with another version of the kernel.

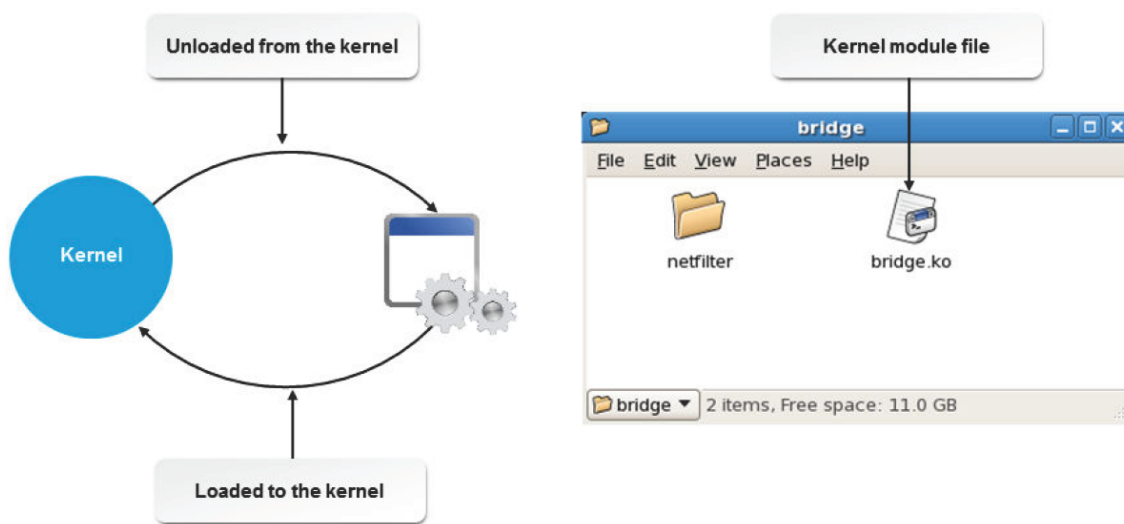


Figure 8-2: *bridge.ko* is the kernel module for network support.

Advantages of Kernel Modules

The advantages of kernel modules are:

- Kernel modules reduce the burden on the kernel. If there are no kernel modules, their functionalities have to be added directly to the kernel image, which can make the kernels larger.
- Kernel modules avoid rebuilding and rebooting of the system when a new functionality is required.
- Dynamic loading of kernel modules facilitates lower memory consumption.

Directories Containing Kernel Modules

The **/lib/modules** directory contains the modules of different kernel versions that are installed. It contains a directory named after the kernel's version number. A list of currently loaded modules is found in the **/proc/modules** file. Modules are stored across various directories based on the categories they belong to. The following table lists the directories containing modules.

Directory	Contains Modules For
pcmcia	PCMCIA (PC Card).
net	Network-related products such as firewalls and protocols.
drivers	Various types of hardware.
fs	Various types of filesystems.
arch	Architecture specific support.

Kernel Module Managing Utilities

A kernel module managing utility enables you to view, load, unload, or modify kernel modules.

Kernel Module Utility	Enables You To
lsmod	Display the currently loaded kernel modules, their sizes, usage details, and their dependent modules.
modinfo	Display information about a particular kernel module such as the file name of the module, license, description, author's name, module version number, dependent modules, and other parameters or attributes. The syntax of this command is <code>modinfo [options]</code> .

Kernel Module Utility	Enables You To
insmod	Install a module into the currently running kernel. This utility inserts only the specified module and does not insert any dependent module. The syntax of this command is <code>insmod {file name} [options]</code> .
modprobe	<p>Add or remove modules from a kernel. This utility is capable of loading all the dependent modules before inserting a specified module.</p> <ul style="list-style-type: none"> The syntax for adding a module is: <code>modprobe {module name}</code>. The syntax for removing a module is: <code>modprobe -r {module name}</code>.

Command Options for modinfo

The command options for the modinfo command are listed in the table.

Command Option	Enables You To
-V	Display the version number of the modinfo utility.
-n	Display the file name of the module.
-a	Display the author of the module.
-d	Display the description about the module.
-p	Display the parameters supported by the module.

Command Options for insmod

There are several command options for the insmod command.

Command Option	Enables You To
-e {persistent name}	Add persistent parameters to the module.
-f	Force the loading of a module even when there is a difference between the module's kernel version and the current kernel version.
-L	Prevent simultaneous loading of the same module.
-o {module name}	Specify a module name while installing the module.

Command Options for modprobe

There are several command options for the modprobe command.

Command Option	Enables You To
-a	Add all the modules specified in the command line.
-r	Remove all the modules specified in the command line.
-v	Display the verbose of all the commands when they are executed.
-l	List all the modules that match the given wildcard information.
-t {directory name}	List all the modules present in a specified directory.

The modprobe.conf File and /etc/modprobe.d/

The **modprobe.conf** file is a configuration file, which contains settings that apply persistently to all the modules loaded on the system. It is used to configure modules and their dependencies and also specify module aliases.



Note: The `/etc/modules.conf` file was used to manage kernel modules in older versions of Linux.

The **modprobe.conf** file, which is located in the `/etc/modprobe.d` directory, has a number of options for configuring kernel modules.

Option	Used To
<code>alias {wildcard} {module name}</code>	Specify an alternate name for a module with a long name.
<code>include {file name}</code>	Add configuration files to a module.
<code>options {module name} {option}</code>	Specify the options to be added to each module before insertion into the kernel.
<code>install {module name} {command}</code>	Run the command specified without inserting the module into the kernel.

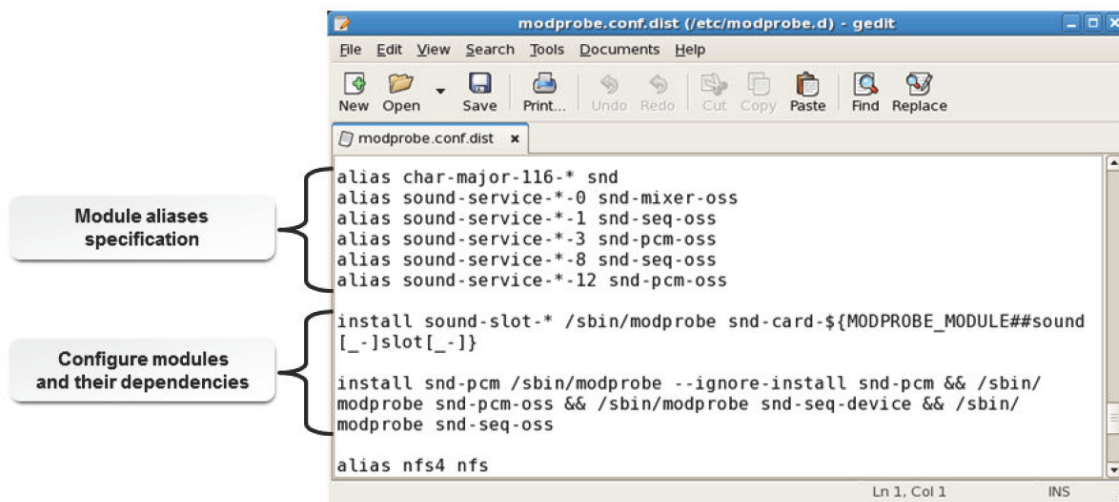


Figure 8-3: The `modprobe.conf` file is used to configure the kernel modules.

Kernel Options

Kernel options allow you to pass parameters to the kernel at the time of booting a system. These options are used to customize the kernel to suit the individual system or to troubleshoot booting issues. Some of the common kernel options are listed in the following table.

Kernel Option	Systemd Target	If You Need To
1 or s	rescue.target	Switch to runlevel 1.
2	runlevel2.target	Switch to runlevel 2.
3	multi-user.target	Switch to runlevel 3.
4	runlevel4.target	Switch to runlevel 4.
5	graphical.target	Switch to runlevel 5.
rw	n/a	Mount the root device in read-write mode while booting the system.
ro	n/a	Mount the root device in read-only mode while booting the system.
debug	n/a	Enable kernel debugging and log important system events.

Types of Kernel Configuration

Linux kernels can be configured in two different ways.

Kernel Configuration Type	Description
---------------------------	-------------

Kernel Configuration Type	Description
Persistent	Refers to the configuration of kernel settings that do not change even after the system is rebooted. The changes made to the kernel are permanent. The kernel configuration with the sysctl.conf file is persistent and does not get effaced when the kernel is initialized again.
Transactional	Refers to updating the kernel settings for a required service. These settings are not permanent and are reverted when the system is rebooted. The settings hold good only for a particular transaction of the kernel. The kernel configuration with the /proc file is transactional and the changes are reflected immediately. This type of configuration can be used for network services modification and features related to memory subsystems.

The /proc/version File

The **/proc/version** file specifies the version of the Linux kernel, the **GNU Compiler Collection (GCC)**, and the Linux distribution installed on the system.



Figure 8-4: Details displayed by the /proc/version file.



Note: GCC originally stood for GNU C Compiler because it was produced by the GNU project.

The /proc Directory

The **/proc** is a directory which contains a Linux virtual filesystem, which provides elaborate information about the kernel's running process. Some of the files in the **/proc** directory are listed in the following table.

File	Description
/proc/cmdline	Contains the command line passed to the kernel by the boot loader at boot time.
/proc/cpuinfo	Stores the CPU information and system architecture dependent items.
/proc/devices	Contains the list of device drivers configured into the currently running kernel.
/proc/filesystems	Contains the list of filesystems that are configured into the kernel.
/proc/partitions	Contains partition information including the major and minor number of each partition, partition name, and number of blocks.
/proc/ip_forward	Permits interfaces on the system to forward packets to one other.

<i>File</i>	<i>Description</i>
/proc/meminfo	Contains memory information such as the used and unused memory on the system and the shared memory and buffers used by the kernel.

The /sys Directory

The /sys is a directory which contains a Linux virtual filesystem, which provides elaborate information about the various kernel subsystems, hardware, and device drivers.

The sysctl Command

The **sysctl** command is used to view or set the kernel parameters at runtime. It has various options.

Persistent kernel settings are added in the **sysctl.conf** file.

<i>Command Option</i>	<i>Used To</i>
-w {variable}={value}	Set a parameter value or to change the sysctl setting.
{variable}={value}	Set a key parameter value.
-n	Disable the printing of the key name while displaying the kernel parameters.
-e	Ignore errors about unknown keys.
-a	Display all the parameter values that are currently available.
-A	Display all the parameter values that are currently available in a tabular format.

```

root@localhost:/
File Edit View Terminal Tabs Help
[root@localhost ~]# sysctl -a | more
sunrpc.max_resvport = 1023
sunrpc.min_resvport = 665
sunrpc.tcp_slot_table_entries = 16
sunrpc.udp_slot_table_entries = 16
sunrpc.nlm_debug = 0
sunrpc.nfsd_debug = 0
sunrpc.nfs_debug = 0
sunrpc.rpc_debug = 0
crypto.fips_enabled = 0
dev.cdrom.check_media = 0
dev.cdrom.lock = 1
dev.cdrom.debug = 0
dev.cdrom.autoeject = 0
dev.cdrom.autoclose = 1
dev.cdrom.info = CD-ROM information, Id: cdrom.c 3.20 2003/12/17
dev.cdrom.info =
dev.cdrom.info = drive name:          hda
dev.cdrom.info = drive speed:         1
dev.cdrom.info = drive # of slots:    1
dev.cdrom.info = Can close tray:      1
dev.cdrom.info = Can open tray:      1
dev.cdrom.info = Can lock tray:      1
dev.cdrom.info = Can change speed:   1

```

Figure 8-5: The different options of the sysctl command.

Syntax

The syntax of the sysctl command is **sysctl [options] {kernel parameter}={value}**.

How to Customize Kernel Modules

Follow these general procedures to customize kernel modules.

Load Modules Using the insmod Command

To load modules using the insmod command

1. Log in to the CLI as **root**.
2. To insert a specified module into the kernel, enter `insmod {module name}`.
3. To view detailed information about the inserted module, enter `modinfo [options] {module name}`.

:

View Information About the Running Kernel Modules

To view information about the running kernel modules:

1. Log in to the CLI as **root**.
2. To view the status of all the loaded modules, enter `lsmod`.

Add or Remove Modules Using the modprobe Utility

To load modules using the modprobe utility:

1. Log in to the CLI as **root**.
2. To add the specified module and all its dependent modules into the kernel, enter `modprobe {module name}`.
3. To view the status of the loaded modules, enter `lsmod`.
4. If necessary, to remove a loaded module, enter `modprobe -r {module name}`.

Configure Modules Using the modprobe.conf File

To configure modules using the **modprobe.conf** file:

1. Log in to the CLI as **root**.
2. To remove a loaded module, enter `cd /etc`.
3. To open the **modprobe.conf** file, enter `vi modprobe.conf`.
 - Specify the parameter to pass through when the module is loaded.
 - Set the aliases for a module name.
4. Save and close the file.

Manage the Kernel Using the /etc/sysctl.conf File

To manage the kernel using the **/etc/sysctl.conf** file:

1. Log in to the CLI as **root**.
2. Open the **/etc/sysctl.conf** file.
3. Make the necessary modifications to the kernel settings.
4. Save and close the file.
5. Reboot the system.

Configure the Kernel Using the /proc Directory

To configure the kernel using the **/proc** directory:

1. Log in to the CLI as **root**.
2. To configure the kernel parameters, enter `echo {value} > /proc/{file location whose value in the kernel needs to be changed}`.
3. Save and close the file.

Configure the Kernel Using the sysctl Command

To configure the kernel using the **sysctl** command:

1. Log in to the CLI as **root**.
2. To configure the kernel parameters, enter `sysctl [options] {kernel parameter}={value}`.

TOPIC C Create an initrd Image

In the last topic, you configured and customized kernel modules. The **initrd** image, or the initial ramdisk image, consists of all the kernel modules that were loaded during the boot process.

Additional modules that are installed also need to be added to the **initrd** image to load them automatically at boot time. In this topic, you will create the **initrd** image to update the kernel.

The existing kernel on your system may have all the necessary modules, but at a later stage, you may need to update the modules when a new set of devices have to be supported. Knowing how to update the existing modules by creating the **initrd** image will enable you to provide support for new devices.

initrd

initrd refers to the initial ramdisk that is temporarily mounted as the root filesystem for loading startup programs and modules. The ramdisk loads along with the kernel, which controls its functionality. **initrd** enables the system to be started in two phases. In the first phase, the system is booted with the minimal set of modules required to load the main or the permanent root filesystem.

In the second phase, when the main root filesystem is mounted, the previously mounted **initrd** filesystem is removed and the ramdisk is released for installing additional modules on demand.

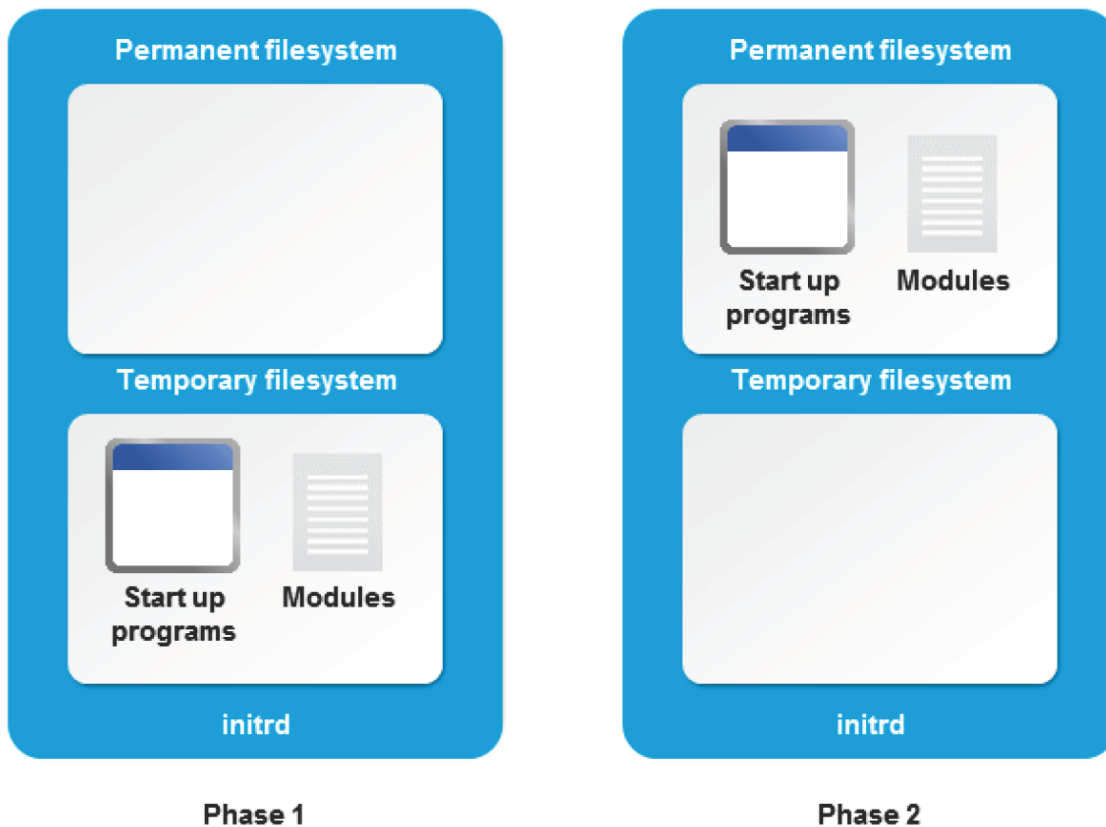


Figure 8-6: The `initrd` starts the system in two phases.

The `initrd` Image

The **`initrd image`** is an archived file containing all the essential files that are required for booting the operating system. It can be built or customized to include additional modules, remove unnecessary modules, or update existing modules.

The `mkinitrd` Command

The **`mkinitrd`** command is used to create the initial ramdisk image for preloading the kernel modules.

```
root@localhost:/  
File Edit View Terminal Tabs Help  
[root@localhost /]# mkinitrd /boot/new-initrd-image.img `uname -r`  
[root@localhost /]#
```

Figure 8-7: New image created using mkinitrd command.

Various options of the mkinitrd command are given in the following table.

Command Option	Used To
--preload={module name}	Load a module in the initrd image before the loading of SCSI modules.
--with={module name}	Load a module in the initrd image after the loading of SCSI modules.
--fstab={fstab}	Automatically determine the type of filesystem that the root device is found on.
--builtin={module name}	Specify that the module is already built into the currently loaded kernel, so that the mkinitrd command will omit it while creating the initrd image.
--omit-lvm-modules	Avoid loading the LVM modules while creating the initrd image.
--omit-raid-modules	Avoid loading the RAID modules while creating the initrd image.
--omit-scsi-modules	Avoid loading the SCSI modules while creating the initrd image.
-f	Overwrite an existing initrd image file.

How to Create an initrd Image

Follow these general procedures to create an initrd image.

Create an initrd Image with Updated Information

To create an initrd image with updated information:

1. Log in to the CLI as **root**.
2. To create an initrd image, enter `mkinitrd [options] /boot/initrd-{kernel version number}.img {kernel version number}`.
3. Update the `/boot/grub2/grub.cfg` file with the updated initrd information.

TOPIC D Manage Device Drivers and Hardware Devices

Throughout this lesson, you have been performing various kernel service management tasks. Device management is another important service provided by the kernel. In this topic, you will manage kernel-based device drivers. Drivers are associated directly with the hardware devices that are installed on your computer. In this topic, you will also monitor various hardware devices.

A system administrator has to read and write details to the driver files frequently when additional hardware is required or existing hardware is upgraded. Knowing how to access drives through the **/dev** directory will enable you to handle this task effectively. A system administrator also needs to track all the devices that are connected to a computer and monitor them continuously. Gaining knowledge about utilities that are used to track these hardware devices is essential for proper management of a Linux system.

udev

The device manager **udev** manages the automatic detection and configuration of hardware devices. udev is an integral part of the kernel, which is initialized during boot time. The udev utility handles module loading for both coldplug and hotplug devices. It loads the modules for coldplug devices, such as a monitor or a sound card, when the system is booted. The modules for hotplug devices, such as a USB drive or a camcorder, are loaded by udev dynamically during system run time.

The /dev Directory

The `/dev` directory includes hardware and software device drivers.

Coldplug vs. Hotplug

Hotplug is the ability of a system to add or remove hardware without rebooting the system, while coldplug is the inability to do so. Hotplug devices are detected by the system as they are plugged in, whereas coldplug devices, such as conventional hard disks, are not sensed when connected to a running system; they need a complete reboot of the system to function. Some coldplug devices, such as hard disk, PCI, and RAM, can be connected only when the system is not running.

The `/sys` Directory

In Kernel 2.6 or above, the `/sys` directory contains information about hotplug hardware devices and displays them in a hierarchical format. It is similar to the `/proc` filesystem because it contains information related to files loaded in the kernel memory. The `sys` directory is mounted by default and can be listed using the mount command. The `sys` directory is mounted as a `sysfs` filesystem, a virtual filesystem.

dbus

The dbus, short for Desktop Bus, is a Inter-Process Communication (IPC) system that was designed to standardize services provided by Linux desktop environments. Applications register with the dbus daemon to receive and post notifications pertinent to the hardware abstraction layer. dbus services are configured by default to for auto-mount CD-ROMs and USB hardware on modern Linux distributions.

Device Drivers

A **device driver** is a software program that enables a computer's operating system to identify the characteristics and functions of a hardware device, communicate with it, and control its operations.

It acts as an interface between the operating system and hardware devices such as hard drives, CD/DVD drives, printers, scanners, monitors, and keyboards. Device drivers can be part of the operating system or installed on demand.

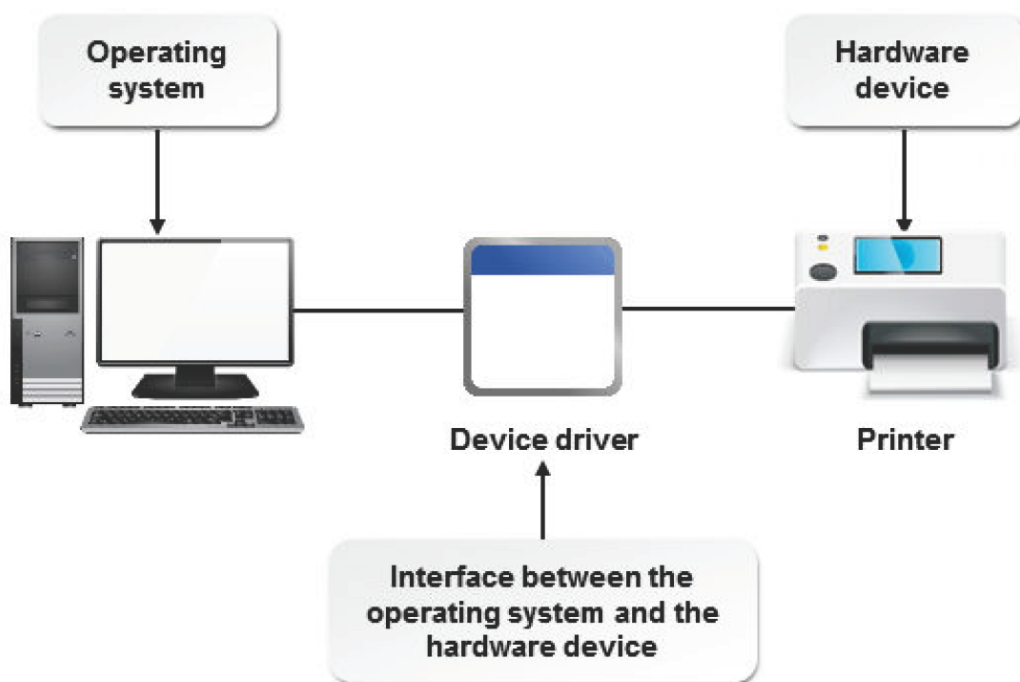


Figure 8-8: A printer driver is used by the operating system to communicate with your printer to print files or documents.

Device Tree

A **device tree** is a structure that lists all hardware devices installed on a system and assigns device nodes to them. It is auto generated by the computer's RAM when the computer is started, when a new device is installed, or when a device or system configuration is modified.

Device Nodes

A **device node** is an access point for device drivers; it is used while mapping service requests with device access. It represents a particular hardware resource in a device tree. It is also known as a device file or a device special file. A node contains vital information such as the device type, the major number, and the minor number. A **minor number** identifies a particular device and the **major number** identifies the device driver that controls this particular device. Device nodes are located in the /dev directory.

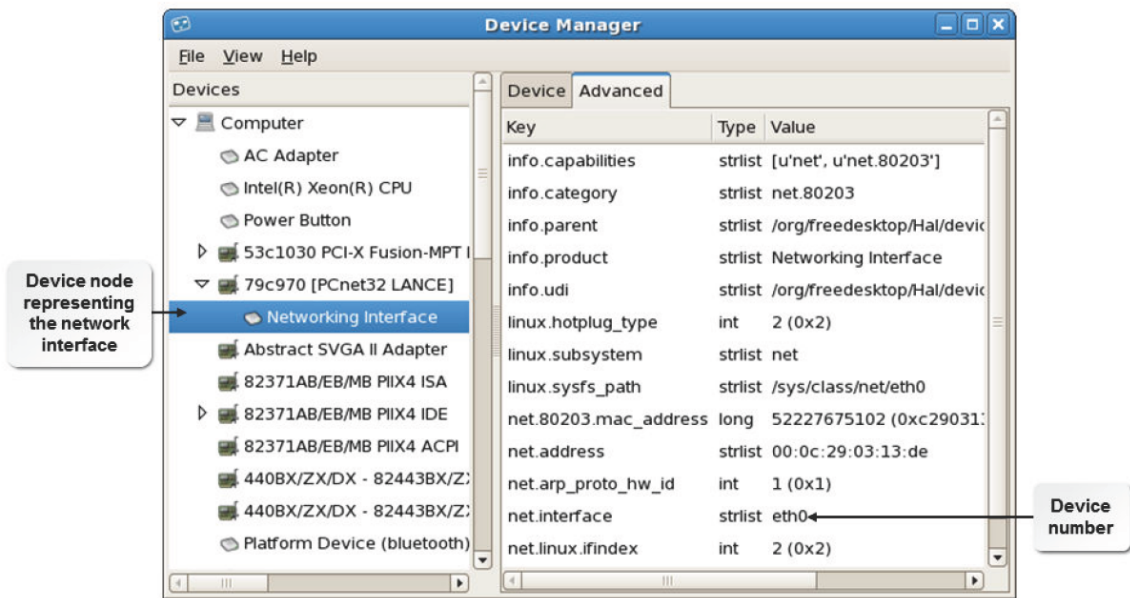


Figure 8-9: A device node representing the disk volume.

Types of Hardware Devices

Hardware devices can be divided into two types based on their usage or function.

Device Type	Description
Block devices	<p>These are typically used for data storage. They buffer all the service requests received to choose the order in which the requests have to be responded.</p> <p>Block devices accept input and provide output in the form of blocks, which are of larger byte sizes. Examples are:</p> <ul style="list-style-type: none">Hard disks: /dev/hda and /dev/sdaSoftware RAID: /dev/md[0-5]
Character devices	<p>These are typically used for data streaming and do not use buffering to handle service requests. They accept input and provide output in smaller byte sizes. Examples are:</p> <ul style="list-style-type: none">Software devices: /dev/null and /dev/zeroVirtual consoles: /dev/tty[0-6]

Special Devices

Linux provides a few special character devices that are used occasionally.

Special Device	Description
----------------	-------------

Special Device	Description
/dev/zero	Provides unlimited null characters (0 bytes) for writing into any program or file. It is used for generating an empty file of certain size.
/dev/null	Does not provide any data to a program or file. It discards all data written to it. It is used as an output file when the output is not required by the user.
/dev/random	Functions as a random number generator. It gathers random input from device drivers and other sources on the system and saves it as bits in an entropy pool. It provides randomly generated output as bytes to applications within the number of bits in the pool. When the pool is exhausted, the /dev/random device will block the reading application until more random input is collected.
/dev/urandom	Functions similarly to /dev/random, except that it does not block the reading application if the entropy pool is exhausted. It uses a software algorithm to generate alternate random input that may be less secure than the input generated by the system.

The mknod Command

The **mknod** command allows you to create device files that are not present, using the major and minor node numbers of a device.

Syntax

The syntax of the mknod command is `mknod [OPTION] {NAME} {TYPE} [MAJOR MINOR]`.

Hardware Communication Channels

The kernel and hardware devices communicate using major channels such as Interrupt Requests, Input/Output (I/O) addresses, and Direct Memory Address (DMA).

Hardware Communication Channel	Description
Interrupt ReQuests (IRQ)	An interrupt request is a signal sent by a hardware device to the kernel to request processing time in order to perform an operation. This enables the kernel to prioritize system events and allocate the CPU's processing time for devices.
Input/Output (I/O) Addresses	Every hardware device communicates with the operating system through a unique I/O address. The kernel uses this address to identify the requests sent to or from the device. It is also used to map the device with user applications requesting the device services.
Direct Memory Address (DMA)	A method by which hardware devices directly communicate with the memory to obtain memory allocation without going through the processor.

The HAL

The **Hardware Abstraction Layer (HAL)** is a logical interface that enables software applications to interact with hardware devices at an abstract level through system calls. This layer converts generic system calls sent by software applications to detailed device-specific instructions. It enables an operating system to adapt to different kinds of hardware platforms without requiring any modification in the kernel.

Figure 8-10: HAL serves as an interface between software and hardware.

HAL Utilities

The HAL utilities enable you to view or monitor the hardware device connected to the computer.

<i>HAL Utility</i>	<i>Used To</i>
lspci	Display information about all the PCI buses and all the peripheral components connected to a computer.
lsusb	Display all the USB components connected to a computer.

The D-Bus System Bus

D-Bus is the system bus that provides the main communication between applications (IPC, Inter-Process Communication). This contains a daemon that can invoke specific services on the system based on the needs of the requesting application. This daemon can send system wide alerts such as "new hardware detected" and "print queue modified."

How to Manage Device Drivers

Follow these general procedures to manage device drivers.

Access Drivers Through /dev

To access device driver files through **/dev**:

1. Log in to the CLI as **root**.
2. To check which terminal is used and the users who are logged in, enter `who`.
3. To view the device driver file, enter `cat /dev/{device node}`.
4. To send messages using the specific device node, enter `echo {messages} > /dev/{device node}`.

Add Files Under /dev

To add files under **/dev**:

1. Log in to the CLI as **root**.
2. Add files under the **/dev** directory.
 - Create files in the **/etc/udev/rules.d** directory.
 - a. To open the **rules.d** directory, enter `cd /etc/udev/rules.d`.
 - b. To view the timestamps of the device file, enter `touch {file name}`.
 - c. Switch to insert mode.
 - d. To open the device file, enter `vi {file name}`.
 - e. To add details to the file, type the text as indicated: `KERNEL=="{device}"`, `NAME="{device node}"`.
 - f. Save and close the file.
 - Create files using the `mknod` command.
 - a. To create the device node, enter `mknod /dev/{device node}{device type} {major number}{minor number}`.

How to Monitor Hardware Devices

Follow these general steps to monitor hardware devices.

Monitor Hardware Devices

To monitor the hardware devices currently connected to a system:

1. Log in as **root**.
2. Monitor hardware devices.
 - To list the status of all PCI devices, enter `lspci`.
 - To list the status all USB devices, enter `lsusb`.

TOPIC E Monitor Processes and Resources

In the last topic, you monitored the hardware devices on your computer. Along with hardware devices, software applications and programs work in conjunction to make the entire system work.

Software programs are handled by the processor. In this topic, you will monitor processes to view how system resources are utilized and how the processor manages them.

As a system administrator, you may need to handle a number of running processes simultaneously.

Based on the need, one program may require a higher priority than another. While the execution of one process is in progress, you may decide to pause or stop the process to start another important process. Performing process monitoring will help you manage multiple programs and their resource allocation.

Load Average

Load average is the average number of processes waiting to run on a system for the last 1 minute, 5 minutes, and 15 minutes. Ideally, the number should be less than one. This information can be used to check whether the system is busy. The load average information is specific to the operating system and the hardware.

Kernel State Monitoring Utilities

Kernel state monitoring utilities are used to gather information about the operating system and its running events and processes. The kernel state monitoring utilities are listed in the table.

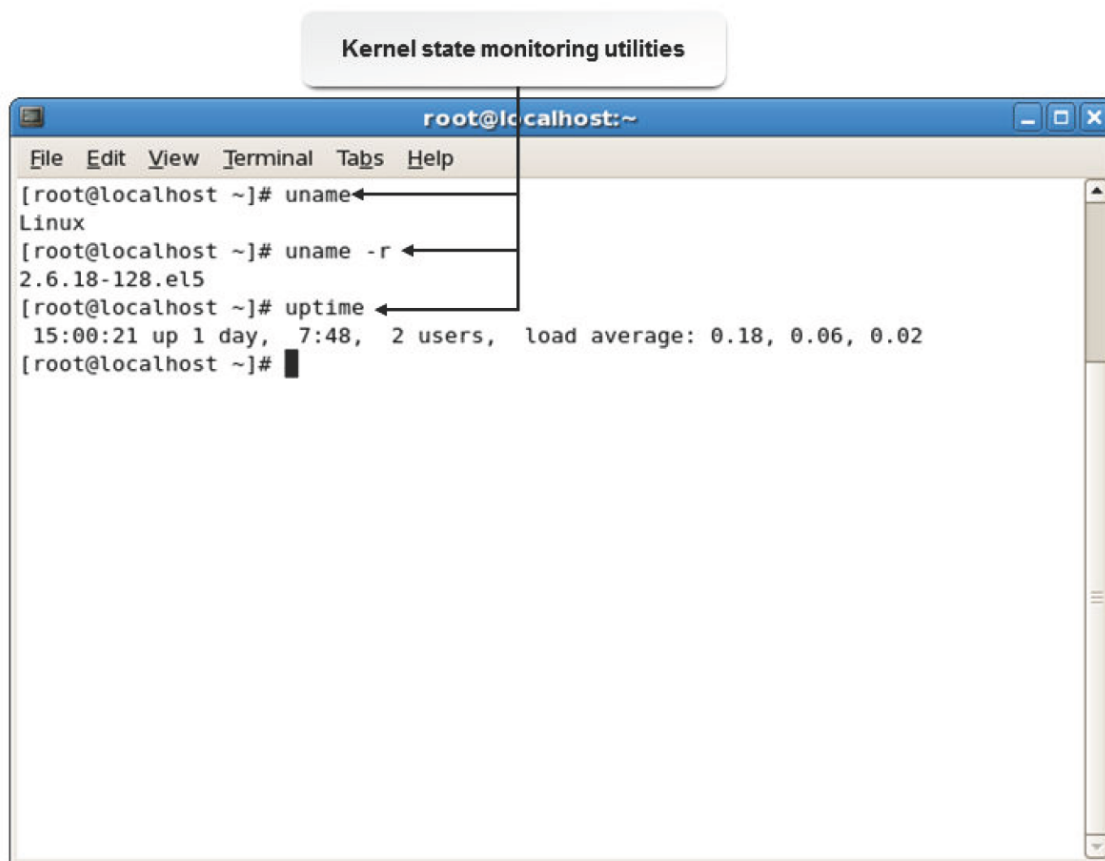


Figure 8-11: Output of various kernel state monitoring tools.

Kernel State Monitoring Utility	Enables You To
uname	Display the name of the operating system, its version, license, processor, and hardware details.
uptime	Display the duration for which the system has been running, the number of users logged on currently, and the load average of the system.
tload	Provide a graphical representation of the system and the load average for the past 1 minute, 5 minutes, and 15 minutes.

System Load

System load is a measurement of the amount of work done by a computer over a given period of time. It is represented in the form of three numbers. The first number indicates the system load during the last 1 minute, the second number indicates the system load during the last 5 minutes, and the last number indicates the system load during the last 15 minutes.

Memory Monitoring Utilities

Memory monitoring utilities are used to view the usage of memory and other related statistics.

Memory Monitoring Utility	Enables You To
free	Display the total memory available on the system and the amount of memory that is free, used, shared, buffered, and cached.
vmstat	Display the statistics about virtual memory usage. It lists the details about currently running processes such as memory usage, interrupts or I/O address information, and processor allocation information.
pmap	Display the mapping of processes with memory resources.
iostat	Generate reports on CPU and device utilization. It provides input and output statistics for storage devices and partitions.

Command Options for the free Utility

There are several command options for the free utility.

Command Option	Used To
-b	Display the amount of memory in bytes, kilobytes, megabytes, and gigabytes, respectively.
-k	
-m	
-g	
-s {delay in seconds}	Update the memory statistics at a delay of the specified seconds.
-o	Disable the display of the buffer or cache information line.
-t	Display the total RAM and swap space.

Command Options for the vmstat Utility

The command options for the vmstat utility are listed in the table.

Command Option	Used To
-a	Display the active or inactive memory.
-s	Display memory statistics in a table format.
-m	Display statistics in the form of slabs.
-d	Display disk statistics.
-p {disk partition}	Display statistics for the specified partition.

Command Options for the pmap Utility

There are several command options for the pmap utility.

Command Option	Used To
-x {PID}	Report the memory map of processes in an extended format.
-d {PID}	Report the memory map of processes in a device format.
-q {PID}	Report the minimal required information of memory mapping.
-V	Display the version of the pmap utility.

	Note: PID (Process ID) is a unique number assigned by the operating system to each process started on the system.
--	--

Process Monitoring

Process monitoring is a mode of tracking the processes running on a system, to determine its performance and reliability. Some processes that run continuously on a system, such as those initiated by databases and web servers, have to be monitored constantly, whereas others can be monitored occasionally. Process monitoring enables a user to identify the causes of low performance in processes and detect the processes run by unauthorized users.

Process monitoring can be performed with the help of various utilities.

Utility	Enables You To
top	Display the list of processes in the descending order of CPU memory usage. It also displays details about the consumption of power, memory, and system resources at a given time. It helps track processes that consume high memory and system resources. The output of the command can also be redirected to a text file. In the KDE desktop environment, the kpm utility is used in place of the top command.
GNOME system monitor	Monitor system performance. It has three tabs, which list the performance history and status of various processes, resources, and filesystems on the system.
sar	Display the system utilization reports that are generated based on the system utilization data. These reports consist of various sections, each of which consists of the type of data and the time at which the data was collected. By default, the sar reports list the data collected every 10 minutes. On an average, the report consists of 17 sections. The sar command is run automatically by a script called sa2 at specified time intervals.

	Note: The ps command allows you to view the running processes on a system.
--	---

sar Options

The sar command can be used to retrieve specific data by specifying the following options. Some of the frequently used options are listed in the table.

Option	Enables You To
-A	Display all reports generated on the current date.
-b	Display I/O statistics.
-B	Display the number of bytes paged in between the system and the disk.
-c	Display the number of processes spawned per second by the system.
-d	Display system activity for each block device.

The GNOME System Monitor

The **GNOME system monitor** is a GUI utility that is used to monitor system processes, resources, and filesystems. The **Processes** tab displays details about the currently running processes such as the name, status, ID, and CPU and memory usage. The **Resources** tab displays the history of CPU, memory, and swap usage and network operations. The **File Systems** tab displays information about currently mounted filesystems, related directories, type, and usage status.

Figure 8-12: The GNOME system monitor displaying the system status.

How to Monitor Processes and Resources

Follow these general steps to monitor processes and resources.

Monitor the Kernel State

To monitor the kernel state:

1. Log in as **root**.
2. To view the information regarding the running kernel, on the terminal, enter `uname [options]`.
3. To view the running time of the system, enter `uptime`.
4. To view the graphical representation of the systems load average in the CLI, enter `load`.

Monitor the Memory Usage

To monitor the memory usage:

1. Log in as **root**.
2. To view the free and used memory of the system, on the terminal, enter `free [options]`.
3. To report the virtual memory statistics, enter `vmstat [options]`.

Monitor the Processes Mapping

To monitor the processes mapping:

1. Log in as **root**.
2. To view the running processes on the system, enter `ps [options]`.

3. To view the memory map of a process, enter `pmap [options]{pid}`.

Manage Processes Using the GNOME System Monitor

To manage processes using the GNOME system monitor:

1. Log in as **root** in the GUI.
2. To open the GNOME system monitor, select **Applications** → **System Tools** → **System Monitor**.
3. On the **Processes** tab, scroll to locate the process.
4. To start, stop, kill, or change priority, right-click on a running process.
5. To close the window, select **System Monitor** → **Quit**.

ACTIVITY 8-1

Managing Kernel Services Review

Scenario

Answer the following review questions.

1. Why do modules affect the way a kernel is loaded? Do you expect that you will need to load any special modules into the kernel to fulfill your organization's requirements?
2. Why is process management important for operating systems?

Summary

In this lesson, you explored the purpose and organization of the kernel and managed its services.

This will enable you to understand the kernel structure, monitor the kernel components, and configure the kernel services. As a Linux system administrator, customizing the kernel to suit your requirements will enable you to manage the kernel efficiently.

9 Working with the Bash Shell and Shell Scripts

Lesson Time: 2 hours

Lesson Introduction

In the previous lesson, you managed Linux® kernel services. In addition to kernel, the Linux shell is an important constituent of the operating system, and it is essential to familiarize yourself with the Bash shell and perform basic operations in it. In this lesson, you will work with the Bash shell and write shell scripts.

The Bash shell functions as an intermediary layer between a user and the operating system.

You can use shell scripts within the Bash shell to automate routine administrative tasks.

Although Bash is not the only shell available, it is one of the most common ones, and so familiarizing yourself with the Bash shell and its functions enables you to interact and work efficiently with the Linux operating system.

Lesson Objectives

In this lesson, you will work with the Bash shell. You will:

- Perform basic Bash shell operations.
- Write a basic shell script.
- Customize the Bash shell.
- Redirect standard input and output.
- Use control statements.

TOPIC A Perform Basic Bash Shell Operations

In previous lessons, you familiarized yourself with the way the Linux shell and CLI are used to manage the system. You are now ready to run commands in the shell to perform basic automation of frequently performed tasks. In this topic, you will perform basic Bash shell operations and create a Bash script.

The Bash shell is the most frequently used shell in Linux. It allows you to effectively perform tasks, such as file management, user and group administration, process management, text editing, and so on, using the command line. Basic Bash shell options allow you to perform simple tasks such as using strings to search for files on your system, reviewing commands that have been previously executed, and many more.

The Bash Shell

The *Bourne-Again SHell (Bash shell)* is the default shell in Linux. It is a superset of the Bourne shell and includes features from the Korn and C shells. The Bash shell facilitates command line editing, command history, command line completion, and shell scripting.

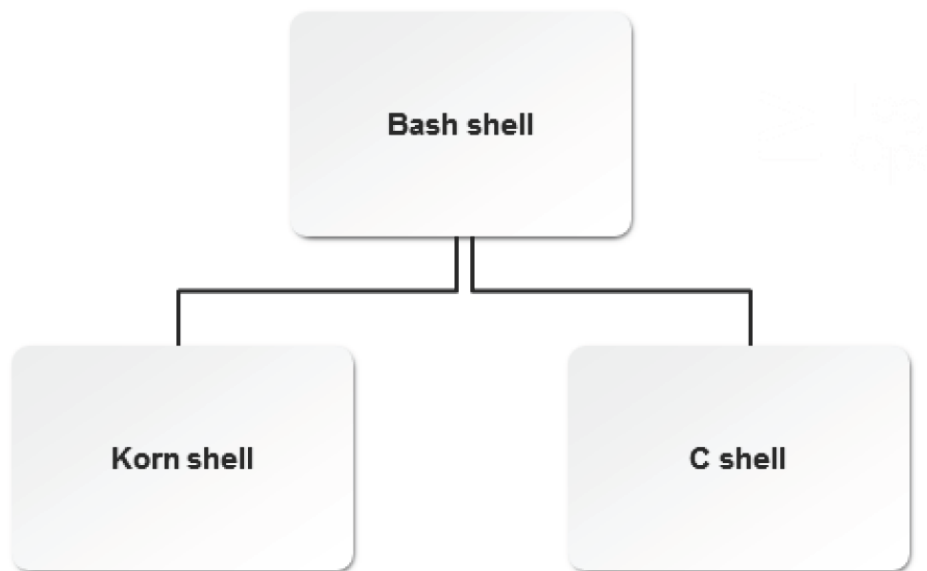


Figure 9-1: The Bash shell includes features from the Korn and C shells.

Bash Shell Functions

The shell is the basic component that provides the CLI in Linux. Some of the functions performed by the shell include:

- Prompting a user for input and waiting for a command to be entered.
- Verifying the correctness of the command and processing the command.
- Expanding wildcards by replacing special characters with portions of the string.
- Determining the source of input and the location of output.
- Returning to the prompt after the completion of a command and restarting the cycle.

The most commonly used shell in Linux, and the default for most popular Linux distributions including CentOS/RHEL, is the Bash shell.

Wildcards

Wildcards are special characters that are used to substitute portions of a string. By using wildcards with appropriate arguments, you can search and locate files on your system. Wildcards are used to narrow down search criteria and obtain accurate search results.

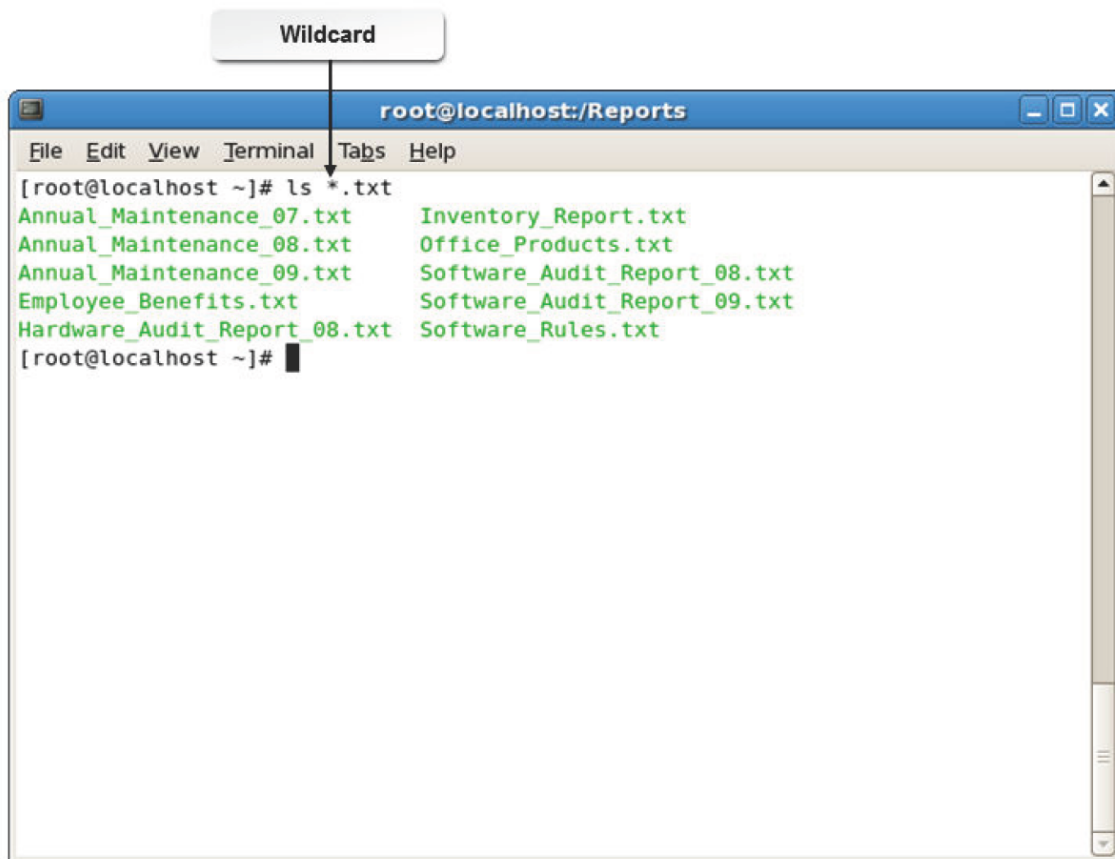


Figure 9-2: Using wildcards in a command.

The following table lists some of the frequently used wildcards.

<i>Wildcard</i>	<i>Used To</i>
*	Match zero or more characters in the file name.
?	Match a single character in the file name.
[abcde]	Match any of the listed characters.
[a-e]	Match any character in the range.
[!abcde]	Match any character that is not listed.
[!a-e]	Match any character that is not included in the range.
{linux, shell}	Match any word in the given options.
\$	List file names that end with the character preceding the \$ symbol.

Complex Wildcards

A **complex wildcard** is a combination of individual wildcards. For example, enter `[a-z]?[1-9]` to search for three characters—the first character is a letter, the last character a number, and the middle character can be a letter, a number, or a special character.

Globbing

Globbing is a function that expands file names (wildcards) using a pattern-matching behavior. The wildcards that globbing interprets are the asterisk (`*`), the question mark (`?`), sets of characters that are included within brackets (`[]` and `{ }`), and special characters such as the caret (`^`).

Tab Completion

Tab completion facilitates auto completion of commands and file names. Pressing **Tab** completes the names of commands, files, directories, users, and hosts.

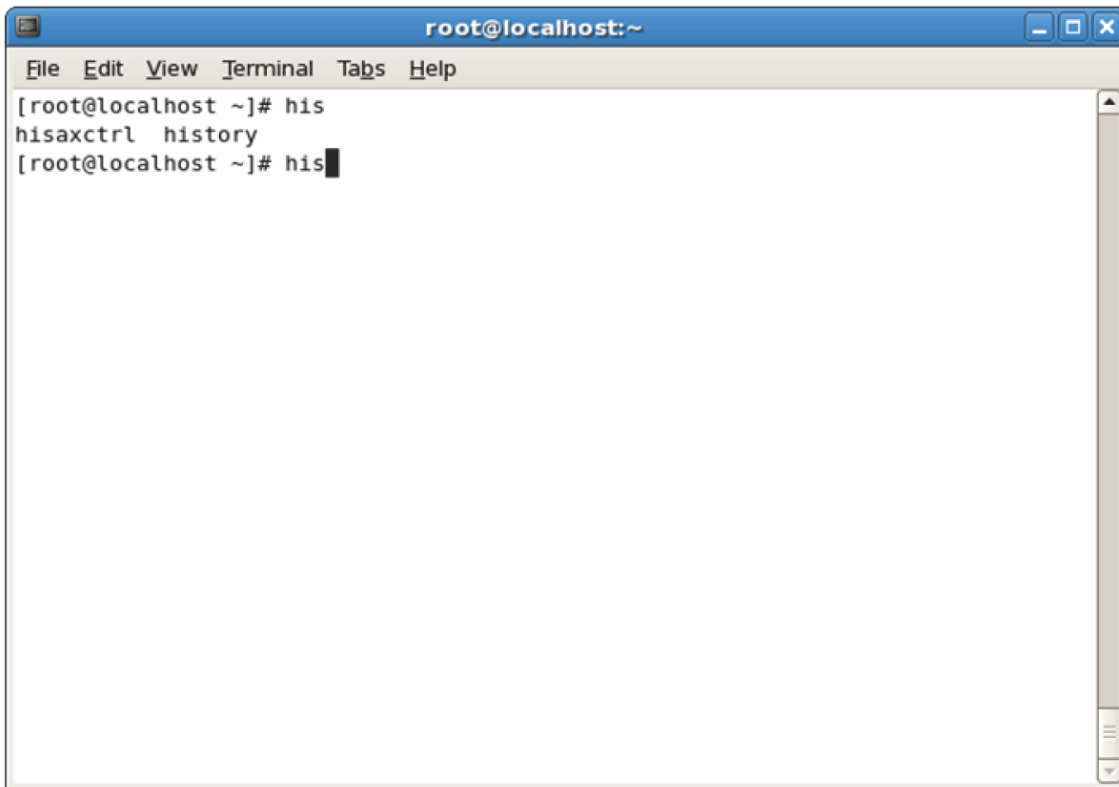


Figure 9-3: Tab completion entries for the text "his".



Note: Pressing **Tab** two times displays all files and directories that begin with the string you typed.

The history Command

The **history** command is used to view previously typed commands. It retrieves the specified number of commands from the `~/.bash_history` file. You can use the **Up Arrow** or **Down Arrow** key to select the desired command. By simultaneously pressing **Alt + Period (.)**, you can recall arguments that have been used with previously executed commands.

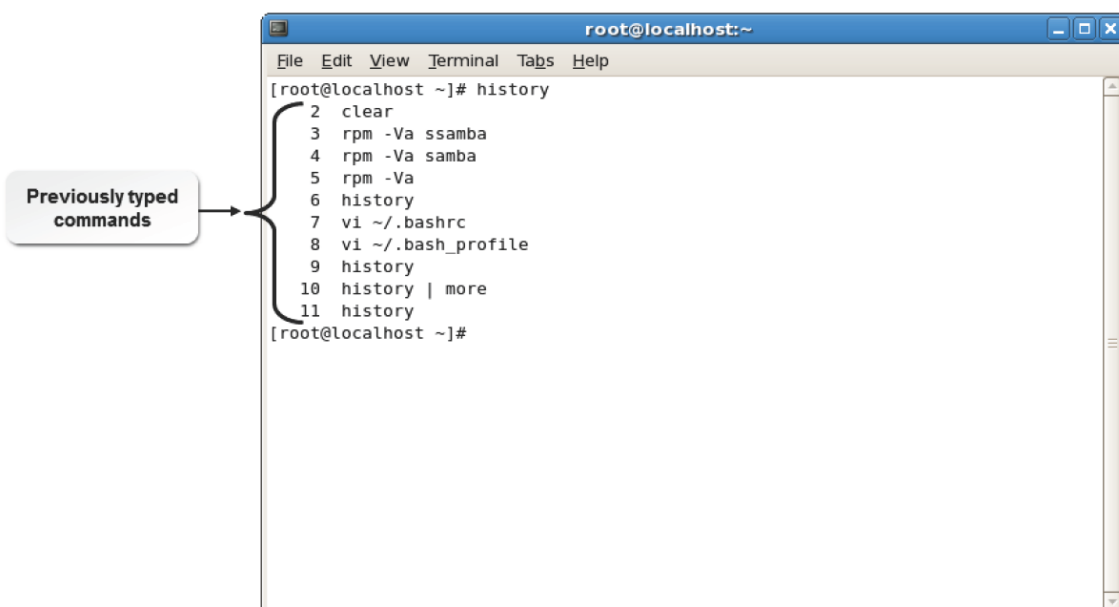


Figure 9-4: Output of the history command.

Recall Commands and Arguments

Pressing the **Up Arrow** key allows you to recall commands that have been run on the terminal.

Pressing **Esc** followed by **Period (.)** is an alternate way of recalling arguments.

How to Perform Basic Bash Shell Operations

Follow these general procedures to perform basic Bash shell operations.

Perform a Search Using Wildcards

To perform a search using wildcards:

1. Log in as a user in the CLI.
2. Perform basic operations.
 - To list all the content that matches the given pattern, enter `ls [wildcard]{string} [wildcard]`.
 - To remove all the content that matches the given pattern, enter `rm [wildcard]{string} [wildcard]`.
 - To print all the content that matches the given pattern, enter `echo [wildcard]{string} [wildcard]`.

Search for Files Using Wildcards

To search for files using wildcards:

1. Log in to the CLI as **root** and navigate to the relevant directory.
2. Locate the desired files using wildcards.
 - To view all files starting with the search string, enter `find -iname '[search string]*'`.
 - To view all files ending with the search string, enter `find -iname '*[search string]'`.
 - To view all files containing the search string, enter `find -iname '*[search string]*'`.
 - To do a more specific search, enter `find -iname '[complex wildcard][search string]'`.

Move Files that Meet a Wildcard Pattern

To move files that meet a wildcard pattern:

1. Log in and navigate to the relevant directory.
2. To move files that meet the search pattern to the target directory, enter `mv *[search string] /{target directory}`.
3. If necessary, to verify that the files have been moved to the target directory, enter `ls /{target directory}`.

Detect a File Name Using Tab Completion

To detect a file name using tab completion:

1. Log in as a user in the CLI.

2. To complete the command name, enter a unique character of the command name and press **Tab**.
3. To complete the file name, enter a unique character of the file name and press **Tab**.

View the Recently Used Commands Using the history Command

To view the recently used commands using the history command:

1. Log in as a user in the CLI.
2. View the history of commands executed.
 - To list all the previously used commands, enter `history`.
 - To execute a particular command from the command history, enter `!history number`.
 - To repeat the previously executed command, enter `!!`.

Perform Basic Command Line Expansion

To perform basic command line expansion:

1. Log in as a user in the CLI.
2. Perform basic command line expansion.
 - Enter `{command} {common string}{unique part of file 1, unique part of file 2}` when a file or directory name has a common string in it.
 - To print a string, enter `{command} '{string}'`.
 - To send the output of one command as the input to another command, enter `{command 2} `{command 1}``.

TOPIC B Write a Bash Shell Script

In the last topic, you worked with the basic Bash shell options to perform various tasks. To execute complex tasks using the operating system, it is essential to script a program for the respective task.

In this topic, you will write a basic Bash shell script.

An in-depth knowledge of shell scripts is required to understand the working of the Linux system.

As a Linux administrator, it is essential for you to work with shell scripts because they enable you to automate routine tasks, saving time and effort.

Shell Scripts

A *shell script* is a file that contains a list of commands to be read and executed by the shell.

Frequently used commands can be stored in a shell script for repeated use. Every shell script starts with a line that designates the interpreter. This line instructs the operating system to execute the script.

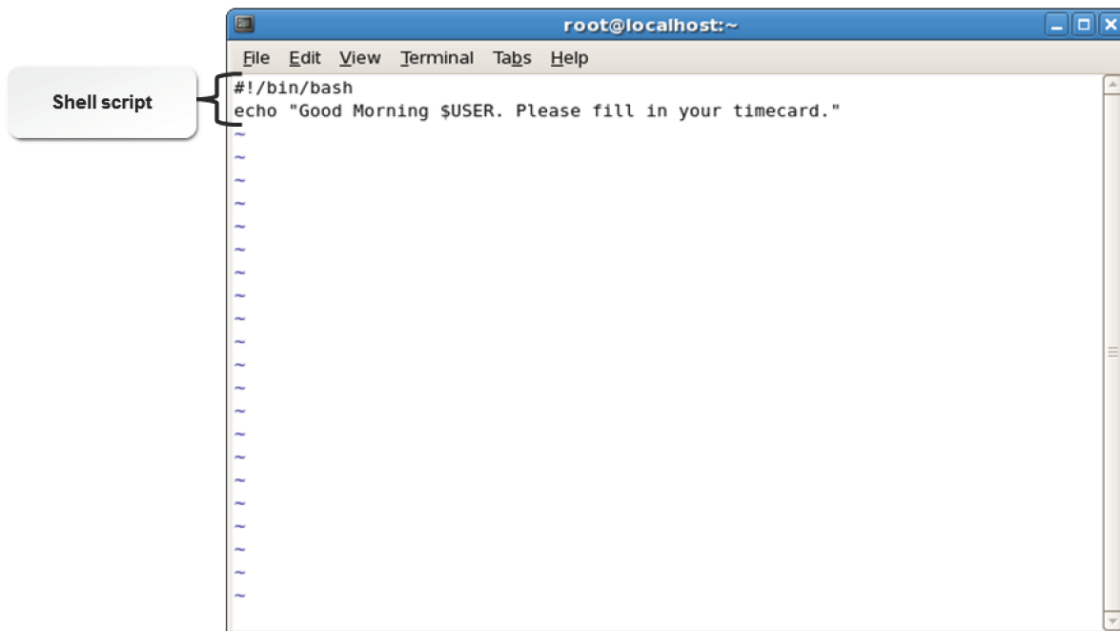


Figure 9-5: Creating a shell script.

Shell scripts allow you to perform various functions. These functions are listed in the following.

- Automation of commands and tasks of system administration and troubleshooting.
- Creation of simple applications.
- Manipulation of text or files.

The Bash shell facilitates command line expansion, inhibition, and substitution with specific symbols called command line operators. These operators are described in the table.

<i>Operator</i>	<i>Description</i>
\$	Expands variables
`	Substitutes commands
\	Inhibits a single character
!	Substitutes history

Calling the Correct Interpreter

When you write an sh script, ensure that the first line is `#!/bin/sh`. For a bash script, the first line is `#!/bin/bash`. The first line that contains `#!` is referred to as the shebang line.

`#!/bin/bash`

Bash scripts contain shell-specific instructions that may not be compatible with other Linux shells.

This will result in a Bash script running on Bash shells correctly, while failing on other non-Bash shells in Linux. To specify that your script is written for the Bash shell, you need to add a line `#!/bin/bash` at the beginning of each script. This line will instruct the operating system to use the Bash shell when executing a script on an incompatible Linux shell.



Note: The special `#!` prefix, consisting of a number sign (hash mark) and exclamation point (bang), that precedes the designated shell interpreter for that file are sometimes referred to as a "shebang."

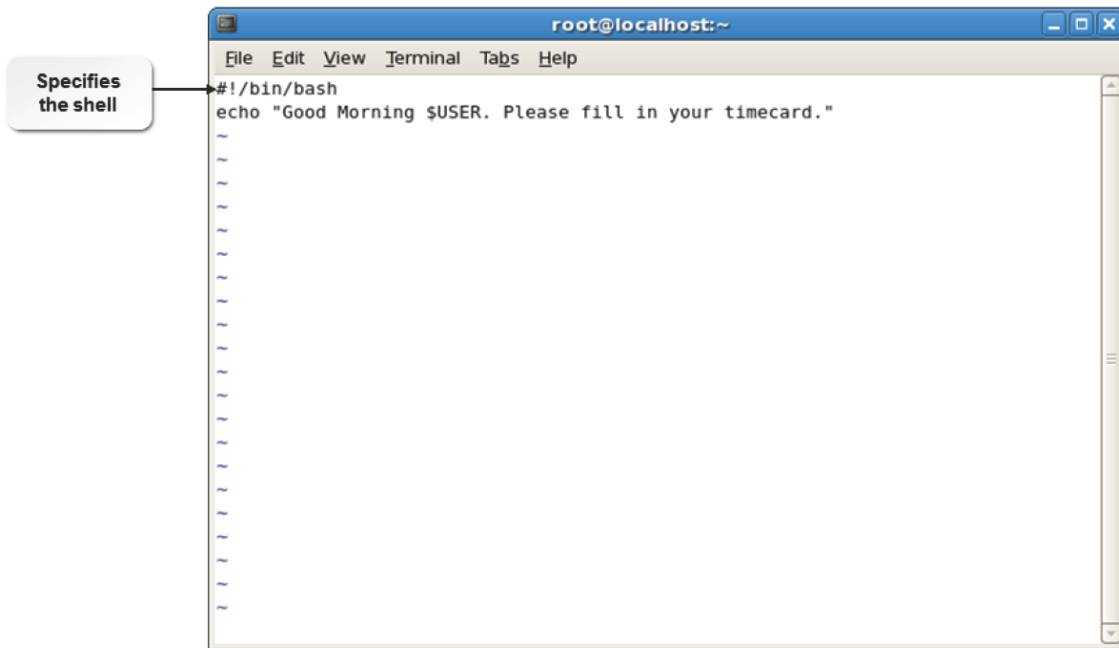


Figure 9-6: *#!/bin/bash will enable Bash scripts to run on all Linux shells.*

The test Command

The test command is used to check file types and compare values. You can use the test command in your shell scripts to validate the status of files and perform relevant tasks. It evaluates a conditional expression and displays an exit status. The exit status is 0 if the expression is true, 1 if the expression is false, and 2 if an error occurs.

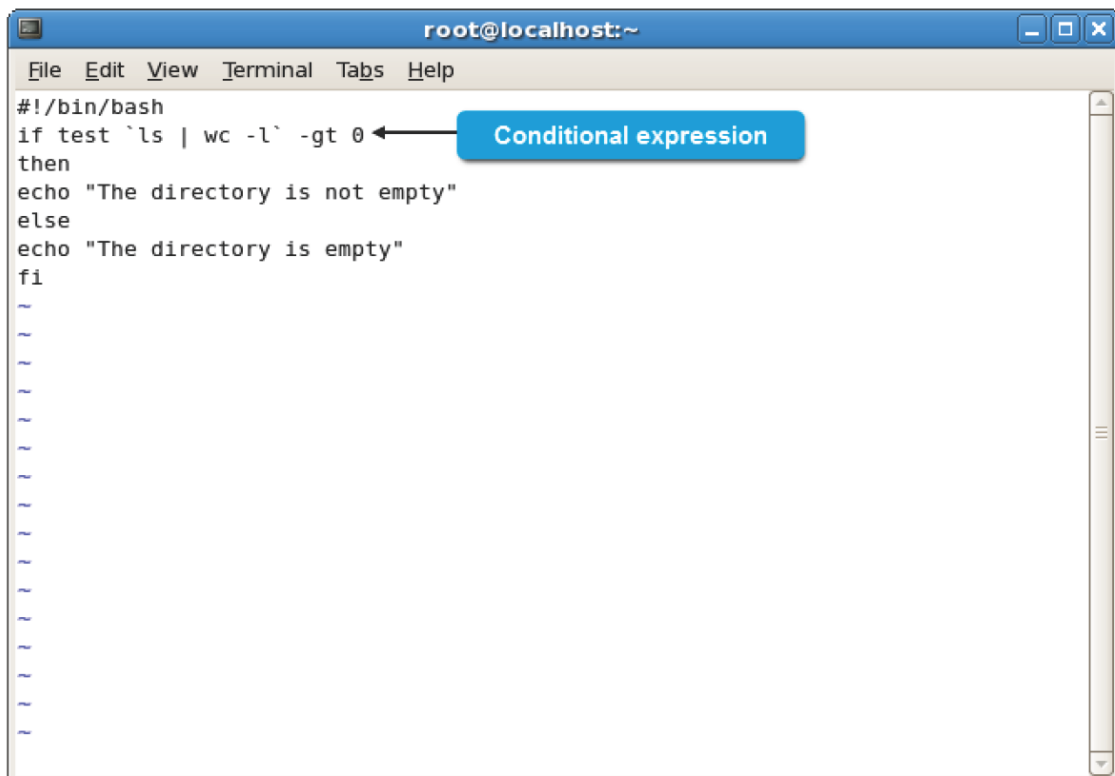
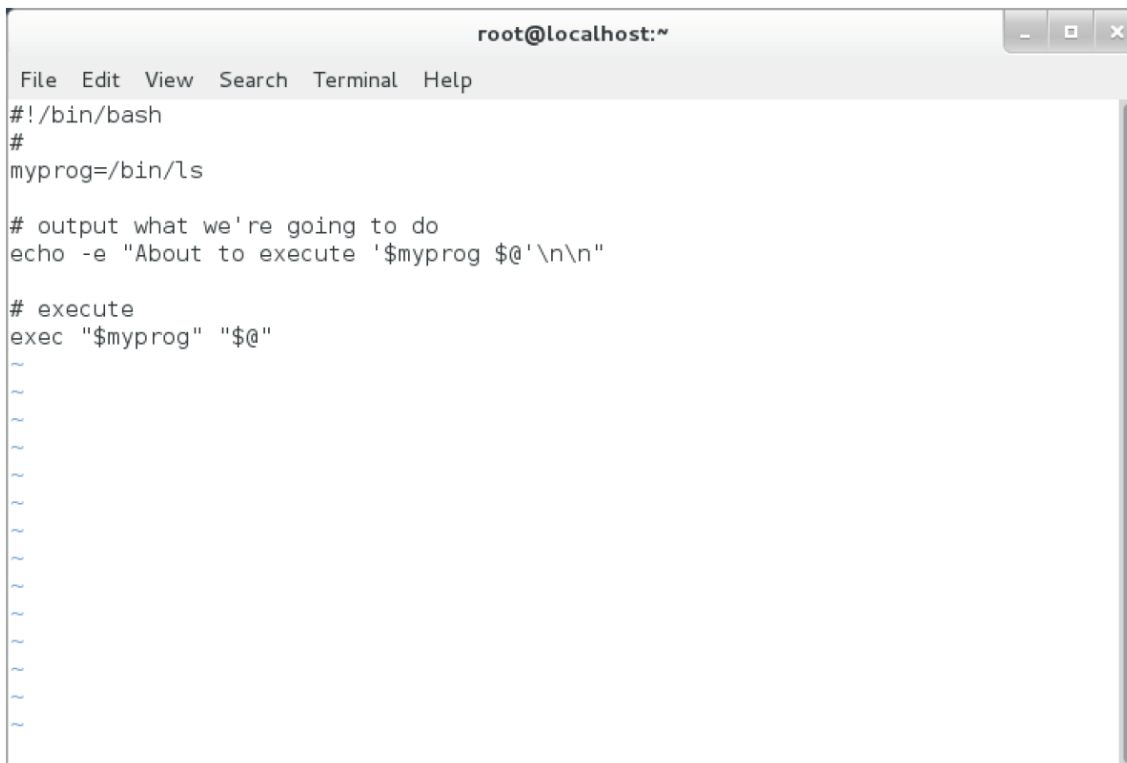


Figure 9-7: *The test command is used to determine whether a directory is empty or not.*

The exec Command

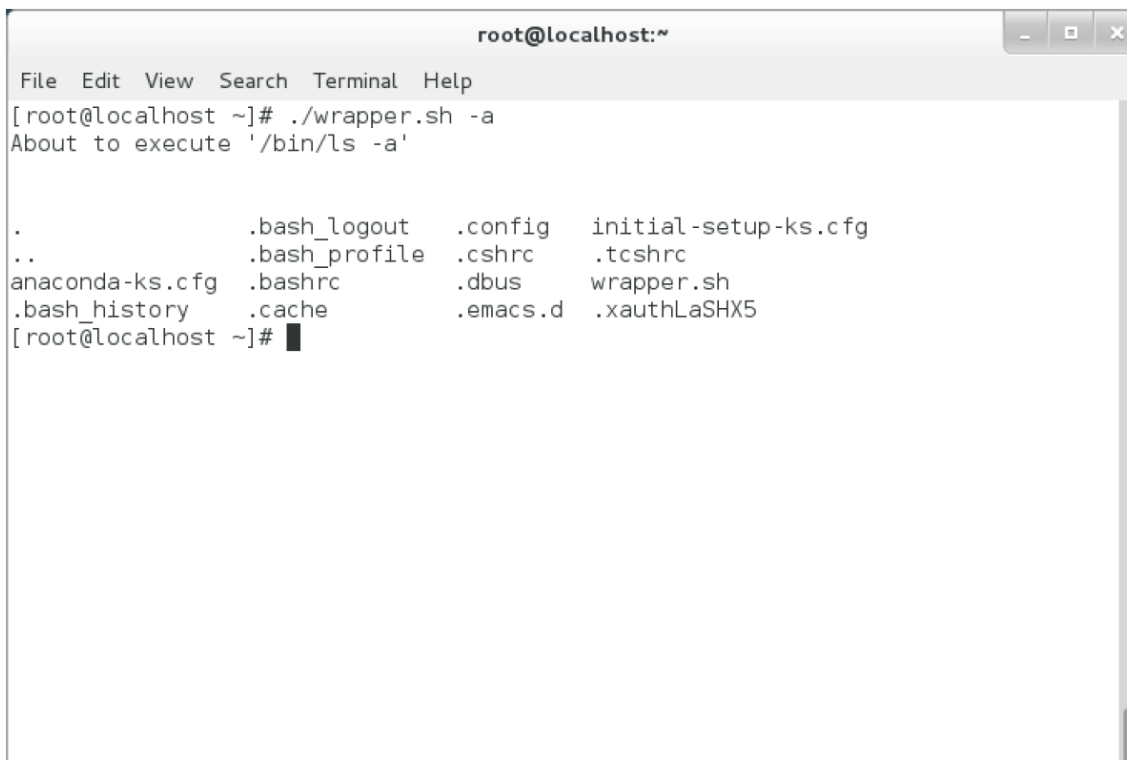
The exec command is used to execute another command, replacing the current shell process with this new program's process (no new process is created). You can also use the exec command in your shell scripts to set redirections for the program to execute or for the current shell. The exec command is most often used to execute

another Linux command programmatically, for example if you want to create a Bash shell script that would be a wrapper for another program.



```
root@localhost:~  
File Edit View Search Terminal Help  
#!/bin/bash  
#  
myprog=/bin/ls  
  
# output what we're going to do  
echo -e "About to execute '$myprog $@'\n\n"  
  
# execute  
exec "$myprog" "$@"  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

Figure 9-8: The `exec` command is used in this example wrapper script to output the full command line of the command before it is executed.



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# ./wrapper.sh -a  
About to execute '/bin/ls -a'  
  
.  
..  
anaconda-ks.cfg  
.bash_history  
.bash_logout  
.bash_profile  
.bashrc  
.cache  
.config  
.cshrc  
.dbus  
.emacs.d  
initial-setup-ks.cfg  
.tcshrc  
wrapper.sh  
.xauthLaSHX5  
[root@localhost ~]#
```

Figure 9-9: When executed, this Bash script outputs the `ls` command it will run and then executes that command.

Editing a Shell Script

To edit a shell script, open the script file in the vi editor, make the necessary changes to the shell script, and then save the file. The table lists the common default shell scripts and their uses.

<i>If You Need To</i>	<i>Use This Default Shell Script</i>
Set user profile variables	~/.bash_profile
Set user login commands	~/.bash_login
Set global profile variables	~/.profile
Set shell variables	~/.bashrc
Set user logout commands	~/.bash_logout
Map the keyboard for situations such as the sound to be played on reaching the end line	~/.inputrc

How to Write a Shell Script

Follow these general procedures to write a shell script.

Write a Bash Script

To write a Bash script:

1. Log in as a user in the CLI.
2. To write a Bash shell script, enter `vi {script file name}`.
3. To switch to insert mode, press **I**.
4. Type `#!/bin/bash` to specify the shell.
5. Type the required command.
6. To return to command mode, press **Esc**.
7. To save the file and exit the text editor, enter `:wq`.

Convert a Script File to an Executable Script

To convert a file to an executable script:

1. Log in as a user in the CLI.
2. Write a Bash shell script.
3. To convert the file to an executable script, enter `chmod a+x {script file name}`.

Execute Scripts Using the Relative Path

To execute scripts using the relative path:

1. Log in as a user in the CLI.
2. Navigate to the directory where the script file is located.
3. To verify that the script file has execute permissions, enter `ls -l`.
4. If necessary, convert the script to an executable script.
5. To execute the script, enter `./{script file name}`.

Syntax

The syntax of the export command is export {*variable*}.

Viewing Variable Values

Variables that are set by the operating system when you log in to a system are automatically exported. Variables created by the shell remain local in scope unless you manually export them. To display the value of a variable, use the echo command followed by a dollar sign (\$) and the variable name (with no space between the \$ and the variable name). For example, to view your default shell, enter echo \$SHELL. The value of the variable is displayed on the screen. Like Linux commands, the shell variables are also case sensitive.

Declaring Variables

In addition to using and modifying predefined variables, you can also create variables. To create variable names, apply the following rules:

- A variable name must begin with an upper or lowercase letter or an underscore.
- The initial letter or underscore can be followed by any number of additional upper or lowercase letters, numbers ranging from 0 to 9, or an underscore.
- The following character combinations have special meanings and should not be used as variable names or to end a variable name: \$@, \$#, \$\$, \$*, \$-, \$_, \$?, and \$0 to \$9.

To assign a value to a variable, type the variable name followed by an equal sign and the value (with no spaces). To export a variable, making it accessible to commands and other shells, type export followed by the variable you want to export.

Special Shell Parameters

Shells treat some characters specially. Such characters cannot be assigned to variables because they convey special meaning. The following table contains the list of special characters and their description.

<i>Character</i>	<i>Description</i>
*	Signifies the positional parameters, starting from one.
@	Signifies the positional parameters, starting from one. When the expansion occurs within quotation marks, each parameter expands to a separate word. For example, "\$@" is equivalent to specifying "\$1" "\$2". When there are no positional parameters, "\$@" and \$@ are removed and do not expand to anything.
#	Signifies the number of positional parameters in decimal.
?	Signifies the exit status of the most recently executed foreground pipeline.
-	Signifies the current option flags as specified upon invocation, by the set built- in command, or those set by the shell itself using the -i option.
\$	Signifies the PID of the shell. In a subshell, it expands to the PID of the invoking shell, not the subshell.
!	Signifies the PID of the most recently executed background command.
0	Signifies the name of the shell or shell script. This is set at shell initialization. If Bash is invoked in a shell script file, \$0 is set as the name of that file.
_	Signifies the absolute path name that is used to invoke the shell or shell script being executed as passed in the environment or argument list.

Working with the CDPATH Variable

In your directory structure, there may be directories that you will want to frequently switch between.

By defining the required directory path in the CDPATH variable, you can easily switch to that path.

Environment Variables

An **environment variable** is a storage location in the operating system's command shell. It is accessible by all programs. An environment variable consists of a name, usually written in uppercase letters, and a value, such as a path name. Environment variables can be directly viewed from the shell.

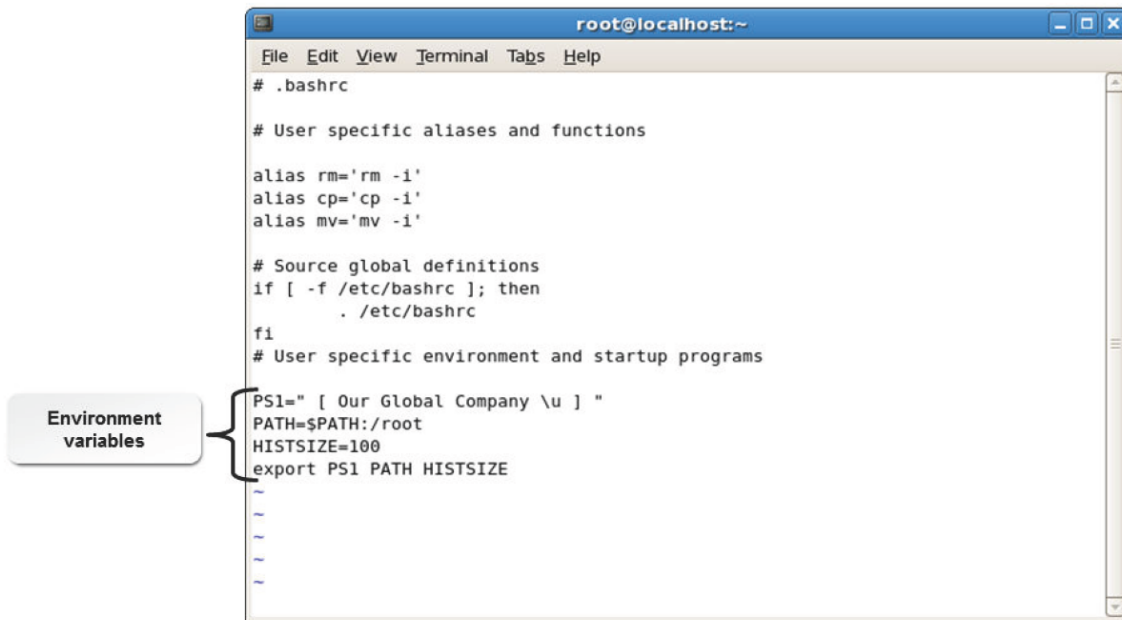


Figure 9-11: Assigning values to environment variables.

Referencing Environment Variables

You can use the existing environment variable in a new or existing shell by referring to it as \${environment variable}.

Default Environment Variables

Some of the default environment variables and their functions are provided in the following table.

Environment Variable	Specifies
HOSTNAME={hostname}	The hostname of the system.
SHELL={shell path}	The shell path for the system.
MAIL={mail path}	The path where the mail will be stored.
HOME={home directory}	The home directory of the user.
PATH={user path}	The path in which the user needs to operate.
HISTSIZE={number}	The number of entries to be stored in the command history.
USER={user name}	The name of the user.
EDITOR={text editor name}	The preferred text editor for the environment.
TERM={terminal name}	The name of the terminal used.
PRINTER={printer name}	The default printer of the system.
PAGER={command}	The command through which the content of long files needs to be listed.
PS1=[prompt]	The primary prompt—the prompt that is displayed on login.

<i>Environment Variable</i>	<i>Specifies</i>
PS2=[prompt]	The secondary prompt.

Local and Environment Variables

When you define a variable in a shell script, it is called a local variable of that particular shell script.

This cannot be used by other shell scripts or outside that shell script. When you export the local variable using the export command, it becomes an environment variable, which can be used by other shell scripts or by the command line.

Referencing Environment Variables

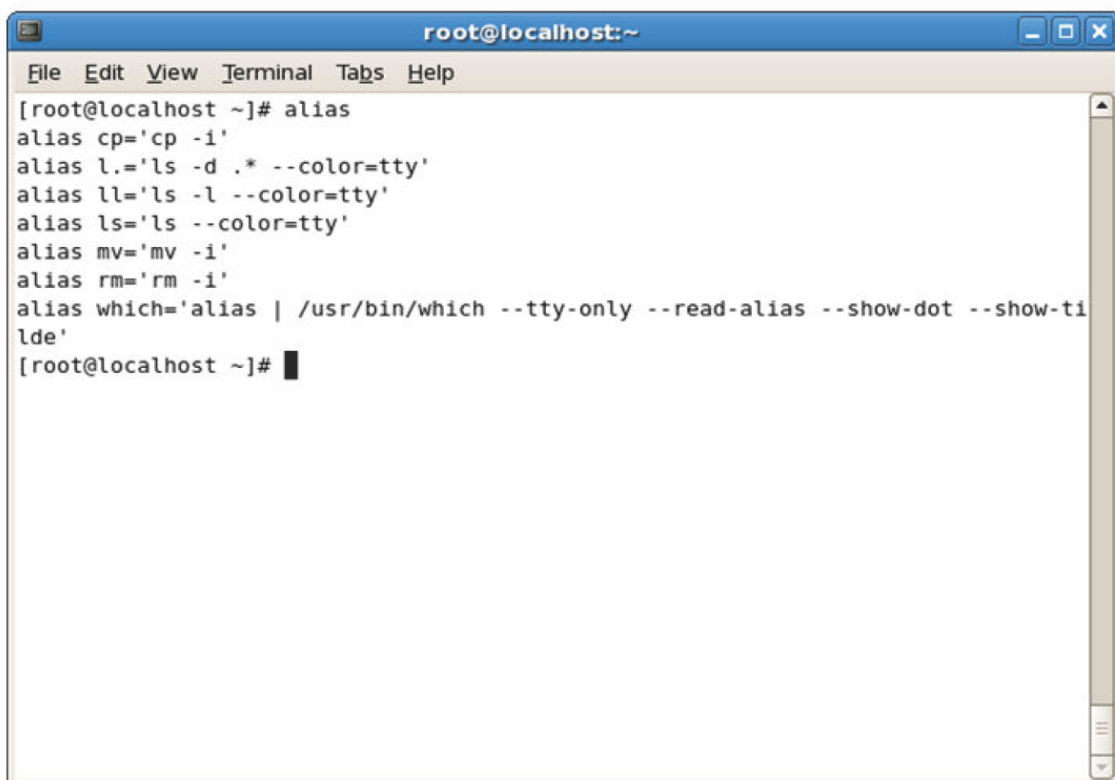
You can use the existing environment variables in a new or existing shell wherever necessary by referring to it as `${ENVIRONMENT VARIABLE}`.

Bash shell configuration

You can use the `/etc/profile` and `/etc/bash.bashrc` configuration files to run commands when the bash shell is invoked. These files are used to set configuration options, environment variables, and system settings that apply to all system users. Per-user configuration is managed via the `~/.bash_profile` and `~/.bashrc` configuration files in each user's home directory.

The alias Command

The [alias](#) command is used to generate command line aliases. Aliases are shorthand for longer expressions. Using aliases, you can substitute a word in a command with a string. The shell maintains a list of aliases that are created and listed using the alias command. It also maintains a list of aliases that are removed using the unalias command.



```

root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# alias
alias cp='cp -i'
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias mv='mv -i'
alias rm='rm -i'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-ti
lde'
[root@localhost ~]#

```

Figure 9-12: Viewing command line aliases.

Syntax

The syntax of the alias command is `alias {alias name}={command} [options]`.

HISTFILESIZE

The HISTFILESIZE environment variable allows you to set the maximum number of lines contained in the history file. It also allows you to specify the number of lines to be displayed on running the history command. For example, on assigning a value of 20 to this variable, the history file gets truncated to contain just 20 lines. The default value to this variable is 1,000.

Figure 9-13: Assigning a value to the HISTFILESIZE variable.

SUID Scripts

An **SUID script** is a program that overrides normal permissions and runs with the permissions of the owner of the program. The chmod command is used to set the SUID to the shell script. Care should be taken to secure the script with the SUID because it is prone to being hacked.

Figure 9-14: A sample SUID script.

Shell Spawning

Shell spawning is a process that allows a shell to create a clone of itself. The copy is called the child process. It becomes the new process and can also create more processes, which result in multiple generations of processes. The shell spawns a child process when the user enters a command. The shell waits for the child process to be completed before displaying the prompt again to accept another command.

Search Paths

A **search path** is a sequence of various directory paths that is used by the shell to locate files. Paths can be assigned to the **PATH** environment variable. The PATH variable comprises a list of directory names separated by colons. You can add a new path to an existing group of path names, modify a path, or delete a path. Usually, directories that contain executable files are assigned to the PATH variable.

Figure 9-15: A search path defined in the .bash_profile file.

The following is an example of a search path where the path names are separated by colons: /

```
home/usr/bin:/usr/local/bin:/usr/bin:/bin.
```

The following is not a search path because the path names are not separated by colons: /usr/

```
bin /usr/local/bin.
```

Default Location of the PATH Variable

Whenever you execute a command or script, the system by default searches the location specified in the default PATH variable. Only the commands or scripts present in these locations can be executed from any location.

The default location of the PATH variable for a user is as follows:

- /usr/local/bin
- /bin
- /usr/bin

The default location of the PATH variable for a root user is as follows:

- /usr/local/bin
- /bin
- /usr/bin
- /usr/local/sbin
- /sbin

- /usr/sbin

How to Use Shell Variables

Follow these general procedures to use shell variables.

Define Variables

To define variables:

1. Log in as a user in the CLI.
2. To define a variable, enter `{VARIABLE}={value}`.
3. To display the value associated with the variable, enter `echo ${VARIABLE}`.

Set an Alias for a Frequently Used Command

To set an alias for a frequently used command:

1. Log in as a user in the CLI.
2. To view the default aliases, enter `alias`.
3. To set an alias for a frequently used command, enter `alias {alias name}='{command}'`.
4. To view the updated alias on the system, enter `alias`.

Set Configuration Variables

To set configuration variables:

1. Log in to the CLI as **root**.
2. To open the `.bashrc` script file, enter `vim ~/.bashrc`.
3. Make the necessary changes such as changing **PS1**, **HISTSIZE**, **PATH**, or **alias**.
4. Export the variables and save the file.
5. For the changes to apply, log out and log in as **root**.

Specify the Script Location to the PATH Variable

You can manually add the location of a shell script file to the default `PATH` variable and execute the script from any location. To specify a script location to the `PATH` variable:

1. Log in as user in the CLI.
2. Enter `vi {script file name}`.
3. Write the script commands and save the file.
4. To make the file an executable script, enter `chmod a+x {script file name}`.

5. Log out as user and log in as **root**.
6. Enter `vi .bash_profile`.
7. To add the location of the script file to the default PATH variable, in the line starting with "PATH=", move to the end of the line and type `:{location of the script file}`.
8. Save and exit.
9. Log out and log in the CLI.
10. To execute the file, enter `{script file name}`.

Add Script to the Location in the PATH Variable

You can manually add the shell script file to the default location specified in the *PATH* variable to execute the script from any location. To add script to the default location in the PATH variable:

1. Log in as user in the CLI.
2. Enter `vi {script file name}`.
3. Write your script and save the file.
4. To convert the file into an executable script, enter `chmod a+x {script file name}`.
5. Log out as user and log in as **root**.
6. To move the script to the default location in the PATH variable, enter `mv /home/User/{location of the script file}/{script file name} /{default location of the PATH variable}`.
7. Log out as **root** and log in as user in the CLI.
8. Enter `{script file name}` to execute the script.

Manage Shell Script Ownership

To manage shell script ownership:

1. Log in to the CLI as **root**.
2. View the owner of the shell script file.
 - a. Navigate to the directory where the shell script is located.
 - b. To view the ownership of the shell script file, enter `ls -l`.
3. Change the owner.
 - To change the user, enter `chown {user name} {script file name}`.
 - To change the group owner, enter `chgrp {group name} {script file name}`.

Manage Execution of the Shell Script

To manage execution of the shell script:

1. Log in as a user in the CLI.
2. Navigate to the directory where the shell script is located.
3. To convert the file into an executable script, enter `chmod a+x {script file name}`.
4. Execute the script.
 - Enter `{script file name}`.
 - Enter `./{script file name}`.
 - Enter `/{directory where the script file is located}/{script file name}`.
 - Add the directory path of the script file to the PATH variable in the `~/.bash_profile` file and then log out to apply the setting. Login and enter `{script file name}`.
 - Add the script file to the default location in the PATH variable and enter `{script file name}`.

Manage the SUID Rights of Shell Scripts

To manage the SUID rights of shell scripts:

1. Log in as a user in the CLI.
2. Navigate to the directory where the shell script is located.
3. To set the UID to the script file, enter `chmod u+s {script file name}`.

Set Environment Variables When Spawning a New Shell

You can create a new variable and make it available to a new shell or to an existing shell. To set environment variables when spawning a new shell:

1. Log in to the CLI as **root**.
2. If desired, to view the environment variables, enter `env`.
3. Enter `vi .bash_profile`.
4. Define and export an environment variable at login.
 - a. Switch to input mode.
 - b. To define a variable, enter `{ENVIRONMENT VARIABLE}={value}`.
 - c. To export the variable, type `export {ENVIRONMENT VARIABLE}`.
 - d. Save and close the file.
5. Log out and log in to apply the changes.
6. If necessary, to verify whether the value that was set for the environment variable is applied, enter `echo ${ENVIRONMENT VARIABLE}`.
7. Enter `vi {script file name}`.

8. Enter the script to use the environment variable in a new shell.
 - a. Switch to input mode.
 - b. Enter `#!/bin/bash`.
 - c. To refer to the variable and get its value, enter the desired script and use `${ENVIRONMENT_VARIABLE}`.
 - d. Save and close the file.
9. To convert the file to an executable script, enter `chmod a+x {script file name}`.
10. To execute the script file, enter `./{script file name}`.

Set Environment Variables Without Spawning a New Shell

While testing changes to your `.bashrc` or `.bash_profile` default config files, you may want to set environment variables without spawning a new shell. To do so, you will use the source bash shell built-in command (which also has a synonym in `'.'` (period)). To set environment variables without spawning a new shell:

1. Log in to the CLI as **root**.
2. If desired, to view the environment variables, enter `env`. and verify that no variable named `variable_name` currently exists.
3. Enter `vi ~/.bashrc`.
4. Define and export an environment variable at login.
 - a. Switch to input mode.
 - b. To define a variable, enter `variable_name=variable_value`.
 - c. To export the variable, type `export _variable_name`.
 - d. Save and close the file.
5. To source the file, enter `source ~/.bashrc`.
6. To view the environment variables, enter `env` and verify that the variable named `variable_name` now exists and is set to `variable_value`.
7. Enter `vi ~/.bashrc` again.
8. Now change the value of your variable.
 - a. Switch to input mode.
 - b. Redefine the variable, enter `variable_name=new_variable_value`.
 - c. Save and close the file.
9. To source the file, enter `. ~/.bashrc`.
10. To view the environment variables, enter `env`. and verify that the variable named `variable_name` now exists and is set to `new_variable_value`.

TOPIC D Redirect Standard Input and Output

In the previous topic, you performed shell scripting using variables. As part of automating tasks using scripts, you may now want to manipulate the input and output of Linux commands and files.

In this topic, you will redirect standard input and output.

Imagine that you need to create a troubleshooting report, which contains a command and the respective errors it generates after execution. In this situation, instead of keying the output or errors in the report, you can redirect the output of the command into the report. Redirection techniques help you accomplish certain tasks with speed and ease.

Standard Input

Standard input, or **STDIN**, is a **text stream** that acts as the source for command input. Usually standard input for the Linux command line is from the keyboard. In the case of the GUI, the standard input can also be from the mouse. The standard input stream is buffered and lends itself to be redirected.

Figure 9-16: Standard input entered using the keyboard.

The read Command

The read command is used to read content from the standard input, the keyboard, and assign it to a variable.

Standard Output

Standard output, or **STDOUT**, is a text stream that acts as the destination for command output.

By default, standard output from the Linux command is directed to the terminal screen. The standard output stream is buffered and lends itself to be redirected.

Figure 9-17: Standard output displayed on the terminal.

The seq Command

The **seq** command prints a sequence of numbers on the standard output. It allows you to specify a start value, an end value, and an incremental value. The syntax of the seq command is: seq

[start value][increment value]{end value}. For example, seq 12 will display numbers

from 1 to 12, while seq 2 500 will display all the odd numbers from 1 to 500.

Standard Error

Standard error, or *STDERR*, is a text stream that is used as the destination for error messages.

The STDERR stream is not buffered. By default, the standard error stream prints error messages on the terminal screen, but this can be changed by redirecting it to the desired location.

Figure 9-18: Standard error message displayed on the monitor.

Redirectors

A *redirector* is an operator that accepts input data from a source other than the keyboard or sends data to a destination other than the monitor. It generally uses files as input or output. A redirector can redirect the output of a command to serve as the input for another command. It can also send output data to both the screen and a file.

Figure 9-19: Output from a command redirected to a file.

There are some operators that are used to redirect input or output. The functions of frequently used operators are described in the following table.

<i>Operator</i>	<i>Enables You To</i>
>	Redirect the standard output to another file.
>>	Append the standard output to the end of the destination file.
2>	Redirect the standard error message to a file.
2>>	Append the standard error message to the end of the destination file.
&>	Direct all the output of a command to a file.
<	Read the input from a file rather than from the keyboard or mouse.
<<string	Provide input data from the keyboard, indicating the end with the specified string.
=	Assign values to variables.
= =	Check if two values are equal to each other.

	Note: The semicolon (;) is used to separate variables, commands, or values.
--	--

Examples of Redirection

mail < myletter.txt: The **myletter.txt** file will be taken as the input.

ls > file1.txt: The output of the ls command will be redirected to a file named **file1.txt**.

ls file3.txt 2> errorfile.txt: Assuming that **file3.txt** does not exist, the resulting errors will not be displayed on the screen, but they will be redirected to a file named *errorfile.txt*.

The Pipe Operator

The **pipe** is an operator that combines commands. It uses the standard output of one command as the standard input for another command. The output format of the first command should be compatible with the format that the second command works with. The pipe operator can be used with most commands in Linux.

Figure 9-20: The pipe operator is used to combine two Linux commands.

Example of Commands Using the Pipe Operator

The output of the ls command is the input for the more command.

Lists in Shell Scripts

In shell scripts, you can write scripts to execute a number of commands in sequence by creating a list. This list can be created by using one of the four symbols, ";", "&", "&&", or "||" to separate the commands. You can use ";" or "&" as the terminal character.

The operators used in shell scripts are listed in the following table.

Operator	Description
;	Separates commands, which will be executed one after the other.
&	Executes the preceding pipeline as a background task.
&&	Executes only if the preceding command or pipe is terminated normally.
	Executes only if the preceding command or pipe terminated with an error.

The xargs Command

The **xargs** command constructs and executes command lines. The pipe operator is used to make the output of the first command the input for the second command. The xargs command adds arguments from the standard input to complete the command and then executes it.

The xargs command has various options.

<i>Option</i>	<i>Used To</i>
-l {replacement string_	Consider each line in the standard input as a single argument.
-L {number of lines}	Read a specified number of lines from the standard input and concatenate them into one long string.
-p	Prompt the user before each command.
-n {number}	Read the maximum number of arguments from the standard input and insert them at the end of the command template.
-E {end of string}	Represent the end of the standard input.
-t	Write each command to the standard error output before executing the command.
-s {maximum allowable size}	Set the maximum allowable size of an argument list to a specified number of characters.
-x	Terminate the xargs command if it creates a command that is longer than the arguments given in the -n option, is longer than the number of lines given in the -L option, or is longer than the size given by the -s option.

The tee Command

The tee command reads the standard input, sends the output to the standard output device, and also copies the output to each specified file. This command enables users to log the output of a command in a file before sending it as the input to the next command; therefore, it serves as a helpful tool in troubleshooting. When used with the -a option, it appends the output to each output file instead of overwriting it. When used with the -i option, it ignores interrupt signals.

Figure 9-21: Output of the df command redirected to a file.

Command Substitution

Command substitution is the ability to reassign the output of a command as an argument to another command. The command line that needs to be reassigned is placed within back quotes (` `), sometimes referred to as

backquotes or backticks. First, the shell executes the commands enclosed within the back quotes. Then, it replaces the entire expression, including the back quotes, with the output of the command.

Single Quotation Marks and Quotation Marks

Unlike back quotes, single quotation marks (') are used in a shell command to disable any kind of transformation or modification. The shell considers whatever is enclosed within the single quotation marks as a single entity or parameter. If single quotation marks are used, substitution will not take place.

By employing quotation marks (" "), the expansion of the file name is suppressed by the shell.

Even if a wildcard, such as the asterisk (*), is enclosed within quotation marks, the standard feature of the wildcard (matching all characters) will be lost.

How to Redirect Input and Output

Follow these general procedures to redirect input and output.

Redirect the Standard Output to a File

To redirect the standard output to a file:

1. Log in as a user.
2. Redirect the standard output to a file.
 - To direct the standard output of the command to the specified file, at the command prompt, enter *{command} > {file name}*.
 - To append the standard output of the command to the end of the specified file, enter *{command} >> {file name}*.

Redirect the Standard Error as Output to a File

To redirect the standard error as output to a file:

1. Log in as a user.
2. Redirect the standard error to a file.
 - To direct the error message from the command to the specified file, enter *{command} 2> {file name}*.
 - To append the error message from the command to the end of the specified file, enter *{command} 2>> {file name}*.

Redirect the Standard Output to a Command

To redirect the standard output to a command:

1. Log in as a user.
2. To redirect the output of one command as input to another command, enter *{command 1} | {command 2} | {command 3}*.

Redirect the Standard Output to a File and Command

To redirect the standard output to a file and command:

1. Log in as a user.
2. To save the output at various stages in files and direct the output to other commands, enter `{command 1} | tee {file name 1} | {command 2} | tee {file name 2} | {command 3}`.

Redirect the Standard Output and the Standard Error to a File and Command

To redirect the standard output and the standard error to a file and command:

1. Log in as a user.
2. Redirect the standard output and the standard error.
 - To direct the standard output of the command to a file and the standard error message to another file, enter `{command} > {file name 1} 2> {file name 2}`.
 - To direct all the output of the command to a file, enter `{command} &> {file name}`.
 - To direct all the output of the command and the standard error messages to another command, enter `{command 1} 2>&1 | {command 2}`.

Redirect the Standard Input

To redirect the standard input:

1. Log in as a user.
2. Redirect the standard input.
 - To send a file as the input to a command, enter `{command} < {file name}`.
 - To accept input from the keyboard until a specified string is provided as input, enter `{command} <<{string}`.

TOPIC E Use Control Statements in Shell Scripts

In the last topic, you redirected the standard input and output between commands and files. Now you may want to write a simple shell script to automate repetitive tasks. In this topic, you will use control statements in shell scripts.

Consider a scenario where you want to greet users by displaying either "Good Morning," "Good Afternoon," or "Good Evening," according to their login time. Using control statements, you can specify the time span for each message and display the relevant message depending on the time the user logs in.

Control Statements

A **control statement** is an instruction that determines the direction a program takes depending on a test condition. The direction can be different from the sequential order in which the instructions are listed. Control statements are associated with one or more action statements that will be executed only when a specified condition is satisfied.

Figure 9-22: Control statements are used to change control flow.

Expressions

Expressions are a group of characters that are generally used to specify conditions. They are formed by combining variables and constants with operators. They are used in the if and while statements. Performing arithmetic comparisons and string comparisons and testing files are the main functions of expressions. If an expression contains the <, >, &, or | symbol, parentheses are required.

Programming Constructs

Programming constructs are parts of a program that define the order in which the instructions in a program are executed. A programming construct is a sequence of statements that starts with a command, such as if, and ends with the corresponding terminal statement. Constructs may or may not return a value.

Figure 9-23: A sample programming construct used in system scripts.

Loops

A **loop** is a programming construct that supports repetitive execution of one or more statements. It is a block of code that repeats a list of commands as long as the condition controlling the loop is true.

Test Constructs

In a programming language, **test constructs** test for a condition and then act according to the result of the test. An if programming construct tests a list of commands to check whether their exit status is 0. If yes, one or more commands are executed. Bash also uses the test command and various bracket and parentheses operators as test constructs.

Functions

A **function** is a subprogram that executes an operation and returns a value on completion of the operation. Functions take variables, called arguments, as input. There are two types of functions: built-in functions and user-named functions. Functions can also execute other functions.

Figure 9-24: A sample function.

Variables

A **variable** is a named storage location in a program's memory that can be assigned a value.

Variables can hold different values at different times, but only one value at a time. User-defined variables, as suggested by the name, are defined by the user and are local to the current shell. System variables are those that are created and maintained by the operating system itself.

Comparison and Logical Operators

Some comparison, logical, and arithmetic operators are listed in the tables.

<i>If You Need To</i>	<i>Use This Operator</i>
Check if both values are equal.	==
Check whether the first value is greater than the second value.	>
Check whether the first value is less than the second value.	<
Check if both values are unequal.	!=
Check whether the first value is greater than or equal to the second value.	>=
Check whether the first value is less than or equal to the second value.	<=

<i>Logical Operator</i>	<i>Description</i>
&&	Boolean AND
	Boolean OR

<i>Logical Operator</i>	<i>Description</i>
^	Boolean XOR
!	Boolean NOT

<i>If You Need To</i>	<i>Use This Arithmetic Operator</i>
Increase the value by one.	++
Decrease the value by one.	--

The if Statement

The most frequently used construct is the if statement. An if statement contains a condition to be evaluated and one or more actions to be performed, if the condition is satisfied. If the condition is not satisfied, the actions are skipped and the next statement in the script is executed. The end of the set of instructions is indicated by the fi statement.

Syntax

The syntax for the if statement is as follows:

```
if ( {condition that needs to be satisfied} )then {commands to be executed}..fi
```

The if...else Statement

The if...else statement allows a choice between two actions based on the evaluation of a condition. If the condition is satisfied, the first action is performed; otherwise, the action following the else segment is performed. The end of the set of instructions is indicated by the fi statement.

If there are more than two sets of instructions, one or more elif statements may be used to specify alternative sequences of action.

Syntax

The syntax for the if...else statement is as follows:

```
if ( {condition that needs to be satisfied} ) then {commands to be executed} .. else {commands to be executed} .. fi
```

Looping Statements

Looping statements, also referred to as iterative statements, are a type of control statement that helps you execute a part of the script repeatedly based on a specific condition that is evaluated. The condition is tested based on the value of a variable. There are two types of loops supported by the Bash shell: the for loop and the while loop. In shell scripts, the commands to be iterated are enclosed within the do and done statements.

Figure 9-25: Loop statements are used to repeat a set of instructions.

The for Loop

The **for loop** executes a part of the script as many times as specified by a numerical variable that is within the conditional part of the statement. The for loop is unique because the conditional part of the statement contains the initial value of the variable, the test condition, and the increment or decrement of the variable value.

Syntax

The for loop in Linux lets you repeat a series of commands based on the evaluation of a condition.

For example, the syntax for a for loop is as follows:

```
for ((expr1; expr2; expr3)) do {commands to be executed} .. done
```

In this syntax, *expr1* is the initial statement executed once before the first loop, *expr2* is the test condition, and *expr3* is the increment or decrement expression that is executed at the end of each loop.

Alternate syntax: The for loop has another syntax:

```
for {variable name} in {list} do {commands to be executed} .. done
```

In this syntax, the variable is assigned a value from the list and the loop executes once for each value given in the list.

The while Loop

The **while loop** enables you to repeat a set of instructions for a fixed number of times, while a specific condition is met. The condition is left open ended in a while loop. The first expression is evaluated, and if the expression is true,

the actions in the loop are performed. The execution returns to the beginning of the loop and the expression is evaluated again. If the expression is false, the execution passes to the next statement.

Syntax

The syntax for a while loop is as follows:

```
while { condition that needs to be satisfied } do {commands to be executed} .. done
```

The until Loop

The until loop is similar to the while loop, except that the code is executed when the control expression is false. For example, the syntax for an until loop is as follows:

```
until ( {condition that needs to be satisfied} ) do {commands to be executed} .. done
```

How to Use Control Statements

Follow these general procedures to use control statements.

Use Control Statements in Scripting

To use control statements in scripting:

1. Log in as a user.
2. To write a Bash shell script, on a terminal, enter `vi {script file name}`.
3. To switch to insert mode, press **I**.
4. Type `#!/bin/bash` to specify the shell.
5. Enter the commands you want to execute.
6. Use control statements or loops, as necessary.
 - To use the if statement, type `if ({condition that needs to be satisfied}) then {commands to be executed} fi`.
 - To use the if...else statement, type `if ({condition that needs to be satisfied}) then {commands to be executed} else {commands to be executed} fi`.
 - To use the for loop, type `for (({expr1}; {expr2}; {expr3})) do {commands to be executed} .. done`.
 - To use the while loop, type `while [{condition that needs to be satisfied}] do {commands to be executed} done`.
7. To return to command mode, press **Esc**.
8. To save the file and exit the text editor, enter `:wq`.
9. To convert the file to an executable script, enter `chmod a+x {script file name}`.

Email the Superuser Based on Command Return Values

To conditionally email the superuser based on command return values:

1. Log in as a user.

2. To email the superuser, type the script.
 - a. Enter `vi {script file name}`.
 - b. Type `if ((conditions))` then and press **Enter**.
 - c. To define a value to a variable, enter `{variable}={value}`.
 - d. Enter `fi`.
 - e. Enter `if [${variable name} = {value}]` then.
 - f. Enter `echo "Type the body of the mail"`.
 - g. To send an email to the root user with the specified subject, enter `mail -s "{subject of the mail}" root`.
 - h. Enter `fi`.
 - i. Save and close the file.
3. To convert the file to an executable script, enter `chmod a+x {script file name}`.
4. To execute the script, enter `./{script file name}`. If the given condition is satisfied, the message, "Type the body of the mail," is displayed.
5. Type the body of the message and, on the last line, enter a period to terminate the email.
6. If necessary, enter the Cc: address.
7. Verify that the specified superuser has received the email message.
 - a. Log in as the superuser.
 - b. To open the mailbox, enter `mail`
 - c. To read the mail's contents, type the mail number. You can identify a new email message by the `>N` symbol at the beginning.

Write a Bash Function for Frequently Used Sequence of Commands

To write a bash function for frequently used sequence of commands:

1. Log in to the CLI as **root**.
2. Write a bash function for a sequence of commands.
 - a. Enter `vi /{bash function name}`.
 - b. Enter `#!/bin/bash`.
 - c. Enter a sequence of frequently used commands using the desired programming constructs.
For example, write a program for the for loop to automate a sequence of commands.
 - d. Save and close the file.
3. To convert the file to an executable script, enter `chmod a+x {script file name}`.
4. To execute the sequence of commands, enter `./{bash function name}`.

Working with the Bash Shell and Shell Scripts Review

Scenario

Answer the following review questions.

1. When might you use variables in your own shell scripts?
2. What are the various tasks that you might perform in your environment by running a shell script?

Summary

In this lesson, you customized the Bash shell and performed various operations in it. You also worked with shell scripts, redirected input and output in shells, and used control statements in scripts to automate repetitive tasks. This will enable you to efficiently perform your job as a system administrator.

10 Managing Jobs and Processes

Lesson Time: 1 hour, 30 minutes

Lesson Introduction

In the last lesson, you worked with the Bourne-Again SHell (Bash shell) and shell scripts in the Red Hat® Enterprise Linux® environment to help successfully manage a system. In your routine management of a Linux system, there might also be instances when various utilities need to be run simultaneously on a system, not necessarily via shell scripts. In this lesson, you will manage jobs and processes.

The multitasking capability of Linux enables you to perform many tasks at the same time, such as compiling a program; sorting a database; and creating a document. When system resources are utilized simultaneously, system performance reduces. A system administrator should be able to manage system resources effectively by allocating the right amount of resources to every task run by users.

Lesson Objectives

In this lesson, you will manage jobs and processes. You will:

- Manage jobs and background processes.
- Manage processes using the process table.
- Work with delayed and detached jobs.
- Schedule jobs.
- Maintain the system time.

TOPIC A Manage Jobs and Background Processes

In the last lesson, you set up Linux variables and executed shell scripts to assist you in daily management of your Linux system. There are, however, other daily tasks that you need to manage in order to run and oversee a Linux system, particularly one that handles multiple users. For example, you may need to manage many processes that can run simultaneously in a Linux environment. In this topic, you will manage multiple jobs and background processes.

Most systems can handle one user running multiple processes, but what happens when hundreds of users run applications simultaneously? As an administrator in such an environment, you need tools and options that allow you to manage system resources efficiently. Using Linux's multitasking capabilities, you can manage jobs and processes in the background.

Processes

A **process** is an instance of a running program that performs a data processing task. A process consists of a sequence of steps stored on a system; these steps convert input data to output data.

Processes can be subdivided into threads. Every process is assigned a unique Process ID (PID) and includes time limits, shared memory, or child processes. Processes may run in the foreground or background of the system.

Process ID

Processes

PID	TTY	STAT	TIME	COMMAND
1	?	Ss	0:00	init [5]
2	?	S<	0:00	[migration/0]
3	?	SN	0:00	[ksoftirqd/0]
4	?	S<	0:00	[watchdog/0]
5	?	S<	0:00	[events/0]
6	?	S<	0:00	[khelper]
7	?	S<	0:00	[kthread]
10	?	S<	0:00	[kblockd/0]
11	?	S<	0:00	[kacpid]
67	?	S<	0:00	[cqueue/0]
70	?	S<	0:00	[khubd]
72	?	S<	0:00	[kseriod]
134	?	S	0:00	[pdfflush]
135	?	S	0:00	[pdfflush]
136	?	S<	0:00	[kswapd0]
137	?	S<	0:00	[aio/0]
290	?	S<	0:00	[kpsmoused]
320	?	S<	0:00	[mpt_poll_0]
321	?	S<	0:00	[scsi_eh_0]
324	?	S<	0:00	[ata/0]
325	?	S<	0:00	[ata_aux]
330	?	S<	0:00	[kstripped]
339	?	S<	0:00	[ksnapd]
350	?	S<	0:00	[kjournald]

--More--

Figure 10-1: PIDs of processes running on a system.

The Process ID

Whenever a process is started, the system allocates a unique PID to identify the process. Also, every process inherits the User ID (UID) and Group ID (GID) of the user who starts the process. This is similar to the ownership of files and directories on the Linux filesystem.

The init Process

The first process, called init in Linux, is started by the kernel at boot time and never terminates.

The PID of the init process is always 1. In modern Linux systems, the init is often systemd.

Foreground Processes

A **foreground process** is a program with which a user interacts at a particular time. Only one foreground process can be run at a time. As the user switches between programs, different programs become the foreground process at different times. A foreground process is initiated by entering a command at the prompt or by clicking a shortcut in the Graphical User Interface (GUI).

Background Processes

A **background process** is a program that allows the Linux shell to execute a command that runs a job in the background, enabling processes to run simultaneously. While the user is interacting with the foreground process, a number of programs can run as background processes. The shell does not have to wait for one process to end before it can run more. A process can be run in the background by suffixing the invoking command with an ampersand (&) separated by a space.

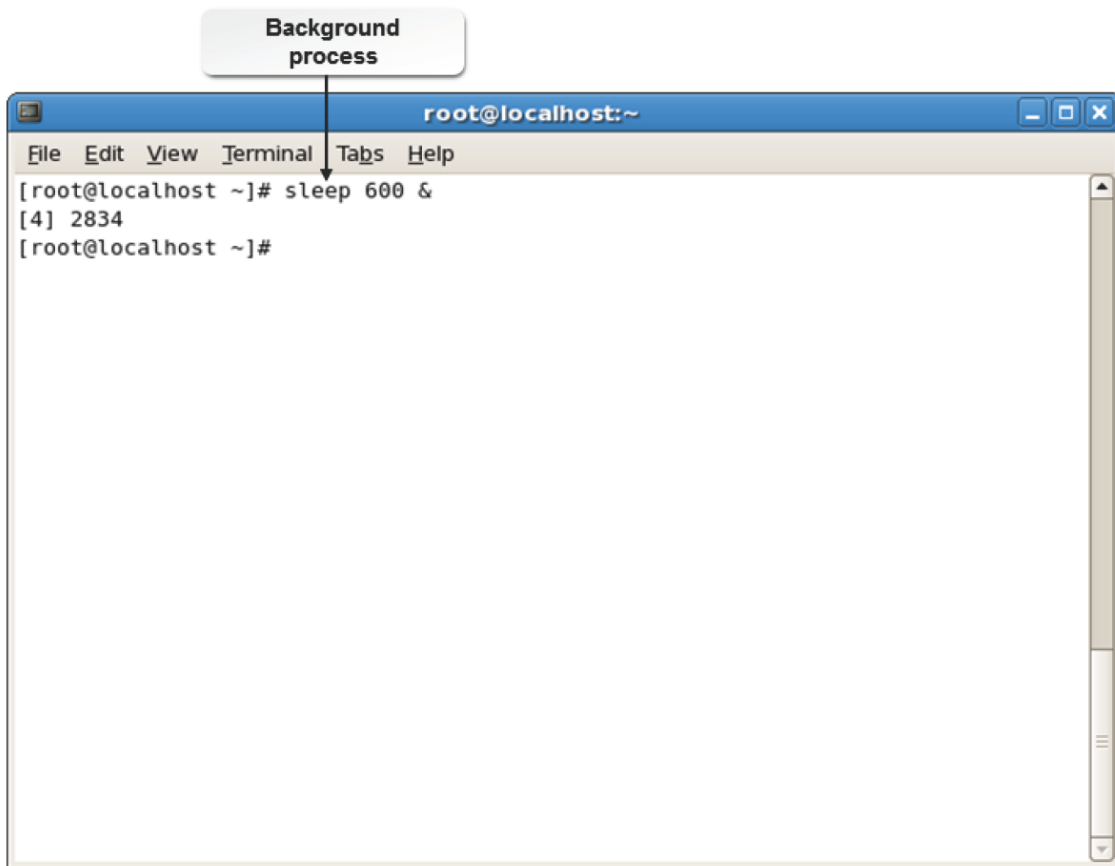


Figure 10-2: A process initiated to run in the background.

Daemons

Daemons always run as background processes that never require user input. Other processes remain in the background temporarily, while the user is busy with the current foreground process.

The Program and Process Relationship

A **program** is a set of instructions describing how to carry out a task. A command that resides on your system is a program. When you enter a command at the prompt, a set of instructions perform a task.

A process is a program that executes instructions. The operating system creates a process to carry out that task. Processes have unique identities and exist until their tasks are completed. When the task is completed, the process is terminated. Each program running on a system is assigned a PID.

Multitasking

Multitasking is a method of allowing the operating system to run concurrent programs simultaneously without degrading system performance. Multitasking enables several programs to share the same system resources. Processes spawned by multitasking are all active at the same time.

They are not in a sequence or a suspended state waiting to be run. Processes placed in a multitasking state remain active until completed, unless terminated or suspended by the user. One or more users may run multiple tasks on a system.

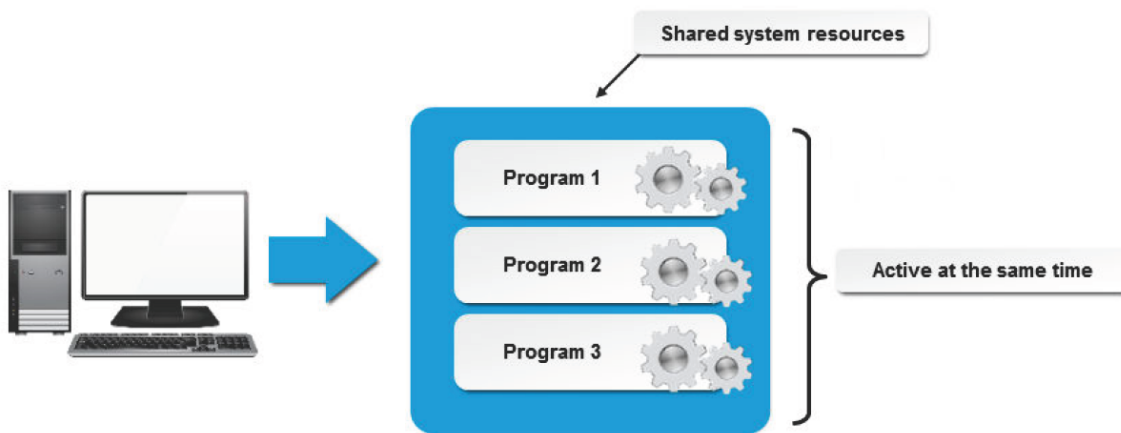


Figure 10-3: Multitasking on a Linux system.

The Jobs Table

The **jobs table**, invoked by the jobs command, is a table containing information about jobs running in the background. It contains entries only for those jobs that are running in the current shell. The jobs table contains a numeric label for each job indicating the order in which the jobs were started. In addition, the jobs table includes a plus sign (+) to designate the current or the most recently started job and a minus sign (-) to designate the job that was started just prior to the most recent job. It also includes the status and name of each job.

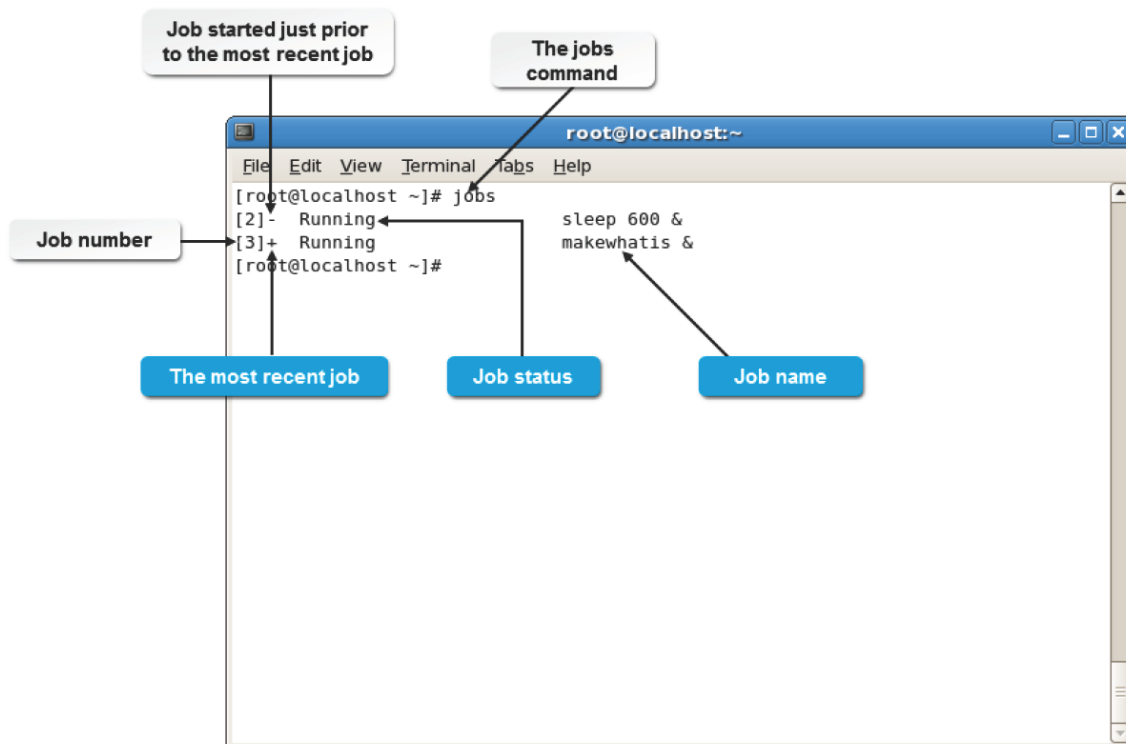




Figure 10-4: The jobs table listing jobs running in the current shell.

	Note: The job name listed in the jobs table is actually the command that initiated the job.
	Note: The plus (+) and minus (-) signs indicate only the order in which jobs are started. All jobs, however, are actually run simultaneously.

Job Status

There are four possibilities for the status of a job.

Status	Description
Running	An active job.
Stopped	A job that is suspended.
Terminated	A job that is killed.
Done	A completed job.

Jobs in a New Shell

Any job that a user placed in the background will appear in that user's jobs table, but other users' jobs will not appear. If you were to start a new shell, the jobs table for the new shell will be empty.

However, the jobs started in the previous shell will continue to run.

Suspend vs. Terminate a Process

The **Ctrl+Z** key combination suspends a job, while the **Ctrl+C** key combination terminates or kills a job. If you display the jobs table after you press **Ctrl+Z** to suspend a job, you will see that the current job is in a suspended state (labeled in the jobs table as "Stopped"). Although the jobs table lists jobs running in the background, a foreground job that gets suspended appears in the jobs table to remind the user that there is a suspended job waiting to be restarted or terminated. Refer to the following table for a summary of job control commands.

Action	Foreground	Background
Suspend a job	Ctrl+Z	Bring to foreground, then press Ctrl+Z
Terminate a job	Ctrl+C	kill % <i>#</i>

Restarting a Suspended Job

The `bg` command, with the syntax `bg {%#}`, can be used to restart a specified background job that has been suspended. You can specify the number of the suspended job you want to restart after the percent sign. If there is only one job running in the background, then you do not have to specify the number. You can type `bg %` to restart it.

Bringing a Job to the Foreground

If you need to bring a job from the background to the foreground, use the `fg` command, with the syntax `fg {%#}`. You do not have to enter a number after the percent sign if there is only one job running in the background.

Job Control Tools

Job control tools enable you to manipulate the jobs appearing in the jobs table.

Tool	Enables You To
<code>jobs</code>	View the status of the jobs running in the background.
Ctrl+Z	Halt a running process temporarily.
<code>fg {<i>%job number</i>}</code>	Bring the specified process to the foreground.
<code>bg {<i>%job number</i>}</code>	Send the specified process to the background.
<code>kill {<i>%job number</i>}</code>	Terminate the specified process.

How to Manage Jobs and Background Processes

Follow these general procedures to manage jobs and background processes.

Manage Jobs

To manage jobs:

1. Type the command with an ampersand (&) after it to put a job in the background.
2. If necessary, execute additional commands.
 - Execute another command in the background using the ampersand.
 - Execute another command without putting it in the background.
3. To see the list of processes that are running in the background, enter jobs at the command line.
4. Manage the jobs that are running in the background.
 - To kill a process in the background, enter kill %[job number].
 - To switch a background process to the foreground, enter fg %[job number].
 - To suspend a foreground job, press **Ctrl+Z**.
 - To restart a suspended job, enter bg %[job number].

TOPIC B Manage Processes Using the Process Table

In the previous topic, you managed multiple processes using the jobs table. While the jobs table is unique to each user's specific shell, the process table is for the entire system. In this topic, you will manage processes using the process table.

Monitoring system processes enables system administrators to track the usage of system resources.

Tracking the processes running on a system helps you manage your resource allocation better. As a Linux administrator, you will find the process table useful because it contains entries for all the processes that are started by all the users on the system. With the process table, you can manage processes on the system in a Linux environment.

The Process Table

The **process table** is a record that summarizes the current running processes on a system. It enables the administrator to keep track of all processes run by different users. Some of the details displayed in the process table include the PID, the size of the program in memory, the name of the user who owns the process, and time.

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	2344	0.0	0.0	1656	460	tty2	Ss+	Jun29 0:00	0:00	/sbin/mingetty
root	2354	0.0	0.0	1656	432	tty3	Ss+	Jun29 0:00	0:00	/sbin/mingetty
root	2357	0.0	0.0	1656	460	tty4	Ss+	Jun29 0:00	0:00	/sbin/mingetty
root	2360	0.0	0.0	1656	432	tty5	Ss+	Jun29 0:00	0:00	/sbin/mingetty
root	2361	0.0	0.0	1656	432	tty6	Ss+	Jun29 0:00	0:00	/sbin/mingetty
root	4824	0.0	0.0	1656	432	tty1	Ss+	06:45 0:00	0:00	/sbin/mingetty
root	5252	0.4	0.9	16748	10064	tty7	Ss+	10:31 0:21	0:21	/usr/bin/Xorg :
root	5513	0.0	0.1	4920	1820	pts/1	Ss	10:49 0:00	0:00	bash
root	5654	0.0	0.0	4252	940	pts/1	R+	11:45 0:00	0:00	ps u

Figure 10-5: The process table listing the processes running on the system.

The Process Table vs. the Jobs Table

The process table has options that are different from the jobs table. The process table can display all processes running on the system irrespective of which user started it, including system processes started automatically at boot time. However, the jobs table shows only the processes started in a user's current shell. Also, the unique PIDs of processes are displayed in the process table, while the jobs table shows only their job number according to the order in which they were started. In the jobs table, only the original process is displayed as an entry, but in the process table, the original process and all subsequent processes that were started are displayed. So, a single entry in the jobs table may have more than one corresponding entry in the process table. Certain job control commands can be applied only by referring to processes by their job number.

The ps Command

The ps command invokes the process table. When the command is run without any option, it displays the processes run by the current shell with details such as the PID, the terminal associated with the process, the accumulated CPU time, and the command that started the process. However, different options may be used along with the command to filter the displayed fields or processes.

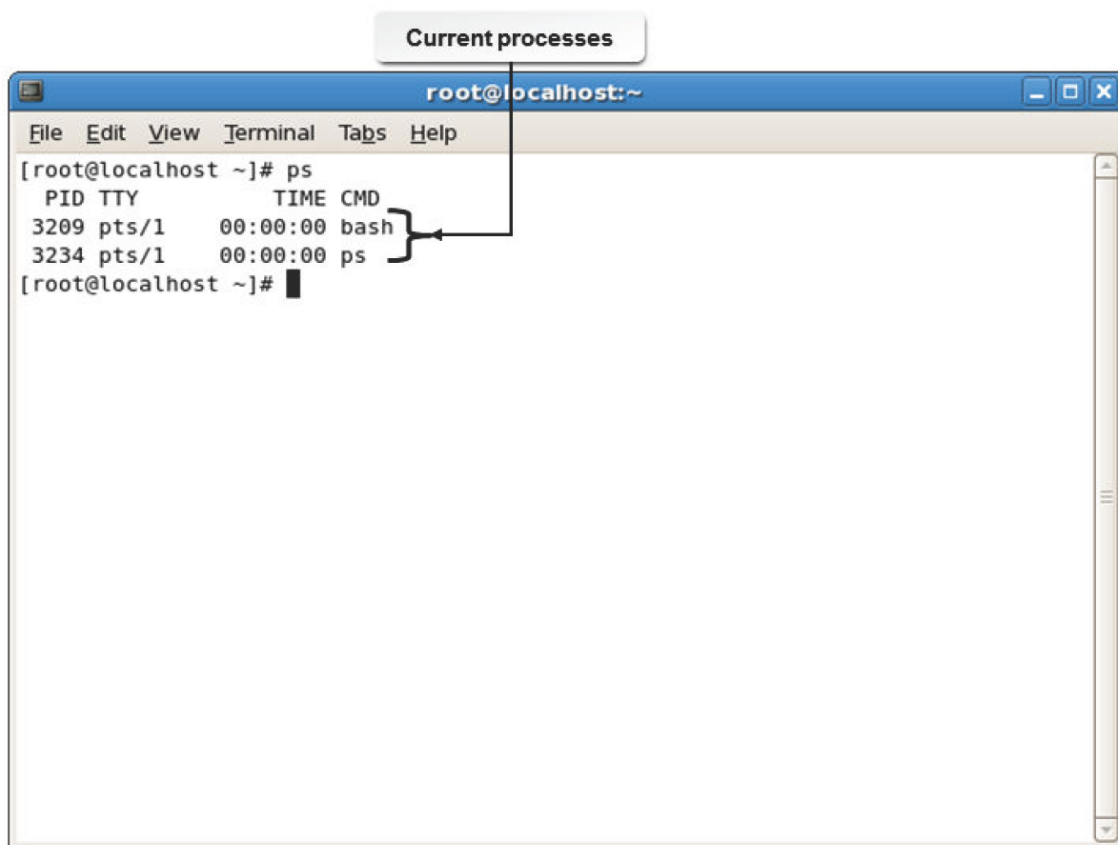


Figure 10-6: The `ps` command displaying the processes run by the current shell.

Syntax

The syntax of the `ps` command is `ps [options]`.

ps Command Options

The `ps` command supports several options. Some of the important options are listed here.

Option	Description
a	Lists all user-triggered processes.
-e	Lists all processes.
-l	Lists processes using a long listing format.
u	Lists processes along with the user name and start time.
r	Excludes processes that are not running currently.
x	Includes processes without a terminal.
T	Excludes processes that were started by any terminal other than the current one.



Note: Unlike many commands in Linux, the `ps` command supports options with and without a hyphen before them. However, the function of the same options with or without a hyphen may differ greatly.

Command Options for Selective Display

Some common `ps` command options can be used to select a specific set of processes.

Option	Used To
-U {user name}	Display the processes based on the specified user.

Option	Used To
-p {PID}	Display only the specified process associated with the PID.
-C {command}	Display all processes by command name.
--tty {terminal number}	Display all processes running on the specified terminal.

Fields Displayed by the ps Command

Various options display different fields. Several fields can be displayed using the ps command.

Field	Description
PRI	Process scheduling priority. Processes with low priority have higher numbers.
NI	Process nice value. Processes using less CPU time have higher numbers.
SIZE	Virtual image size.
RSS	Physical memory in KB.
WCHAN	Kernel function in which the process resides.
STAT	Status. Values include R (running), T (stopped), D (asleep and uninterruptible), S (asleep), Z (zombie), and N (positive nice value).
TT	The TTY or terminal associated with the process.
PAGEIN	The number of major page faults.
TRS	Resident text size.
SWAP	Number of KB of swap used.
SHARE	Amount of shared memory.

Child Processes

A process created by a running process is called a **child process**. The process table contains both **parent processes** and child processes. There may be several levels of processes. The parent process can spawn a child process, the child process can spawn another child process, and so on. The parent process must be running for the child processes to run. Parent processes are assigned a unique **Parent Process ID (PPID)**.

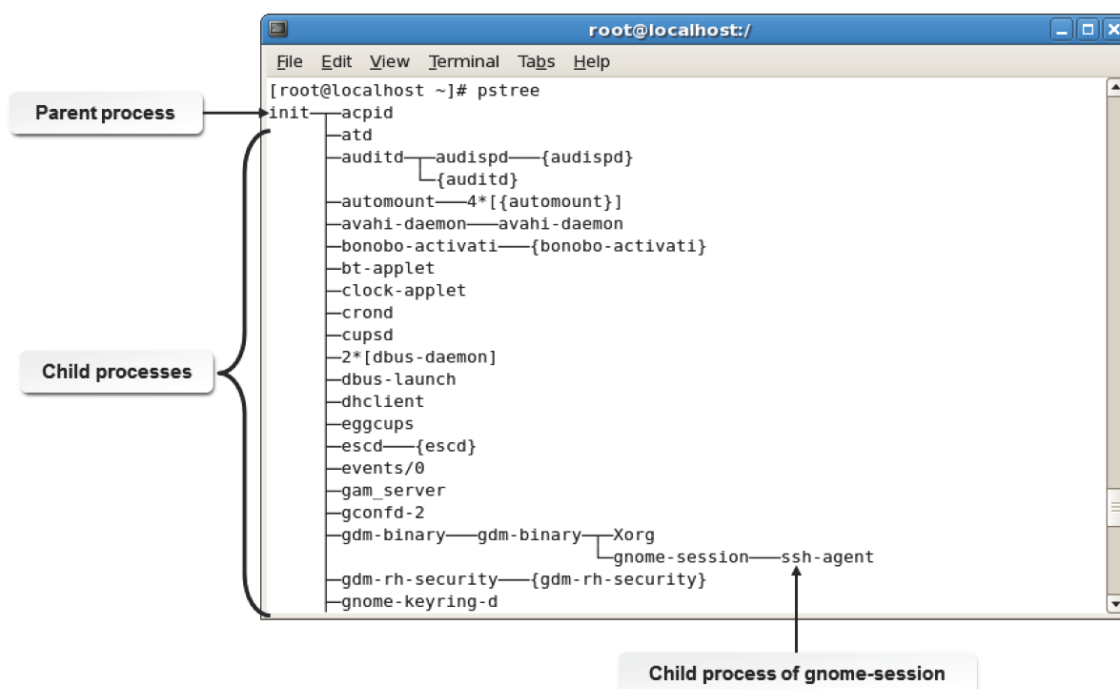


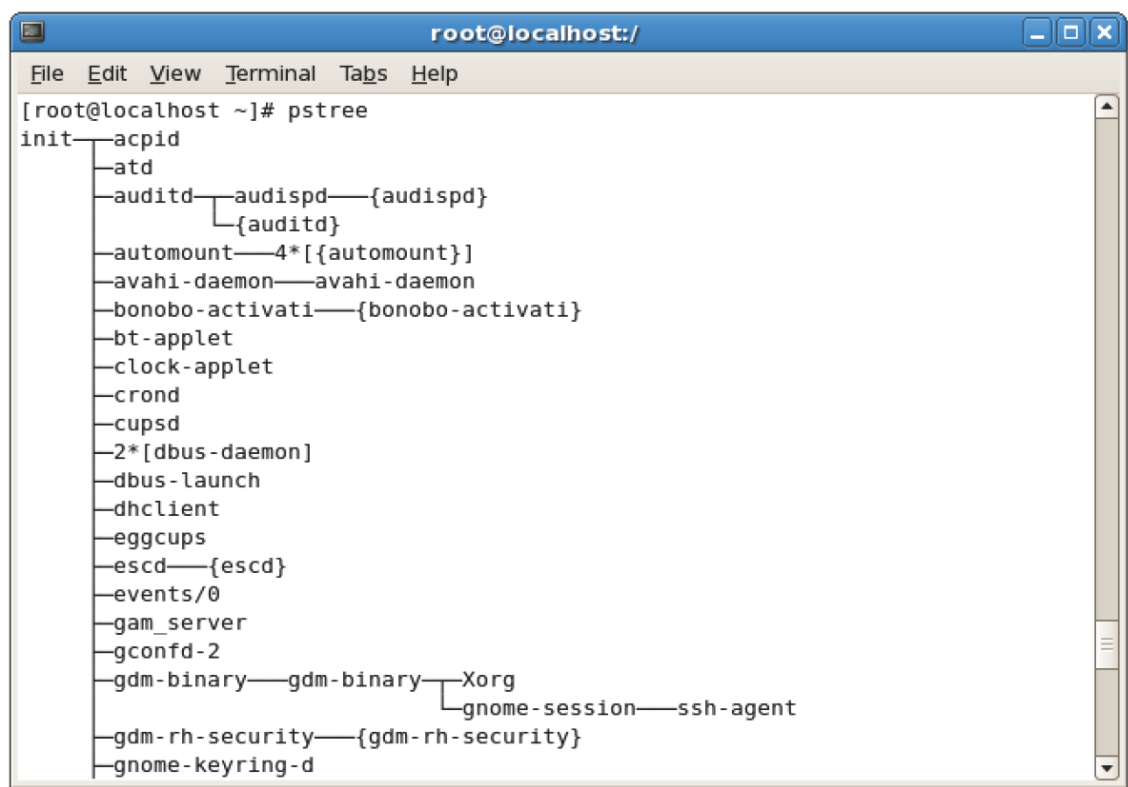
Figure 10-7: The process tree showing parent and child processes.

Identifying Child Processes

Identifying child processes is not an easy task, especially if there are multiple processes and child processes running at the same time. By examining the order of the PIDs, you may be able to determine the order in which the processes were created and infer which processes are related.

The pstree Command

The pstree command enables you to list the processes running on a Linux system in a tree-like format. This helps you track parent and child processes. All processes are listed as child processes to init and this is represented by the initial branching. The processes started within a shell will branch out of the shell's parent process.



pstree lists processes in a tree format.

Figure 10-8: The pstree command displaying parent and child processes.

Process Identification Commands

Process identification commands enable you to extract information about a process using its name or some other attribute associated with it.

Command	Description
pidof	Displays the PID of the process whose name is specified and can be used only when the name of the process is known. However, it is recommended that a full path name of the process be given because more than one process could run with the same name. The syntax of this command is pidof [options] {string}.
pgrep	Displays the PID of processes that match any given criteria such as the name or UID of the user who invoked it, the start time, the parent PID, and so on. The syntax of this command is pgrep [options]{process name}.

pidof Command Options

The pidof command supports only two options.

Option	Used To
-s	Instruct the program to display only one PID.
-c	Instruct the program to display the PIDs that are running from the same root directory.

pgrep Command Options

The pgrep command supports different options by which one or more conditions for search may be specified.

Option	Used To
-f	Specify the full path name of the process.
-l	Print the name of the process along with its PID.
-u {userid}	Specify the UID of the user who started it.
-G {groupid}	Specify the GID related to the process.
-n	Specify the most recent process.
-o	Specify the oldest process.

Signals

Signals are messages sent to a process to perform certain actions. They are used to suspend or terminate processes. Signals may affect only the process specified and its child processes. Signals may be executed, caught, blocked, or ignored by processes.

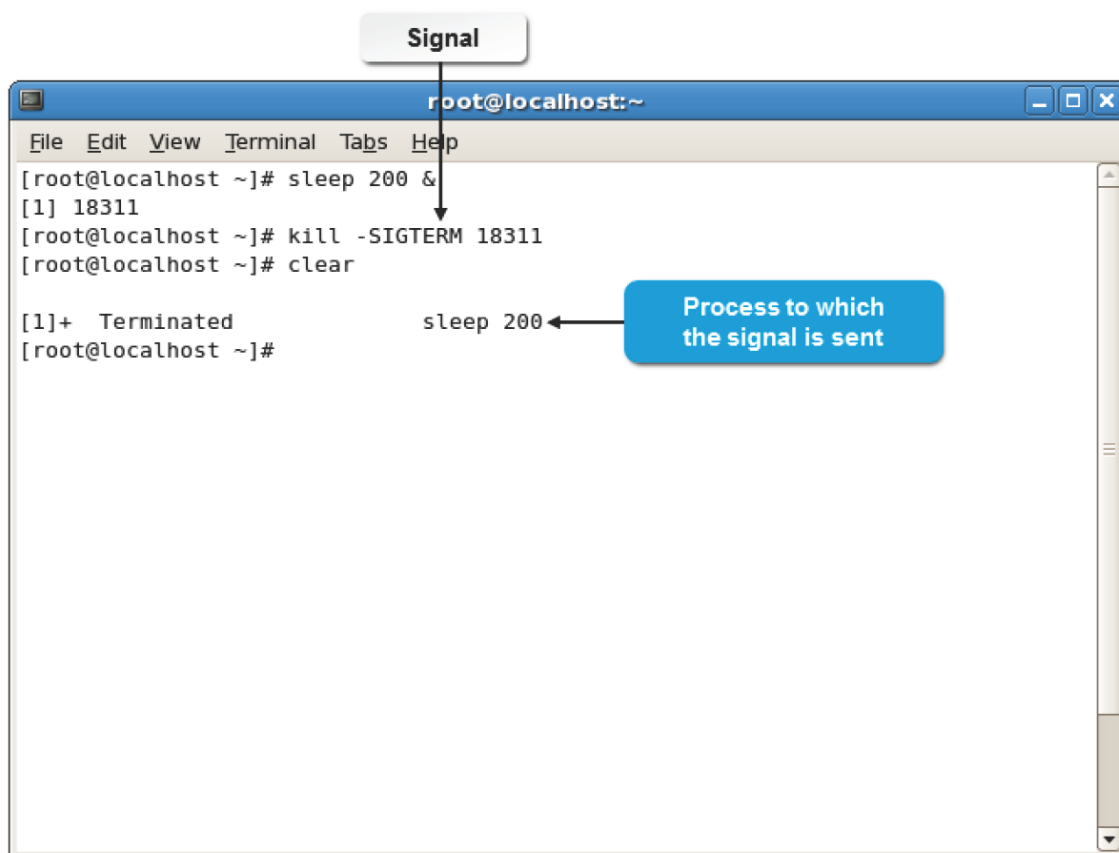



Figure 10-9: A signal sent to a process.

kill Commands

Different commands are used to send signals to processes to end or "kill" them.


Command	Description
kill	Sends any specified signal, or by default the termination signal, to one or more processes. The PID must be specified as the argument. The syntax of this command is kill <i>[options]</i> {PID}.
pkill	Signals processes based on the name and other identifiers as in the pgrep command. The syntax of this command is pkill <i>[options]</i> {command}.
killall	Kills all processes by the name specified. The syntax of this command is killall <i>[options]</i> {command}.

	Note: The kill command accepts either the PID or the job number as an argument. So, this command can also be used as a job control tool.
---	---

Kill Signal Options

You can either use the kill signal option or its corresponding numerical value to send a signal to terminate a process. The following table lists the most frequently used kill signal options and their description.

Option	Used To
SIGKILL or 9	Send the kill signal to a process.
SIGTERM or 15	Send the termination signal to a process.
SIGSTOP or 19	Stop a process.

	Note: Sometimes, even after closing an X session, some of the X applications may not get terminated properly. In such cases, you need to use the ps command to identify the PID of that application and then kill the process.
---	---

Using the PID Number to Terminate Processes

You can use the kill command with the process table to end processes. By entering kill followed by the PID, you can terminate specific processes.

When you use the kill command with the jobs table, you are working only with the jobs that you started. However, the process table may display processes that do not belong to you. As a user, you can use the kill command only with processes that you own. As root, you can kill anyone's processes.

There are many options available with the kill command. These options are referred to as kill signals. Some processes cannot be eliminated by the kill command. To terminate these processes, use the kill command with the -9 signal. This terminates the processes immediately.

Process States

A process state enables you to identify the current stage of a process. It is indicated by a single letter notation in the process table.

The various process states are given in the following table.

State	Description
Uninterruptible sleep (D)	The process is permanently inactive.
Running (R)	The process may be running or ready to be run.
Interruptible sleep (S)	The process is waiting to be run after some specific trigger.

State	Description
Stopped (T)	The process may be temporarily stopped by a job control tool or because it is being traced.
Dead (X)	The process has been killed. This state is never displayed.
Defunct (Z)	The process has ended, but only after its parent process. This implies that it has not been killed properly and it will remain as a "zombie."

The top Command

The top command lists all tasks running on a Linux system. It acts as a process management tool by allowing users to prioritize, sort, or terminate processes interactively. It displays a dynamic process status, reflecting real-time changes. Different keystrokes within this tool execute process management actions.

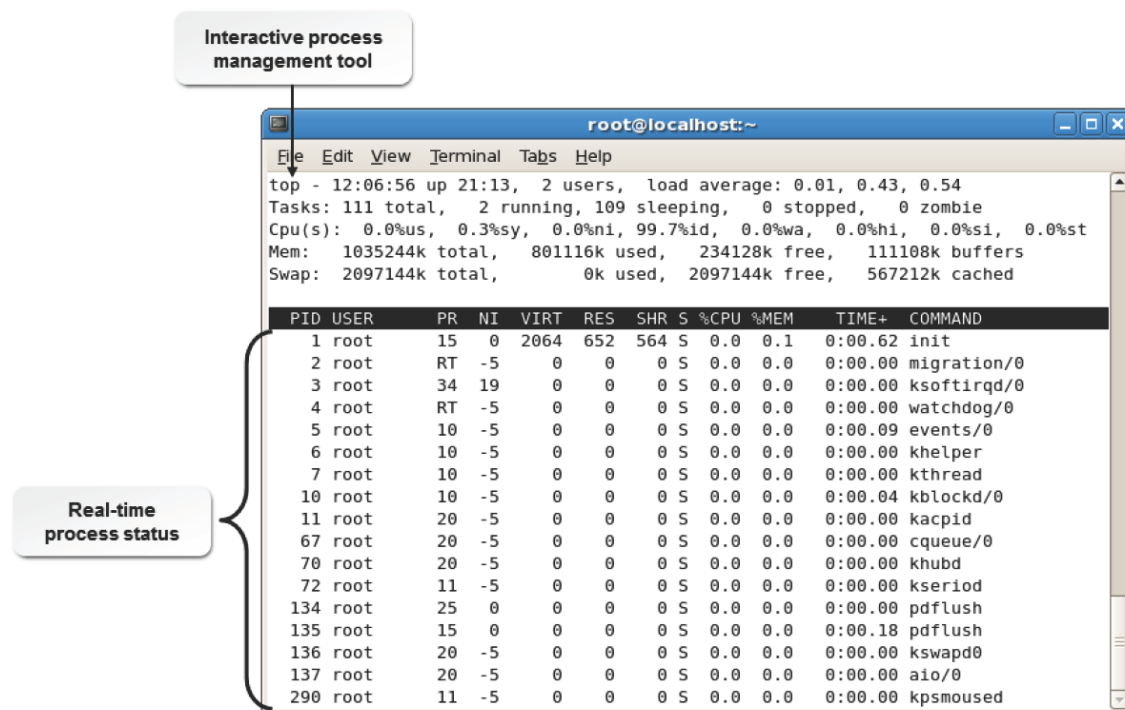


Figure 10-10: Managing processes using the top command.

Useful Keys to Manage Processes

The top command provides an interactive tool to manage processes by using some simple shortcuts. Some of the frequently used shortcuts are listed here.

Key	Function
Enter	Refreshes the status of all processes.
Shift+n	Sorts tasks in the decreasing order of their PID.
u	Displays processes belonging to the user specified at the prompt.
k	Terminates the process for which you specify the PID.
r	Renices the process for which you specify the PID.
h	Displays a help screen.
q	Exits the task list.

The nice Command

The *nice* command allows you to assign a priority level to a process. The nice value of a process indicates how "nice" the process is to others in sharing system resources. You can run a command at a priority higher or lower

than the command's normal priority. You must have the root user authority to run a command at a higher priority. The priority of a process is often called its *nice value*.



Figure 10-11: *The nice values of processes running on a system.*

Syntax

The syntax of the command is `nice -n {priority} {command}`, where the priority is specified by a number.

The nice Values of Processes

The niceness of a process may range from -20 to 19, where -20 indicates the highest priority and 19 the lowest. In the absence of an increment value, the nice command assumes an increment of 10 by default. Once lowered, the priority for any process cannot be increased by normal users, even if they own the process. By default, all processes in Linux have a nice value of zero.

The renice Command

The *renice* command enables you to alter the scheduling priority of a running process. When you renice a process group, it causes all processes in the process group to have their scheduling priority altered. When you renice a user, it alters the scheduling priority of all processes owned by the user.

By default, the processes affected are specified by their PIDs.

Figure 10-12: Changing the priority of a process using the renice command.

Syntax

The syntax of the renice command is: `renice {priority} {PID} [[-g] [groupid]] [[-u] [userid]]`.

How to Manage Processes Using the Process Table

Follow these general procedures to manage processes using the process table.

Manage Processes

To manage processes:

1. Log in as **root**.
2. Manage the processes on the system.
 - To view the processes running from the current table, enter `ps`.
 - To view all processes running on the system, enter `ps -e`.
 - To terminate a process, enter `kill [PID]`.
 - To terminate a process unconditionally, enter `kill -9 [PID]`.

Change the Priority of a Process

To change the priority of a process:

1. Log in as **root**.
2. Change the priority of a process as necessary.

- To start the process with the specified priority, at the command prompt, enter `nice -n {priority} {command}`.
- To change the priority of a running process with a specified priority, enter `renice {priority} [options]`.

Change the Priority of a Process Using the top Command

To change the priority of a process using the top command:

1. Log in as **root**.
2. To display processes sorted according to their CPU usage, enter `top`.
3. To alter the priority of a particular process, press **r**.
4. Enter the PID of the process for which you want to change the priority.
5. Enter the priority number.
6. To quit the display, press **q**.

TOPIC C Delay and Detach Jobs

You now have a basic understanding of managing jobs and processes. At times of high CPU usage, you may have to delay or stop some jobs in order to complete jobs of higher priority at a faster pace.

In this topic, you will delay and detach jobs.

Some jobs can make use of a lot of system resources, and you may want to run these jobs when the system is less busy—for instance, during evening hours. Linux allows you to delay and even detach jobs, facilitating efficient utilization of system resources.

Delayed and Detached Jobs

Delayed and detached jobs are job processes that enable users to put off the start of a job.

<i>Process</i>	<i>Definition</i>
Delay a job	A delayed job is one that can be run at some specified time after you issue the command. For example, a CPU-intensive job that can slow down the system is one that you may want to delay for off-peak work hours.
Detach a job	A detached job is a job that can be set to run after you log out of the system. For example, a task that will not be completed until after you leave can be set to continue running after you log out of the system.

To delay the start of a job, use the `sleep` command followed by the delay in seconds and the command name. The `sleep` command suspends any action upon the specified command for the specified number of seconds and then the command specified is executed. The delay can be up to 2,147,483,647 seconds. This is roughly 596,523 hours; 24,855 days; or 68 years so that the amount of time can easily be customized. You may also use the `at` command to run a command at a specified date and time.

The nohup Command

The `nohup` (no hangup) command tells a program to ignore the hangup signal that was sent while disconnecting. The **nohup.out** file stores the output of the `nohup` command, which will normally be displayed on the terminal.

Figure 10-13: Enabling a background process to run after logging out of the system using the `nohup` command.

If you have a task that cannot be completed until after you leave work, or if you have a task that is CPU-intensive and may slow the system, you can start the task before you leave and specify that it continues even after you log out of the system. You can do this by using the `nohup` (no hangup) command. The `nohup` command should run in the background so that it does not tie up your terminal. To enable a command to run in the background after you have logged out, use the syntax `nohup [command] &`.

The `screen` Command

The GNU `screen` command is a full-screen window manager that multiplexes a physical terminal between several processes, typically interactive shells. The `screen` command is another way that you can leave work running after you leave the system, which can then be resumed at a later point by reconnecting to your active `screen` session.

Figure 10-14: When screen is running, enter Ctrl-A and ? to view the screen help page.

If you have a task that cannot be completed until after you leave work, or if you have a task that is CPU-intensive and may slow the system, you can start the task before you leave and specify that it continues even after you log out of the system. You can do this by using the screen command. The screen command will continue to keep the interactive shell open and run your program in the background so that it does not tie up your terminal. When you next connect to your server, you can restore the active screen session via the screen -r command.

How to Delay or Detach Jobs

Follow these general procedures to delay or detach jobs.

Delay a Job

To delay a job:

1. Log in as **root**.
2. Determine the job that you want to schedule later. To schedule the job at a later time, enter `sleep {number of seconds to delay} [command to issue]`.

Detach a Job

To detach a job:

1. Log in as **root**.
2. Determine the job that you want to run after you log out of the system. To schedule the job to continue running even after you log out, enter `nohup {command to issue} &`.

Run a Job in a Screen session

To start a new screen session and run your job:

1. Log in as **root**.
2. To start a new screen session, enter `screen`.
3. Determine the job that you want to run after you log out of the system, and enter the command of that job.
4. To detach from the current screen session, press **Ctrl-AD**.
5. To return to your screen session to check on the progress of your job, enter `screen -r`.

TOPIC D Schedule Jobs

In the last topic, you delayed and detached jobs to overcome high CPU usage. You may also need to designate jobs that are to be executed at a specific time. In this topic, you will schedule jobs.

As a system administrator, you may need to schedule repetitive tasks to run at a specific time. For example, you may schedule a backup process every night so that it does not affect the work schedule of the users. Scheduling jobs is an important part of a system administrator's daily tasks.

Cron

Cron is a daemon that runs in the background on a Linux system and executes specified tasks at a designated time or date. Cron is normally used to schedule periodically executed tasks defined in the **crontab** file.

Figure 10-15: Cron executes specified tasks on the system.

Syntax

The syntax of the cron daemon is `cron [option] {mail command}`.

Significance of the /etc/cron Directories

Under the `/etc` directory, you will find directories such as **cron.d**, **cron.hourly**, **cron.daily**, **cron.weekly**, and **cron.monthly**. Depending on the frequency of the execution of bash script, you need to place your script file in the **cron.hourly**, **cron.daily**, **cron.weekly**, or **cron.monthly** directory. If you want to run a shell script for a frequency other than hourly, daily, weekly, or monthly, you need to place the script in the **cron.d** directory.

Cron Jobs

A task scheduled via cron is called a **cron job**. These jobs will run either at system level or at user level. The cron jobs that you create for users are stored in the **/var/spool/cron/[user name]** file.

System default cron jobs are stored in the **/etc/crontab** file. Only a root user can add system level jobs.

Figure 10-16: Cron jobs listed in the crontab file.

Scheduling a cron job is accomplished by adding the job to the system-wide **/etc/crontab** file. The **crontab** file may also contain environment variables that will be passed to the commands at the time of execution. Jobs in the **crontab** file are called entries, and they include a time description, the user name to run the command, and the command. The format of a crontab entry is: *{minute}*

{hour} {day of month} {month} {day of week} {user command}. The time fields in the

crontab entry are listed here.

<i>Field</i>	<i>Allowed Value</i>
Minute	0-59
Hour	0-23
Day of the month	1-31
Month	1-12 or Jan-Dec
Day of the week	0-7 (0 or 7 is Sunday) or Sun-Sat

In addition to specifying a particular time and day, a pattern can be described by using asterisks (*****) to specify all of a particular field. For example, an asterisk in the minute field indicates that the command should be carried out every minute. In addition to asterisks, time ranges are permitted by separating values with a dash (**-**) and lists of values are specified by separating values with a comma (**,**).

The tmpwatch Command

The **tmpwatch** command is run as a daily cron job that is used to delete files, such as the files in the **/tmp** directory, which have not been accessed for some time and are utilizing disk space. The tmpwatch command has the following options.

<i>Option</i>	<i>Enables You To</i>
---------------	-----------------------

<i>Option</i>	<i>Enables You To</i>
-u	Delete files according to the time they were accessed.
-m	Delete files according to the time they were modified.
-a	Remove all file types, including directories.
-d	Restrict the tmpwatch command from removing directories, even if they are empty or marked for deletion.
-f	Remove files forcefully, overriding all access regulations.

	Note: Even if one error is encountered during the cleanup process, the tmpwatch utility will exit.
--	---

Figure 10-17: Deleting files in the /tmp directory using the tmpwatch command.

Syntax

The syntax of the tmpwatch command is `tmpwatch [options] {hours}`.

The logrotate Command

The **logrotate** command is run as a daily cron job that is used to compress, delete, or mail log files.

It may be configured to run on a weekly or monthly basis depending on the log size. The logrotate command has the following options.

	Note: The configuration file for logrotate is <code>/etc/logrotate.conf</code> .
--	---

<i>Option</i>	<i>Enables You To</i>
-d	Turn on debug mode to disable any change from being made to the logs.
-f	Force log rotation by deleting old files irrespective of their importance and create new ones.
-m {subject} {recipient}	Mail the logs to the recipient. The default syntax is <code>/bin/mail -s</code> .

Figure 10-18: The `/etc/logrotate.conf` file allows you to control the rotation of logs.

Most Linux applications and commands store their log files in the `/var/log` directory. This is frequently where log files and their archives are managed, rotated, and archived.

The logwatch Utility

The `logwatch` utility is run as a daily cron job that is used to monitor logs. It is fully customizable via the `/etc/logwatch/conf/logwatch.conf` file. The utility searches logs and reports suspicious messages, and enables you to set detail levels for reports. 10, 5, and 0, correspond to high, medium, and low level details, respectively.

The logwatch utility has the following options.

Option	Enables You To
--detail {level}	Set the detail level of the log report.
--print	Print the report generated by the command.
--range {range}	Set the range for analysis. It can accept any value among Yesterday, Today, and All.
--mailto {address}	Mail the results to the recipient's mail ID.
save {file name}	Save the output to a file instead of displaying it.

System crontab Files

System **crontab** files are the configuration files for the cron utility. They are stored in the `/etc/ crontab` file. The name of the user running the command is indicated in the sixth field of the file.

When you create a **crontab** entry for a specific user, the sixth field contains the command that needs to be run at the specified time. System **crontab** files can be edited by the root user.

Figure 10-19: The /etc/crontab file with system-level cron jobs.

User crontab Files

In addition to system-level cron jobs, individual users can schedule cron jobs. Unlike the system-level **crontab**, users have their own **crontab** files. The format of entries in this file is the same as that of the system-level **crontab**, with the exception of the user field. Because the entire **crontab** file is dedicated to a single user, the user field is not included. While the **/etc/crontab** file can be edited directly, user **crontab** files are best edited via the crontab utility.

The at Command

The **at** command executes a given set of commands at a specified time. This command is useful for executing a set of commands only once. Using either the -f option or input redirection, the at command reads the list of commands from a file. This file needs to be an executable shell script.

The following table lists some frequently used at command options and their descriptions.

<i>Option</i>	<i>Enables You To</i>
atq	Display the job queue of all users except the superuser.
atq -V	Display the version number.
at -q [a-z]	Display the jobs in the specified queue.
at -m	Send mail to the user when the job is complete.
at -f {file name}	Read the job from the file rather than the standard input.
at -l	Print all the jobs queued for the user.
at -v	Display the time that the job will be executed before reading the job.

Figure 10-20: Executing a command at a specific time using the at command.

Syntax

The syntax of the at command is at *[options] {time}*.

Specifying Time Using the at Command

There are a number of common time formats. Some of the common time formats in which you can schedule a job are given in the following table.

<i>Time Format</i>	<i>Description</i>
HH:MM A.M. or HH:MM P.M.	Specifies the hour and minute.
MMDDYY or MM/DD/YY or DD.MM.YY	Specifies the day, month, and year.
JAN or FEB or MAR	Specifies the month.
SUN or MON or TUE	Specifies the day of the week.

Anacron

Anacron is a daemon that executes jobs at intervals, which are specified in days, without requiring the system to be running continuously. Anacron is used to control the execution of daily, weekly, or monthly jobs.

Figure 10-21: The anacron daemon executing scheduled jobs.

The **/etc/anacrontab** file is the configuration file for the anacron utility. This file has four fields.

The first field displays the number of days the job has not been run, the second field displays the time after which the job has to be run (after reboot), the third field displays the job identifier, and the fourth field displays the job to be run by the anacron utility.

How to Schedule Jobs

Follow these general procedures to schedule jobs.

Delegate Tasks Using the cron Command

To delegate tasks using the cron command:

1. Log in to the CLI as **root**.
2. To create a cron job for the root user, enter `crontab -e`.
3. To switch to input mode, press **I**.
4. To specify a schedule for the job, type *{minute} {hour} {day of month} {month} {day of week} {command that has to run}*.
5. To install the new cron job, save and close the file.
6. To verify that the new cron job has been executed, check if you have received an email message regarding the job that has been scheduled.
7. To list the cron jobs, enter `crontab -l`.
8. To remove the job from the queue, enter `crontab -r`.

Configure Access to cron and at Services

To configure user access to cron and at services, you need to perform the following actions in the corresponding files listed in the table.

<i>If You Need To</i>	<i>You Should</i>
Allow cron services to users	Add users to the /etc/cron.allow file.
Deny cron services to users	Add users to the /etc/cron.deny file.
Allow at service to users	Add users to the /etc/at.allow file.
Deny at service to users	Add users to the /etc/at.deny file.

Schedule Jobs to Run at a Specific Time

To schedule jobs to run at a specific time:

1. Log in to the CLI as **root**.
2. To specify an at job, enter at *{specific time format}*.
3. Enter *{job that has to be run}*.
4. To exit the process, press **Ctrl+D**.
5. To verify that the at job has been executed, check if you have received an email message for the job that has been executed.

Manage Jobs Using the at Command

To manage jobs using the at command:

1. Log in to the CLI as **root**.
2. To view the queue of pending at jobs, enter atq.
3. To delete a job from the queue, enter atrm *{job number}*.

TOPIC E Maintain the System Time

In the previous topic, you scheduled jobs to run at a specific time. It is not always enough to simply schedule a recurring job, however, as the system time might not be correct. You also need to monitor system clocks so that all systems show the same time. In this topic, you will maintain the system time.

Suppose you work in a company with clients in various cities around the world. You may need to synchronize your system time with that of your client's time zone to enable easier business transactions.

The Network Time Protocol (NTP)

Network Time Protocol (NTP) is a standard Internet protocol for synchronizing the internal system clock with the **true time** or the **average time** on a number of high accuracy clocks around the world. NTP is used for transmitting and receiving time on Transmission Control Protocol/ Internet Protocol (TCP/IP) networks. NTP is also used to set the clock of one computer to match that of another and synchronize it with the network clock.

Figure 10-22: Synchronization of system clocks with network time using NTP.

The pool.ntp.org Service

The **pool.ntp.org** is a collection of servers on the Internet that provides accurate time to the Linux systems using NTP.

Drift Files

A drift file is a file found in the **/etc/ntp** directory. The NTP drift file is used by the ntpd daemon to reset the time when the system is restarted. The drift file synchronizes the system clock and the clock drift to display the time from the NTP server.

The ntp.conf File

The **ntp.conf** file found in the **/etc** directory contains configuration options for the NTP server.

The file contains settings for all hosts on local and public servers. The ntpd daemon reads the **ntp.conf** file for synchronization settings and then connects to the NTP server.

UTC

Coordinated Universal Time (UTC) is a time scale that forms the official measure of time in the world. UTC is independent of time zones. It was previously referred to as **Greenwich Mean Time (GMT)**. It is the time at the prime meridian at Greenwich, England. Unlike GMT, leap seconds are included in UTC.

Figure 10-23: World time zone.

Leap Seconds

A **leap second** is the adjustment made to UTC, to account for the irregularity in the earth's rotation. The standard second is stable, while the motion of the earth is not. Therefore, occasionally, the standard minute is adjusted by adding a leap second. As a result, some minutes have 61 seconds.

Standard hours are always 60 minutes, though one of the minutes may be a second longer than usual. Standard days are always 24 hours.

Linux and Time Zones

In Linux, you can use the `tzselect` command to access a menu driven utility that will allow you to select the time zone for your system according to your geographic location. You need to define an environment variable, **TZ**, in the **/etc/profile** file to set the time zone for your system.

Locale Settings

Some settings of the system vary according to the geographic location of the system. These settings are known as locale settings. Some of the common locale settings are listed in the following table.

<i>Setting</i>	<i>Description</i>
Language	The language of the system must be adjusted according to the language spoken in the country or based on the user's choice. In Linux, you can use the <code>system-config-language</code> command to set the default language of the system.
Keyboard Layout	The keyboard layout of the system must be set according to the layout meant for inputting the language of the system. You can use the <code>system-config-keyboard</code> command to select the keyboard layout.

Setting	Description
Character Set	<p>There are different character sets or encoding methods that are available for displaying different languages. The main character sets are:</p> <ul style="list-style-type: none"> • Unicode • ISO-8859: An industry standard for 8-bit character encoding. • ASCII (Acronym for American Standard Code for Information Interchange): A standard used for specifying numbers to represent the alphabet in both upper and lowercase. • UTF-8 (Unicode Transformation Format): An encoding scheme for Unicode that allows you to handle 8-bit variable length characters. <p>You need to use the iconv program to convert from one encoding format to another.</p>
Environment Variables	<p>Some common environment variables associated with locale settings are:</p> <ul style="list-style-type: none"> • LANG: Helps in determining the local language to be used for displaying on the system and for use in system based messages. • LC_ALL: Helps in setting values for all locale settings. • LC_*: Helps in setting values specific to each locale settings. <p>This covers the environment variables:</p> <ul style="list-style-type: none"> • LC_COLLATE: Helps in specifying the collation of characters. Used with regular expressions and for sorting purposes. • LC_CTYPE: Helps in specifying the locale for converting casing of characters. • LC_MESSAGES: Helps in specifying the language to be used for displaying system error messages. • LC_MONETARY: Helps in specifying the currency formats for the system. • LC_NUMERIC: Helps in specifying the number formats for the system. • LC_TIME: Helps in specifying the date and time formats for the system.

The /usr/bin/locale File

The **/usr/bin/locale** file is an executable program that prints the current value of the locale settings on the standard output device. It displays the current value set for the locale variables such as LANG, LC_CTYPE, LC_NUMERIC, LC_TIME, LC_COLLATE, LC_MONETARY, LC_MESSAGES, and LC_ALL.

Clock Drift

Clock drift is the gradual variation in time that sets between the hardware clock and the system clock. The hardware clock is also known as the **Real Time Clock (RTC)**. It keeps track of the time when the system is turned off and not when the system is on. The system clock, however, functions only when the system is running and needs to be initialized at boot time. The hardware and system clocks will drift at different rates, apart from each other and also away from the real time. To synchronize both clocks, their drift rates need to be measured and corrected.

System Time

System time is the time maintained by a computer's internal clock. It is coordinated universal time with a resolution in milliseconds. The internal clock circuitry is backed up by a battery that keeps the clock running even when the computer is switched off. System time is used to date-stamp files with the time of their creation or revision. It can also be changed with difference in time zones.

Figure 10-24: System time configuration via the system-config-date Command.

The system-config-date Command

The system-config-date command allows you to open the **Date/Time Properties** dialog box that facilitates changing the system date and time and configuring the time zone.

The Date/Time Format

The International Organization for Standardization (ISO) specifies numeric representation of date and time. The standard format for date is YYYY-MM-DD, where YYYY represents the year in the Gregorian calendar, MM represents the month in the year, and DD represents the day in the month.

The American format of date is MM-DD-YYYY. However, Europeans write the day before the month. The separators used with numbers also vary among countries. The common format for time is hh-mm-ss, where hh represents hours, mm represents minutes, and ss represents seconds.

The /etc/timezone File

The **/etc/timezone** file is available with the Debian® and Ubuntu® distributions of Linux and is used to store the time zone information of the system. This file typically consists of a single line entry based on the continent/time zone format, such as America/New_York.

The /usr/share/zoneinfo/ Directory

The **/usr/share/zoneinfo/** directory contains time zone details relating to different countries.

When you export a time zone, details of that time zone are obtained from this directory.

The `/etc/localtime` Directory

The current time details of the system are stored in the `/etc/localtime` directory. If you make any change to your system time, the `/etc/localtime` directory gets updated.

The `tzconfig` Utility

The `tzconfig` utility allows you to set the time zone for Debian and Ubuntu systems. When you run this utility it will update the time zone in the `/etc/timezone` file and the files in the `/etc/localtime` directory.

`hwclock` Command Options

The `hwclock` command is used to access the hardware clock. The `hwclock` command consists of the following options.

<i>If You Need To</i>	<i>Use This <code>hwclock</code> Command Option</i>
Set the BIOS clock to the time given by <code>--date</code> .	<code>--set</code>
Specify the time that will be set for the BIOS clock.	<code>--date=[YYYY-MM-DD hh:mm:ss]'</code>
Set the system time from the BIOS clock.	<code>--hctosys</code>
Set the BIOS clock from the system time.	<code>--systohc</code>
Set the BIOS clock to the UTC.	<code>--utc</code>

The `ntpdate` Command

When the `ntpd` daemon is not running, you can use the `ntpdate` command to manually synchronize your system time with the NTP server. For example, `ntpdate -s {FQDN of the`

`NTP server}`.

How to Maintain the System Time

Follow these general procedures to maintain the system time.

Synchronize the System Clock with the Remote Time Server Using the `system-config-date` Command

To synchronize the system clock with the remote time server using the `system-config-date` command:

1. Log in as **root** in the GUI.
2. Open the **Date/Time Properties** window.
 - In **GNOME Panel**, select **System Tools** → **Settings** → **Date & Time**.
 - On the terminal, enter the `system-config-date` command.
3. Select the **Network Time Protocol** tab.
4. Synchronize the system clock with the remote time server using NTP.
 - a. Enable the **Network Time** option.
5. To apply the settings and close the window, select **OK**.

Synchronize the System Clock with the Remote Time Server Using the `/etc/ntp.conf` File

To synchronize the system clock with the remote time server using the `/etc/ntp.conf` file:

1. Log in as **root**.
2. To navigate to the **/etc** directory, enter `cd /etc`.
3. To open the **ntp.conf** file, enter `vi ntp.conf`.
4. Specify the time server details.
 - To set the server details, enter `server { ip-address | FQDN of the time server }`.
 - To set the drift file location, enter `drift file { drift file location }`.
5. Save and close the file.
6. To manually reset the clock, enter `ntpdate`.
7. To display a list and summary of the NTP servers known to the server, enter `ntpq -p`.

Note: <code>ntpq</code> is a utility that is used to monitor the <code>ntpd</code> service and determine its performance.
--

Set the System Date and Time

To set the system date and time:

1. Log in to the CLI as **root**.
2. Enter date `[MMDDHHmm][CC][YY][.ss]`, where MM indicates the month, DD indicates the date, HH indicates the hour, mm indicates the minute, CC refers to the century part of the year, YY refers to the year, and ss refers to the seconds. For example, 10111555 indicates October 11th 3:55 P.M.

Configure the Correct Time Zone for the System

To configure the correct time zone for the system:

1. Log in to the CLI as **root**.
2. If desired, to view the current time zone, enter `date`.
3. To start the menu-driven time zone selection process, enter `tzselect`.
4. Enter the number corresponding to the continent or ocean.
5. Enter the number corresponding to the country.
6. If necessary, for countries with multiple time zones, enter the number corresponding to the time zone.
7. To accept the settings, enter **1**.
8. Apply the changes.
 - a. To view the value of the TZ variable, make a note of the line `TZ='{Continent/City}'`; and export TZ.

- b. Enter `vi /etc/profile`.
 - c. In a new line, enter `TZ='{Continent/City}'`.
 - d. To export the variable, type `export TZ`.
 - e. Save and close the file.
9. To view the changed settings, log out and log in.

Set the BIOS Clock to the Correct Time in UTC

To set the BIOS clock to match the correct time in UTC:

1. Log in to the CLI as **root**.
2. To verify the UTC for your time zone, enter `date --utc`.
3. Set the BIOS clock.
 - Enter `hwclock --set --date='YYYY-MM-DD hh:mm:ss'`.
 - Enter `hwclock --utc`.

Synchronize the Clock Over NTP

To synchronize the clock over NTP:

1. Log in to the CLI as **root**.
2. Configure the NTP client.
 - a. Enter `vi /etc/ntp.conf`.
 - b. To specify the NTP server, enter `server {FQDN of the NTP server}`. If you have multiple NTP servers, specify the server address in sequence.
 - c. To synchronize with the NTP server, enter `restrict {FQDN of the NTP server} mask 255.255.255.255 nomodify notrap noquery`.
 - d. Save and close the file.
 - e. To start the NTP service, enter `service ntpd start`.
 - f. If necessary, to specify the drift in the time, edit the `/var/lib/ntp/drift` NTP drift file.

	Note: The <code>ntpd</code> daemon is used to synchronize the system time with the NTP server.
--	---

ACTIVITY 10-1

Managing Jobs and Processes Review

Scenario

Answer the following review questions.

1. Which of the process management tools presented do you expect will be useful in your environment? Why?

2. Do you think automating system processes affect a system's performance?

Why?

Summary

In this lesson, you managed essential jobs and processes on a Linux system. This will enable you to effectively track the usage of system resources and manage resource allocation efficiently.

11 Managing System Services

Lesson Time: 1 hour, 30 minutes

Lesson Introduction

Now that you examined the way processes operate in a Red Hat® Enterprise Linux® environment, you can manage the services that run on a Linux system. These include basic services and other services that are required by the kernel to process requests. In this lesson, you will manage system services.

With system services, you can make various system resources available to different users. At times, you may need to start, stop, or restart services to keep a system running efficiently. By managing system services, you can ensure that the system is working at the optimum level and users are able to derive maximum benefits.

Lesson Objectives

In this lesson, you will manage system services. You will:

- Configure system services to improve system performance.
- Monitor system logs.
- Configure Security-Enhanced Linux (SELinux).

TOPIC A Configure System Services

On a Linux system, there are numerous services that will be running simultaneously. These services need to be secured and managed properly in an efficient manner to avoid any clogging of system resources. In this topic, you will configure system services.

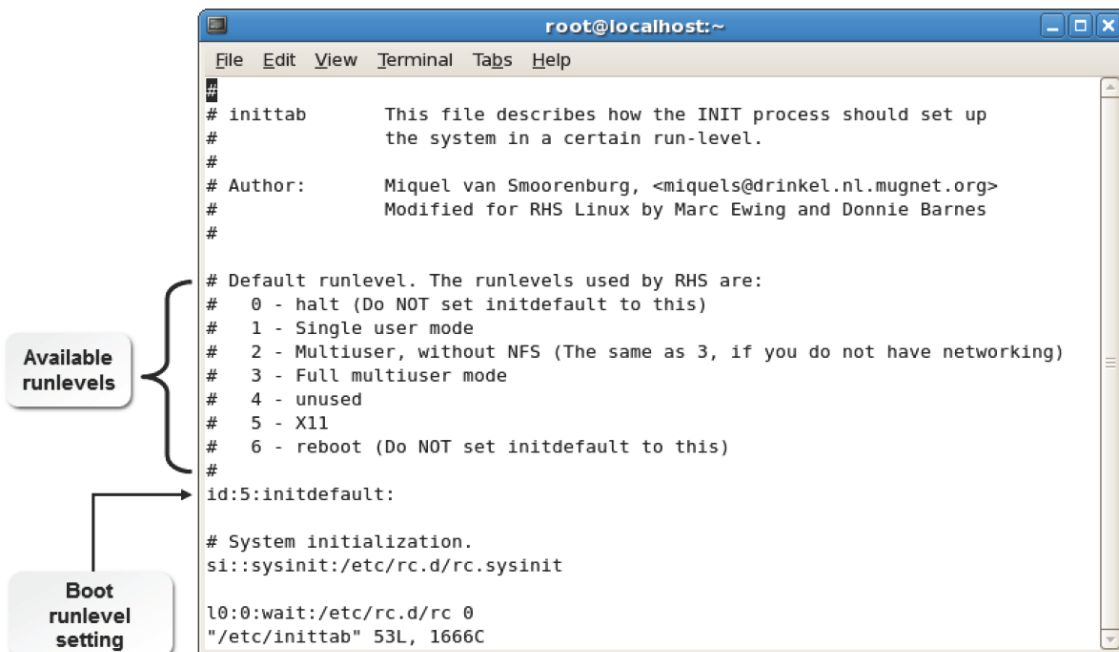
As a Linux administrator, you will often face performance problems with your system, such as slow processing and improper system response. Often, these problems are a result of improperly managed services that utilize more system resources, causing other processes to run on minimal resources. By managing system services properly, you will be able to increase the efficiency of your system.

System Initialization

System initialization begins when a system is booted. It involves the loading of the operating system and its various components, including the boot process. System initialization is carried out by the init program in Linux. The init program refers to the configuration file and initiates the processes listed in it. This prepares the system to run the required software. Programs on the system will not run without system initialization. We will cover both SysVinit and Systemd initialization in this topic.

The inittab File

The **inittab** file found in the **/etc** directory stores details of various processes related to system initialization. It also stores details of the runlevels in use.



```

root@localhost:~
File Edit View Terminal Tabs Help

#
# inittab          This file describes how the INIT process should set up
#                  the system in a certain run-level.
#
# Author:          Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#                  Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
#
# Default runlevel. The runlevels used by RHS are:
#  0 - halt (Do NOT set initdefault to this)
#  1 - Single user mode
#  2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#  3 - Full multiuser mode
#  4 - unused
#  5 - X11
#  6 - reboot (Do NOT set initdefault to this)
#
#
id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
"/etc/inittab" 53L, 1666C

```

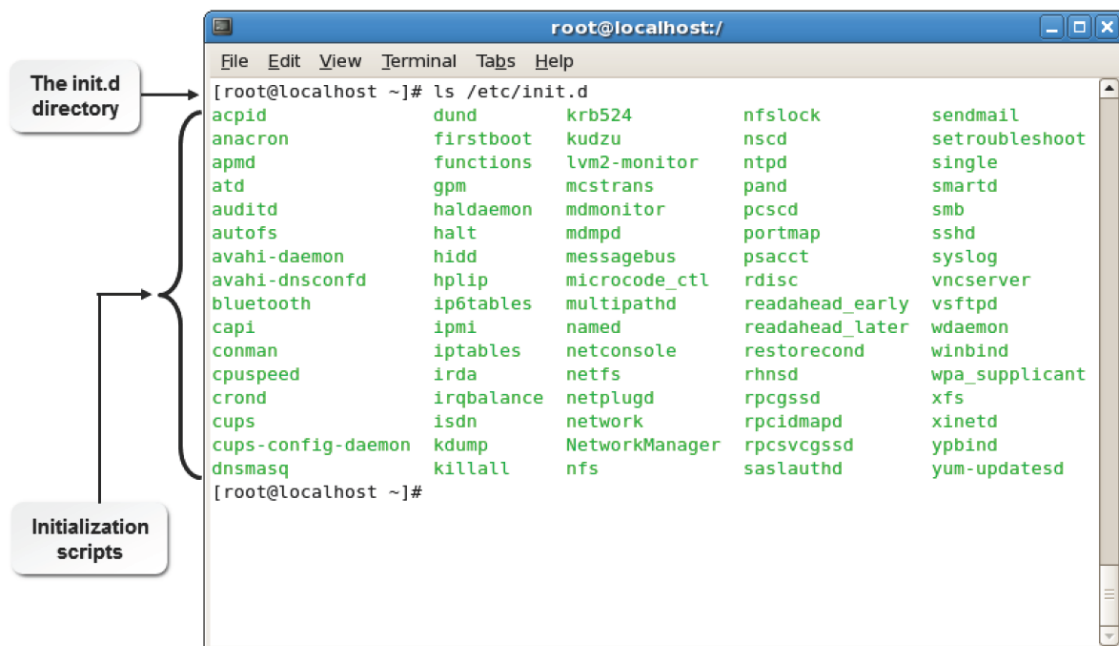
Figure 11-1: The inittab file showing runlevel details.

Data Storage Format

The **inittab** file stores data in the *id:runlevels:action:process* format.

The /etc/init.d Directory

The **init.d** directory found in the **/etc** directory stores initialization scripts for services. These scripts, called system V scripts, control the initiation of services in a particular runlevel. These runlevels are called system V runlevels. The scripts are invoked from the **/etc/inittab** file when the system initialization begins, using the symbolic links found in the file. System V scripts are highly flexible and can be configured according to the needs of a user. Some of the services listed in the **init.d** directory are anacron, cups, and bluetooth.



```

root@localhost:/
File Edit View Terminal Tabs Help

[root@localhost ~]# ls /etc/init.d
acpid          dund           krb524         nfslock        sendmail
anacron        firstboot     kudzu          nscd           setroubleshoot
apmd           functions     lvm2-monitor   ntpd           single
atd            gpm           mcstrans       pand           smartd
auditd         haldaemon     mdmonitor      pcsd           smb
autofs         halt          mdmptd         portmap        sshd
avahi-daemon   hidd          messagebus     psacct         syslog
avahi-dnscfnd hplip         microcode_ctl  rdisc          vncserver
bluetooth      iptables      multipathd     readahead_early vsftpd
capi           ipmi          named           readahead_later wdaemon
conman         iptables      netconsole     restorecond    winbind
cpuspeed       irda          netfs          rhnsd          wpa_supplicant
crond          irqbalance   netplugd       rpcgssd        xfs
cups           isdn          network        rpcidmapd      xinetd
cups-config-daemon kdump        NetworkManager rpcsvcgssd     ypbind
dnsmasq        killall       nfs            saslauthd     yum-updatesd

```

Figure 11-2: System and service initialization scripts are found in the init.d directory.

Syntax

The syntax for running scripts of the services in the **/etc/init.d** directory is */{service name}*

{start|stop|status|restart}.

The chkconfig Command

The chkconfig command can be used to control services in each runlevel. It controls services through the symbolic links found in the initialization scripts of services. It can also be used to start or stop services during system startup.

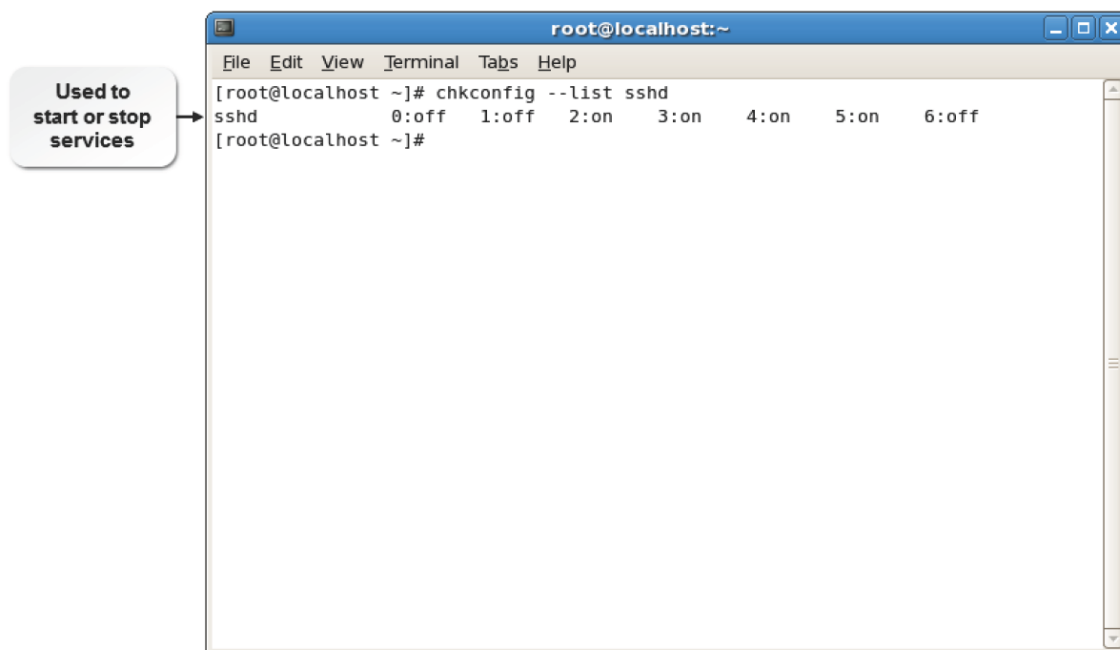


Figure 11-3: The chkconfig command and its options.

The chkconfig command has various options. Some of the frequently used options are listed in the following table.

Option	Enables You To
--level	Specify the runlevel in which the service has to be enabled or disabled.
--add	Add a service to the list of services managed by the chkconfig command.
--del	Delete a service from the list of services managed by the chkconfig command.
--list	List the services managed by the chkconfig command in all runlevels.
on	Start a service at system startup.
off	Stop a service at system startup.
reset	Reset the status of a service.

Syntax

The syntax of the chkconfig command is `chkconfig [option] {service name} {on|off| reset}`.

The chkconfig, service, and systemctl commands

In Systemd-based systems, such as CentOS/RHEL 7, the chkconfig and service commands are provided for compatibility, but have been replaced with the systemctl command.

chkconfig/service Command	systemctl Command Equivalent	Description
chkconfig --add service	systemctl enable service	Enable a service to be started on boot.
chkconfig --list	systemctl list-unit-files	List configured system services and their boot configuration.

chkconfig/service Command	systemctl Command Equivalent	Description
chkconfig --del service	systemctl disable service	Disable a service so that it is no longer started on boot.
service start service	systemctl start service	Start (activate) a service immediately.
service stop service	systemctl stop service	Stop (deactivate) a service immediately.
service restart service	systemctl restart service	Restart a service immediately.
service status service	systemctl status service	Show the status of a service, and whether it is running or not.

The /etc/sysconfig Directory

The **/etc/sysconfig** directory contains configuration files for services that should be started at system startup. These files contain settings that describe how these services must be initialized when the system boots. Some of the services listed in the **/etc/sysconfig** directory include bluetooth, irda, and kdump.

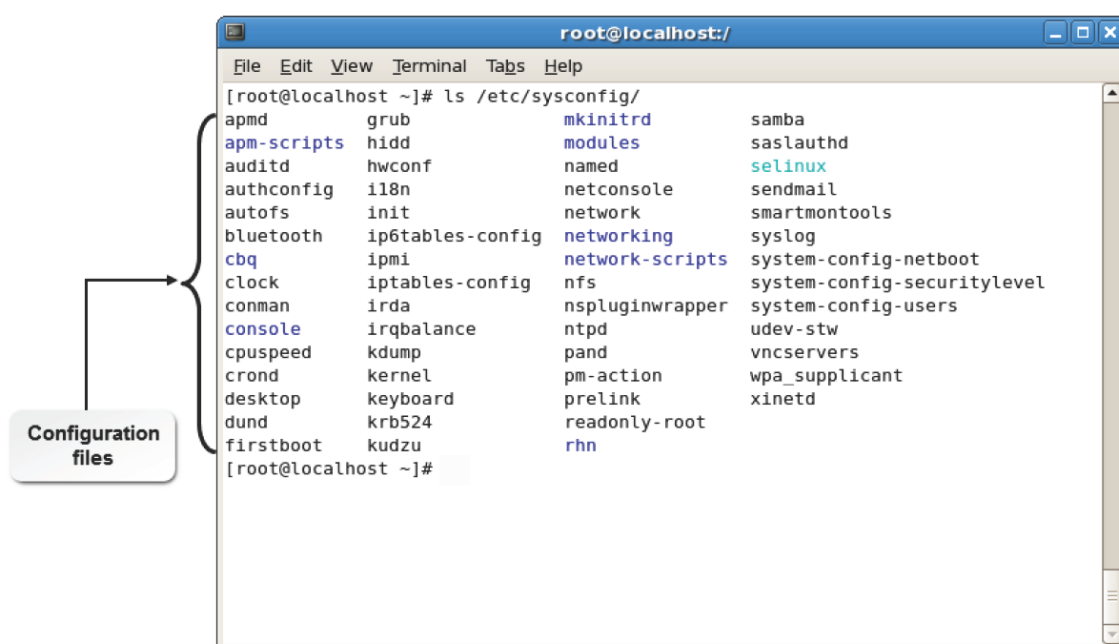


Figure 11-4: The sysconfig directory with various configuration files invoked during system startup.

The inetd Command

The **inetd** command, also called the Internet super-server, is a system service daemon that enables you to start programs needed for accessing different Internet services. When you request for a specific Internet service, inetd will start all the related services. It reduces the system load by running one daemon to support several related services without actually running all the daemons at the same time. The inetd command uses the **/etc/inetd.conf** file to configure services. This command is no longer used in most of the latest versions of Linux distributions and is replaced by the **xinetd** command.

How to Configure System Services

Follow these general procedures to configure system services.

Configure Services Using the chkconfig Command

To configure services at different runlevels using the chkconfig command:

1. Log in as **root**.
2. Configure services at different runlevels.

- To display whether the service name should be stopped or started at each runlevel, enter `chkconfig --list {service name}` or `systemctl list-unit-files |grep {service name}`.
- To add a service to `chkconfig` management, enter `chkconfig --add {service name}` or `systemctl enable {service name}`.
- To remove a service from `chkconfig` management, enter `chkconfig --del {service name}` or `systemctl disable {service name}`.
- To stop, start, or reset a service for the mentioned runlevel, enter `chkconfig --level {levels} {service name} {on|off|reset}`.

TOPIC B Monitor System Logs

In the last topic, you configured system services to improve your system's efficiency. To ensure that the changes made to the system services are applied correctly, you need to track the status of each change you make. In this topic, you will monitor system logs.

As a system administrator, you will need to check whether all the changes made to a system are applied, to ensure that the system is working fine. When managing system services, it will be practically impossible to manually track each change made to different services. You can track these changes using the system log files.

System Logs

System logs are records of system activities that are tracked and maintained by the **syslogd** utility.

The **syslogd** utility runs as a daemon. System logs are usually started at boot time. System log messages include the date, the process that delivered the message, and the actual message.

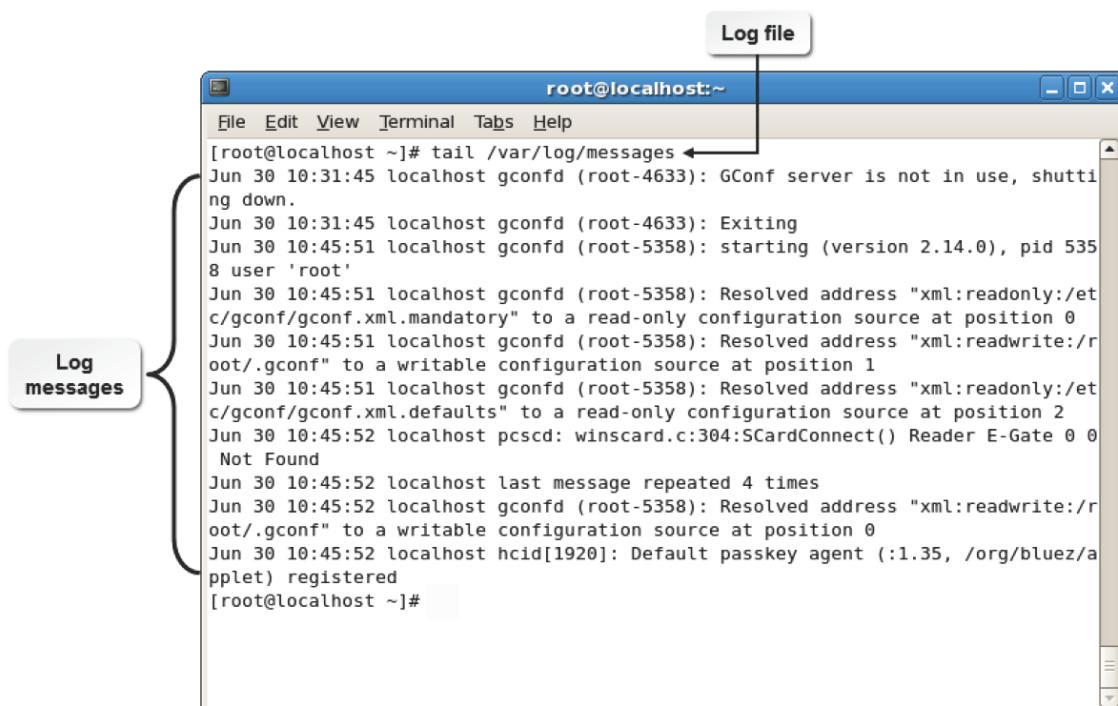


Figure 11-5: System logs and their messages.

Logging Services

A **logging service** is a daemon that is used to track logs or errors that are generated in a system.

Log messages are stored in a separate file called the **log file**, which is stored in the **/var/log** directory. The main log file is **/var/log/messages**. In addition to this log file, some services create their own log files.



Figure 11-6: Tracking log files using logging services.

The Central Network Log Server

The **central network log server** is a server that is used to implement centralized logging services.

This server receives all **syslog** messages from Linux or Windows® servers and from network devices such as routers, switches, firewalls, and workstations, across a network. The server logs data mining and online alerts, performs log analysis, and generates reports.

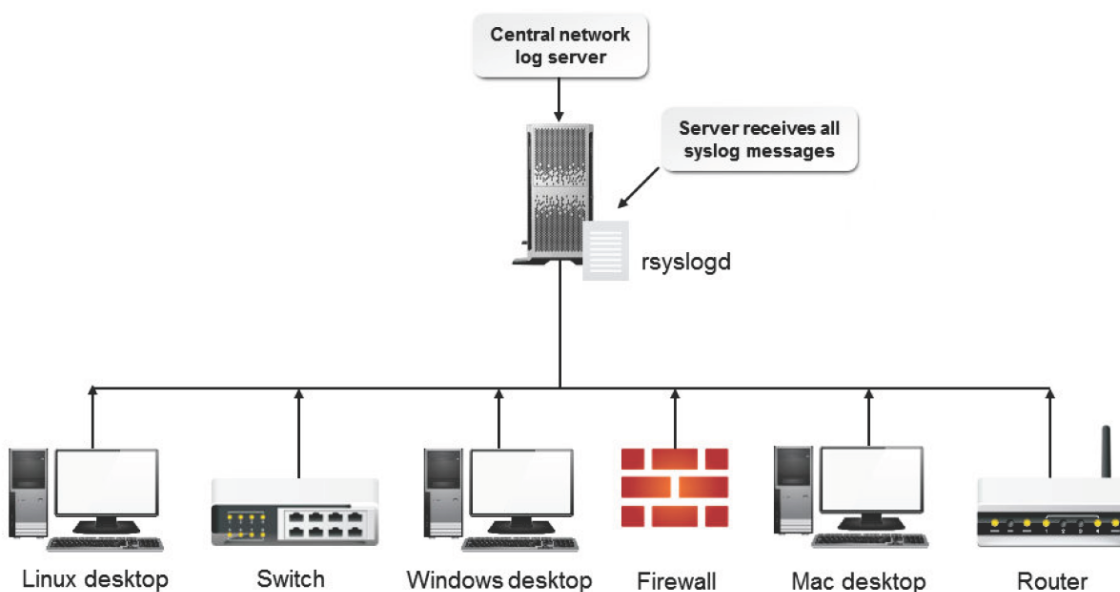


Figure 11-7: The central network log server receiving syslog messages from other servers and network devices.

Automating Log Analysis

During maintenance sessions, instead of manually parsing large log files, you can automate the log analysis by writing Perl or Bash scripts. For example, you can write a Perl script to automatically parse a mail log file and inform you about the rejected email messages. Ensure that you make a crontab entry for the script.

Perl

Practical Extraction and Reporting Language (Perl) is a programming language that is used to write scripts. Perl has a powerful feature that is used for manipulating strings; this is why it is extensively used by web servers to process data received from client browsers. In your Perl scripts, you can use **grep** and other textutils to extract specific text from log files.

Automatic Rotation

Automatic rotation is a system of regular rotation of logs to maintain a minimum log file size. The logrotate utility is used to perform automatic rotation. When executed, logrotate adds a .1 to the end of the file name of the current version of the log files. Previously rotated files are suffixed with .2, .3, and so on. Older logs have larger numbers at the end of their file names. Using automatic rotation, all copies of a file, with dates from when they were created, will be stored. Log files can be rotated on a daily, weekly, or monthly basis. Automatic rotation saves disk space because older log files are pushed out when a size limit is reached.

The syslogd Utility

The **syslogd** utility tracks remote and local system logs. Logs are characterized by their hostname and program field. The settings for syslogd are configured using the **/etc/syslog.conf** file.

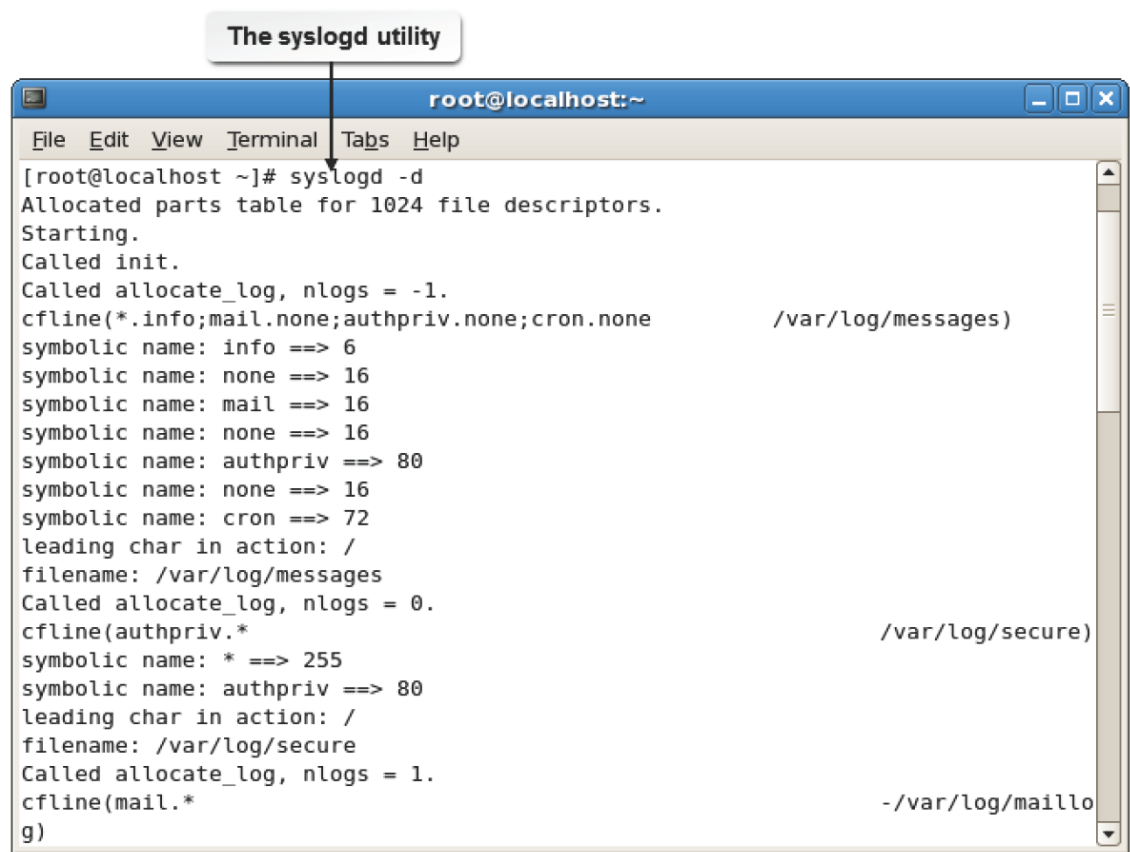


Figure 11-8: Turning on the debug mode in the syslogd utility.

The syslogd utility provides a number of options to manage specific functions. Some of the frequently used options are listed in the following table.

Option	Used To
-d	Turn on debug mode.
-f {file name}	Specify a new configuration file instead of <i>/etc/syslog.conf</i> .
-m {interval}	Specify a time interval between two mark timestamp lines in the log.
-r	Enable the syslogd utility to receive messages from a network.

Syntax

The syntax of the syslogd utility is `syslogd [options]`.

logger

The logger is the command interface to the system log module. The logger has options that allow you to customize the content that needs to be logged.

The klogd Utility

The **kernel logging daemon (klogd)** tracks kernel messages by prioritizing them. It listens to the source for kernel messaging and intercepts the messages. klogd runs as a client of syslogd, where the kernel messages are sent through the syslogd daemon. klogd also acts as a stand-alone program.

The klogd command provides a number of options to manage specific functions. Some of the frequently used options are listed in following the table.

Option	Enables You To
-c {n}	Set the default log level to n for messages, where n ranges from 0 to 7. <ul style="list-style-type: none">• 0–Emergency• 1–Alert• 2–Critical• 3–Error• 4–Warning• 5–Notice• 6–Information• 7–Debug
-p	Load the kernel module symbol information.
-k {file name}	Use the specified file as the source to store the kernel module symbol information.
-o	Read and log all kernel messages in the buffer in a single read.
-d	Switch to debugging mode.
-f {file name}	Log messages to the file that is specified.
-s	Use the system call interface for buffering the kernel messages.

Syntax

The syntax of the klogd command is klogd [options].

The /etc/syslog.conf File

The **/etc/syslog.conf** file controls the location where the syslogd information is recorded. This file consists of two columns. The first column lists the facilities and severities of the messages. The second column lists the files the messages should be logged to. By default, most messages are stored in the **/var/log/messages** file.

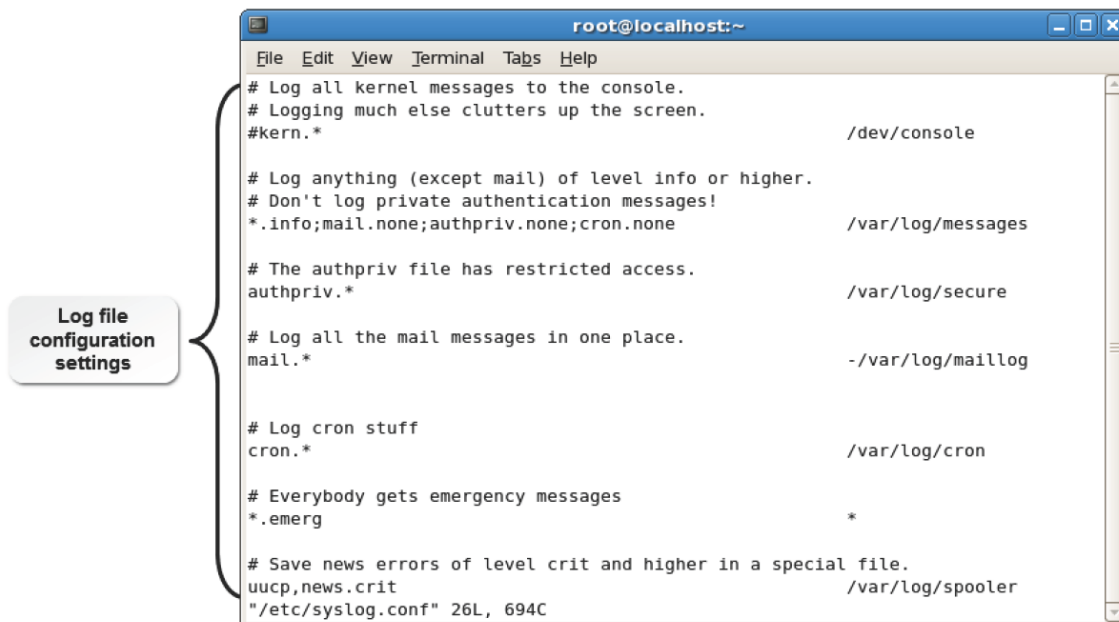


Figure 11-9: The syslog.conf file with the logging configuration settings.

Some applications maintain their own log files and directories independent of the **syslog.conf** file.

Each service has its own log storage file. Some of the frequently used log files are listed in the following table.

Log File	Description
/var/log/syslog	Stores the system log file, which contains information about the system.
/var/log/maillog	Stores mail messages.
/var/log/samba	Stores Samba messages.
/var/log/mrtg	Stores Multi Router Traffic Grapher (MRTG) messages.
/var/log/httpd	Stores Apache web server messages.

MRTG

Multi Router Traffic Grapher (MRTG) is free software, licensed under GNU General Public License (GPL), that is used to monitor and measure the traffic load on network links. The traffic load on a network is represented in graphical form.

The rsyslog Utility

The **rsyslog** utility tracks, forwards, and stores messages via the syslog protocol and local system logs, and is a more modern alternative to the older **syslogd** utility. Logs are characterized by their hostname and program field. The settings for rsyslog are configured using the **/etc/rsyslog.conf** file as well as multiple configuration files in the **/etc/rsyslogd** configuration directory.



Note: Modern versions of Red Hat Enterprise Linux, CentOS Linux, and Ubuntu® all use rsyslog as the default system log tool. Older versions of these Linux distributions used syslogd as the default system log tool.

```
localhost ~l# rsyslogd -N 5
gd: version 7.4.7, config validation run (level 5), master config /etc/rsyslog.conf
gd: End of config validation run. Bye.
localhost ~l# _
```

Figure 11-10: Validating configuration files for the rsyslog utility with rsyslogd.

The rsyslog utility provides a number of options to manage specific functions. Some of the frequently used options are listed in the table.

Option	Used To
-d	Turn on debug mode.
-f {file name}	Specify a new configuration file instead of the default <i>/etc/rsyslog.conf</i> .
-N {level}	Check configuration files to confirm they are correct and valid. Use a <i>level</i> of 1 or higher to control verbosity.

Syntax

The syntax of the rsyslog utility is `rsyslog [options]`.

The syslog-ng Utility

The [syslog-ng](#) utility tracks, forwards, and stores messages via the syslog protocol and local system logs, and is another more modern, enhanced alternative to the older **syslogd** utility. Logs are characterized by their hostname and program field, and include timestamps with millisecond granularity and timezone information. The settings for syslog-ng are configured using the */etc/syslog-ng/syslog-ng.conf* file as well as various related configuration files in the */etc/syslog-ng/* directory.

The syslog-ng utility provides a number of options to manage specific functions. Some of the frequently used options are listed in the following table.

Option	Used To
-d	Turn on debug mode.
-f {file name}	Specify a new configuration file instead of <i>/etc/syslog.conf</i> .
-F	Specify that syslog-ng should be run as a foreground process (do not go into the background after initialization).
-v	Display more verbose output.

<i>Option</i>	<i>Used To</i>
-e	Log error messages to stderr.
-t	Enable the display of trace messages.

Syntax

The syntax of the syslog-ng utility is `syslog-ng [options]`.

The journalctl Utility

The [journalctl](#) utility is a component of Systemd that manages and views log files created by the Journal component of Systemd. It may be used on its own, but is often used in conjunction with a traditional syslog daemon such as syslogd or rsyslog. Log information is collected and stored via the Systemd journald service, and may be viewed with the journalctl utility. The settings for journald are configured in the `/etc/systemd/journald.conf` file.

The journalctl utility provides a number of options to manage specific functions. Some of the frequently used options are listed in the following table.

<i>Option</i>	<i>Used To</i>
-n {number of lines}	Specify the number of lines of journal logs to display.
-o {output format}	Specify the format of the output, for example: short, verbose, or export.
-f	Display the most recent journal entries, and continuously update the display with new entries as they are added to the journal.
-p	Filter journal log output by priority (alert, err, warning, notice, info, etc.).
-b	Show log message from the current boot only (although previous boots may also be specified).

Syntax

The syntax of the journalctl utility is `journalctl [options]`.

The /var/log/journal/ directory

In its default configuration, the Systemd Journal only stores logs in memory, and logs are cleared on each system reboot. The Systemd Journal logs may be persisted after a reboot by creating the directory `/var/log/journal`. Systemd is configured to automatically persist logs into this directory if it exists.

Log File Analysis

The process of examining messages generated by logging daemons in log files is referred to as [log file analysis](#). Log messages are created in a format that is specific to an application or a vendor and are arranged in chronological order. During analysis, the format of log messages from different logging sources, such as operating systems, networks, and databases, is compared with a preset format. Also, log messages are categorized for each user with respect to the application, system, or system configuration accessed, to ensure user authentication.

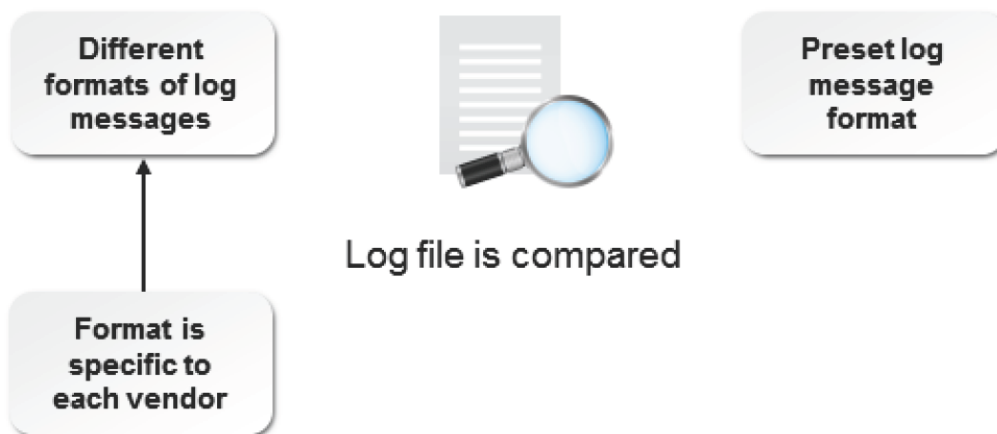


Figure 11-11: Various steps involved in log file analysis.

The lastlog Command

The **lastlog** command utilizes data from the **/var/log/lastlog** file to display the latest login details of all users. In addition to the login name, date, and time, it displays the terminal from where a user last logged in. The lastlog command is used by administrators to view user accounts that have never been used.

```

root@localhost:~# lastlog
Username      Port      From      Latest
root          :0        -         Wed Jul  8 12:05:46 -0400 2009
bin           -         -         **Never logged in**
daemon       -         -         **Never logged in**
adm          -         -         **Never logged in**
lp           -         -         **Never logged in**
sync         -         -         **Never logged in**
shutdown     -         -         **Never logged in**
halt         -         -         **Never logged in**
mail         -         -         **Never logged in**
news         -         -         **Never logged in**
uucp        -         -         **Never logged in**
operator     -         -         **Never logged in**
games        -         -         **Never logged in**
gopher       -         -         **Never logged in**
ftp          -         -         **Never logged in**
nobody       -         -         **Never logged in**
rpc          -         -         **Never logged in**
mailnull     -         -         **Never logged in**
smmsp        -         -         **Never logged in**
nscd         -         -         **Never logged in**
vcsa         -         -         **Never logged in**
rpcuser      -         -         **Never logged in**
  
```

Figure 11-12: The output of the lastlog command.

The grep Command

The grep command searches a file or list of files for a string and prints the lines that match the search string. The grep command has various options that allow you to specify search criteria.

The following table lists the options of the grep command.

Option	Used To
-h	Print matching lines without file names.
-w	Restrict the search to whole words only.
-c	Display a count of the number of matching lines and not the lines themselves.
-i	Ignore case while searching.
-l	List the file names that contain matching lines.
-n	Precede each line with the line number where it was found.
-s	Suppress the display of any error message.

<i>Option</i>	<i>Used To</i>
-e	Specify one or more patterns for searching.

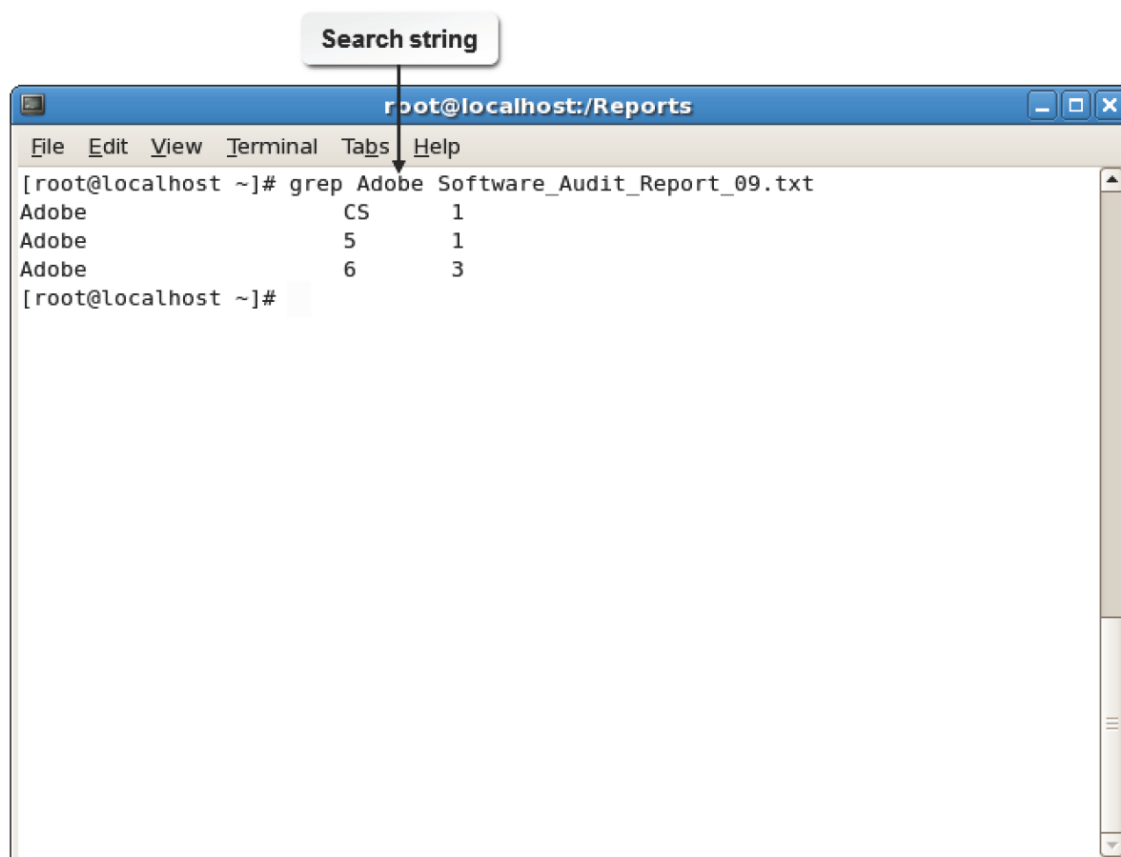


Figure 11-13: The *grep* command is used to search for a string in a file.

Syntax

The syntax of the *grep* command is *grep [options] {keyword} {file name}*.

fgrep and egrep Commands

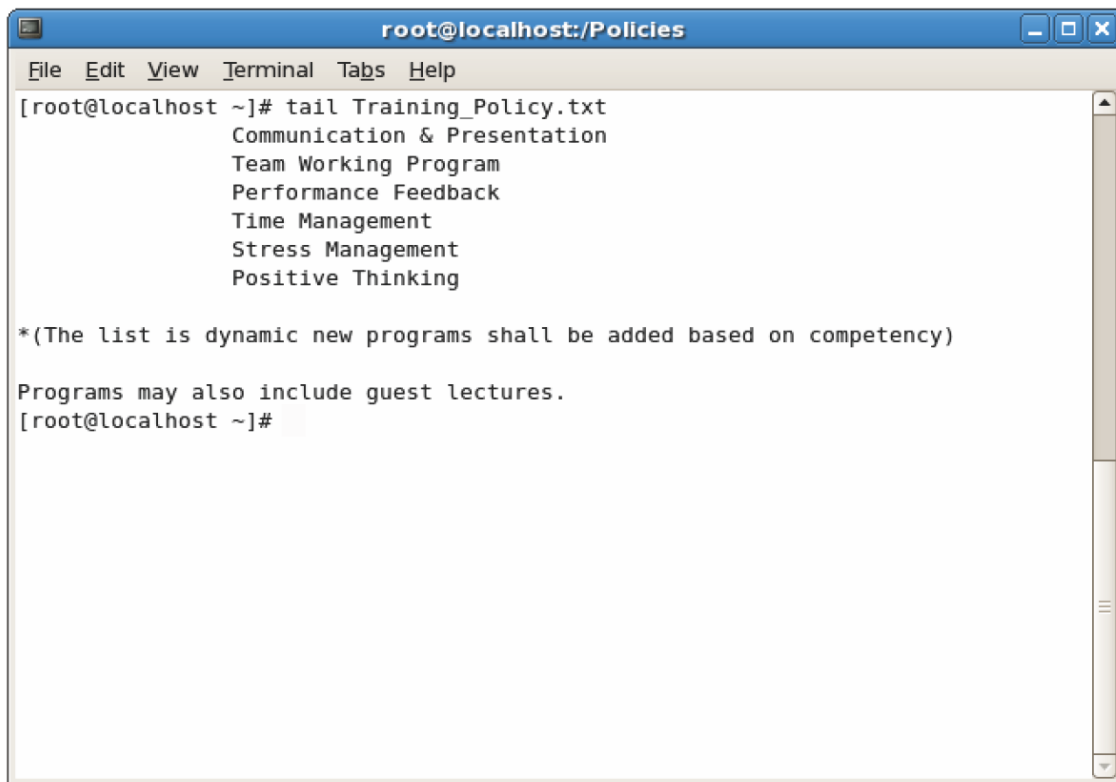
The *fgrep* command searches for multiple text patterns; however, this command's search is not based on regular expressions.

The *egrep* command searches for multiple text patterns, which may include a larger set of regular expression elements than *grep*.

The tail Command

The *tail* command is used to retrieve data from a file. By default, it displays the last 10 lines of the file. The *tail* command has various options. Some of the frequently used options are listed in the following table.

<i>Option</i>	<i>Enables You To</i>
--retry	Force the <i>tail</i> command to open a file that cannot be opened.
-n {total no. of lines}	Print the specified number of lines from the end of a file.
-c {total no. of bytes}	Print the specified number of bytes from the end of a file.
-f	Update the output of the <i>tail</i> command if any change is made to a file.



```
root@localhost:/Policies
File Edit View Terminal Tabs Help
[root@localhost ~]# tail Training_Policy.txt
Communication & Presentation
Team Working Program
Performance Feedback
Time Management
Stress Management
Positive Thinking

*(The list is dynamic new programs shall be added based on competency)

Programs may also include guest lectures.
[root@localhost ~]#
```

Figure 11-14: The output of the tail command.

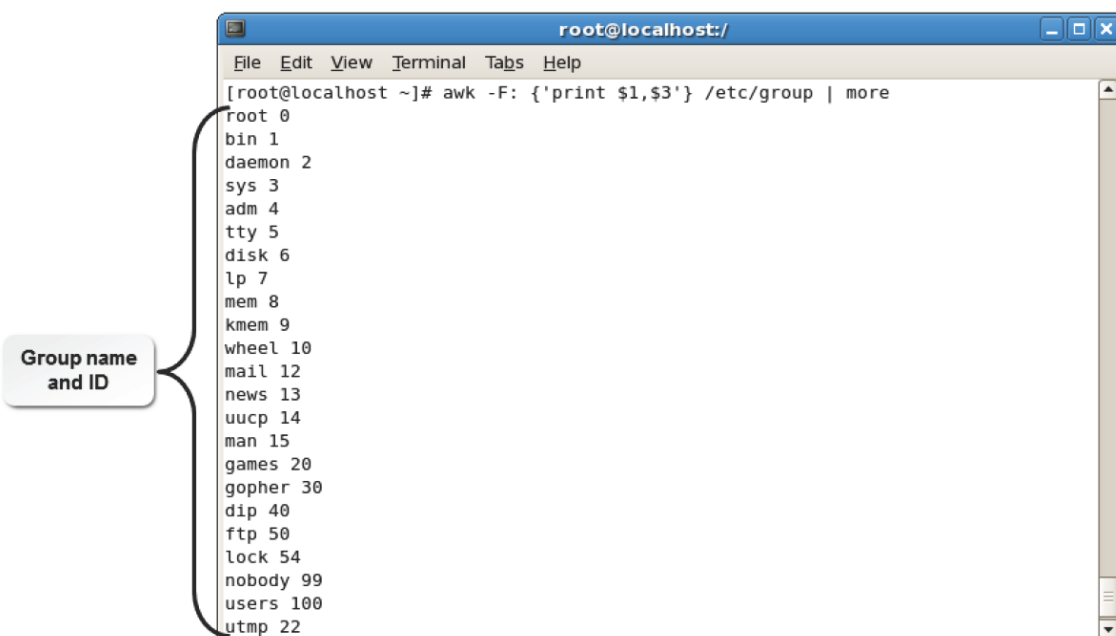
Syntax

The syntax of the tail command is `tail [options] {file name}`.

The awk Command

The **awk** command is a command that performs pattern matching. GNU's version of awk is called gawk. The awk keyword is followed by the pattern, the action to be performed, and the file name.

The action to be performed is given within curly braces. The pattern and the action to be performed should be specified within single quotes. If the pattern is not specified, the action is performed on all input data; however, if the action is not specified, the entire line is printed. The awk command can be executed from the command line or from within an awk script file.



```
root@localhost:/
File Edit View Terminal Tabs Help
[root@localhost ~]# awk -F: {'print $1,$3'} /etc/group | more
root 0
bin 1
daemon 2
sys 3
adm 4
tty 5
disk 6
lp 7
mem 8
kmem 9
wheel 10
mail 12
news 13
uucp 14
man 15
games 20
gopher 30
dip 40
ftp 50
lock 54
nobody 99
users 100
utmp 22
```

Group name and ID

Figure 11-15: Using the awk command to display only the group name and ID from the /etc/ group file.

Syntax

The syntax of the awk command is `awk [options] {file name}`.

Patterns

In awk scripts, you can provide patterns along with blocks of code. If a pattern matches any line in the input file, the code blocks in the script will be executed. The following table lists the types of patterns used.

Pattern	Description
/regular_expression/	Retrieves all the records beginning with "a," "b," or "c." Example: /[abc]/
relational_expression	Retrieves all the records whose first field contains the value "abc." Example: \$1 == "abc"
pattern_1 && pattern_2	Retrieves all the records whose first field contains the value "abc" and the second field contains the value "01." Example: (\$1 == "abc") && (\$2 == "01")
pattern_1 pattern_2	Retrieves records that satisfy the condition that the first field contains the value "abc" or the second field contains the value "01," or both. Example: (\$1 == "abc") (\$2 == "01")
pattern_1 ? pattern_2 : pattern_3	If the first field in a record contains the value "10," the fifth field is tested for its value. If the fifth record contains the value "20," then the record is printed. If the first field of a record does not contain the value "10," then the ninth field of the record is evaluated. If the ninth record contains the value "30," then the record is printed. Example: \$1 == "10" ? \$5 == "20" : \$9 == "30"
pattern_1, pattern_2	Prints a range of records, starting from the record whose first field contains the value "01." The records will be printed until the awk command finds a record whose first field contains the value "02." Example: \$1 == "01", \$1 == "02"

The sed Command

The [sed](#), or stream editor, command is a command line program that can be used to modify log files or text files according to command line parameters. The sed command can be used for global search and replace actions. The sed command has various options. Some of the common sed command options and their uses are given in the following table.

Option	Used To
d	Delete the lines that match a specific pattern or line number.
-n,p	Print only the lines that contain the pattern.
s	Substitute the first occurrence of the string in the file.
s,g	Globally substitute the original string with the replacement string for each occurrence in the file.

Syntax

The general syntax of the sed command is `sed'{address/pattern/action}' {file name}`

If there is an address, it follows the command name. The pattern formed by the user comes next, followed by the action to be performed when a match is found. The last argument is the name of the input file. The address, pattern, and action parameters are enclosed within single quotation marks.

How to Monitor System Logs

Follow these general procedures to monitor system logs.

Configure System Logs

To configure system logs:

1. Log in as **root**.
2. To open the configuration file, enter `gedit /etc/rsyslog.conf`.
3. To set the type and level of severity to be logged in the specified file, type `{facility}. {level of severity} / {location of the file that stores the log messages}`.
4. Save and close the file.
5. To restart the system log service and apply the changes, enter `service rsyslog restart`.

Configure rsyslog to Act as a Central Network Log Server

To configure rsyslog to act as a central network log server:

1. Log in to the CLI as **root**.
2. To open the **rsyslog** file, enter `gedit /etc/rsyslog.conf`.
3. Uncomment the lines that provide TCP syslog reception (**\$ModLoad imtcp** and **\$InputTCPServerRun 514**).
4. Save and close the file.
5. To restart the rsyslog service, enter `service rsyslog restart`.

Configure rsyslog to Send Log Output to a Central Log Server

To configure rsyslog to send the log output to a central log server:

1. Log in as **root**.
2. To open the system log configuration file, enter `gedit /etc/rsyslog.conf`.
3. To send the log output to a remote log server, type `{facility} {level of severity} @ {IP or FQDN of the log server}`.
4. Save and close the file.
5. To restart the system log service and apply the changes, enter `service rsyslog restart`.

Search and Replace Strings

To search and replace strings:

1. Log in as a user in the CLI.
2. Search and replace strings.
 - To replace the old string with the replacement string even if multiple occurrences of the old string are found in a single line, enter `sed 's/{old string}/{replacement string}/g' {file name}`.
 - To replace only the first occurrence of the old string with the replacement string even if multiple

occurrences of the old string are found in a single line, enter `sed 's/{old string}/{replacement string}' {file name}`.

Extract Information Manually from Log Files

To extract information manually from log files:

1. To view the current list of system log files, enter `ls /var/log`.
2. Use suitable commands to extract information from log files.
 - To print the output of specific columns from the selected file, enter `awk '{print ${column name} ${column name}}' {file name}`.
 - To manually scan the log files and extract values that match the specific activity, enter `grep -r "{d} {file name}"`.

TOPIC C Configure Security-Enhanced Linux (SELinux)

Previously, you monitored system log files to ensure that the changes made to the services are applied correctly. Information on a system needs to be protected from misuse or damage by using appropriate security measures such as Security-Enhanced Linux (SELinux). In this topic, you will configure SELinux.

Even when a server is configured and running correctly, it is possible that security attacks may occur, which could be aimed at both organizations and individuals. Imagine that your company's servers are damaged and all your critical data are erased. You can prevent this by setting up the required security checks using SELinux.



Note: SELinux introduces considerable security restrictions at every level of access on a Linux server, and as such has found slow adoption in production due to conflicts with various popular Linux applications and services. Having said that, SELinux is a best practice for ensuring the security of a Linux server and disabling it is not recommended in production servers.

Types of Access Controls

Access control is a method of restricting access to system resources. Only authorized programs will be allowed to access system resources. In Linux, there are two types of access controls.

Access Control Method	Description
Discretionary Access Control (DAC)	<p>In DAC, the system checks the resources over which a user has access rights. The rights of the user are identified using the authentication information such as user identity and password. Under DAC, there are two types of permissions: the administrator permissions and the non-administrator permissions. For application programs to run, administrator access has to be provided. Administrator access provides full discretion over the filesystem and exposes it to security threats. For example, a malicious program or process started by a user having administrator access can damage data in a filesystem.</p> <p>DAC is the standard security strategy in Linux in which the User/ Group/Other file permissions are managed.</p>
Mandatory Access Control (MAC)	<p>In MAC, the system checks the resources over which a user does not have access rights. MAC is applied through SELinux. The rights of the user are identified using authentication such as the SELinux user identity, role, and type of access.</p> <p>MAC is the opposite of DAC, where permissions have to be defined for all processes (known as subjects) as to how they access resources (known as objects) such as files, directories, devices, memory resources, and other processes. An action is an operation, such as append, write, read, create, execute, and rename, that a subject can perform on an object. This is implemented using security policies that control the interaction between the processes and the objects.</p> <p>For example, when a subject tries to access an object, the security policy is checked to verify whether the subject is authorized to access the object before granting the access.</p>

Security-Enhanced Linux

Security-Enhanced Linux (SELinux) is the default security enhancement feature provided with CentOS and Red Hat Enterprise Linux, and is available on other distributions. It was developed by the U.S. National Security Agency while implementing various security policies on Linux operating systems. It provides additional filesystem and network security so that unauthorized processes cannot access or tamper with data, bypass security mechanisms, violate security policies, or execute untrustworthy programs. It enforces MACs on processes and resources and allows information to be classified and protected based on its confidentiality and integrity requirements. This confines the damage caused to information by malicious applications.



Note: The SELinux feature comes as part of CentOS and Red Hat Enterprise Linux (RHEL) 4 and the later versions.

SELinux Modes

SELinux has three different modes.

<i>Mode</i>	<i>Description</i>
Disabled	In this mode, SELinux is turned off. So, MAC will not be implemented and the default DAC method will be prevalent.
Enforcing	In this mode, all the security policies are enforced. Therefore, processes cannot violate the security policies.
Permissive	In this mode, SELinux is enabled, but the security policies are not enforced. So, processes can bypass the security policies. However, when a security violation occurs, it is logged and a warning message is sent to the user.

Security Context

Security context is the collection of all security settings pertaining to processes, files, and directories. Security context consists of three elements: user, role, and type. Based on the security context attributes, SELinux decides how subjects access objects on the system.

Security Policies

A security policy defines access parameters for every process and resource on the system.

Configuration files and policy source files located in the **/etc/selinux** directory can be configured by the root user.

<i>Security Policy Type</i>	<i>Description</i>
Targeted	According to the targeted policy, except the targeted subjects and objects, all other subjects and objects will run in an unconfined environment. The untargeted subjects and objects will operate on the DAC method and the targeted ones will operate on the MAC method. A targeted policy is enabled by default.
Strict	A strict policy is the opposite of a targeted policy, where every subject and object of the system is enforced to operate on the MAC method.

How to Configure SELinux

Follow these general procedures to configure SELinux.

Control the SELinux State on the System

To control the SELinux state on the system:

1. Log in as **root** in the GUI.

2. Control the SELinux state on the system.

- Control the SELinux state using the **/etc/sysconfig/selinux** file.
 - a. To open the **selinux** file, enter `vi /etc/sysconfig/selinux`.
 - b. Switch to insert mode.
 - c. To change the **SELINUX** variable to control the mode of the SELinux policy, set **SELINUX={enforcing | permissive | disabled}**.
 - d. To change the **SELINUXTYPE** variable to control the type of the SELinux policy, set **SELINUXTYPE={targeted | strict}**.
 - e. Save and close the file.
- Switch between enforcing mode and permissive mode.
 - a. To switch between enforcing mode and permissive mode, enter `setenforce {1 | 0}`, respectively.
 - b. To view the mode, enter `getenforce`.

View the Security Context for Files and Processes

To view the security context for files and processes:

1. Log in to the CLI as **root**.
2. View the security context for files and processes.
 - To view the security context of the specified file or directory, enter `ls -Z[options] {file or directory name}`.
 - To view the security context of the specified process, enter `ps -Z[options] {process name}`.

Change the Security Context for Files

To change the security context for files:

1. Log in to the CLI as **root**.
2. Change the security context for files.
 - To set the specified security context to the specified file or directory, enter `chcon -[options] {security context} {file or directory name}`.
 - To restore the default security context to the specified file or directory, enter `restorecon {file or directory name}`.

ACTIVITY 11-1

Managing System Services Review

Scenario

Answer the following review questions.

1. How will you use system logs to troubleshoot system problems?
2. Which level of access control and/or SELinux would you use in your organization and why?

Summary

In this lesson, you configured system services, monitored system logs, and configured SELinux.

This will enable you to utilize your Linux system at its optimum level.

12 Configuring Network Services

Lesson Time: 3 hours, 15 minutes

Lesson Introduction

In the last lesson, you configured basic system services on your computer. Sometimes, you may need to establish a connection with other computers to communicate with them. In this lesson, you will configure network services.

A network enables computers to communicate with each other and share data, software, and hardware resources. Network services allow system administrators to disseminate information, administer systems remotely, enable communication through mail or chat systems, facilitate technology sharing, manage software licenses, and control unauthorized access.

Lesson Objectives

In this lesson, you will configure Linux services to provide users with network connectivity.

You will:

- Connect to a network.
- Configure routes.
- Configure client network services.
- Manage remote network systems.

TOPIC A Connect to a Network

In the last lesson, you worked with data and other utilities solely on your computer. Sometimes, you may need to share data and devices with other computers on a network. In this topic, you will connect to a network.

As a network administrator, you may need to manage and troubleshoot servers, network services, and workstations. To manage a network, you should understand the basic concepts of a network and its components. By connecting to a network, you will be able to implement network services required to efficiently manage a network with numerous systems.

Networks

A **network** is a group of computers connected together to communicate with each other and share resources. Each device on the network is referred to as a node. The components of a network are servers; clients; communication cables; resources, such as files or printers; network adapters; and network protocols.

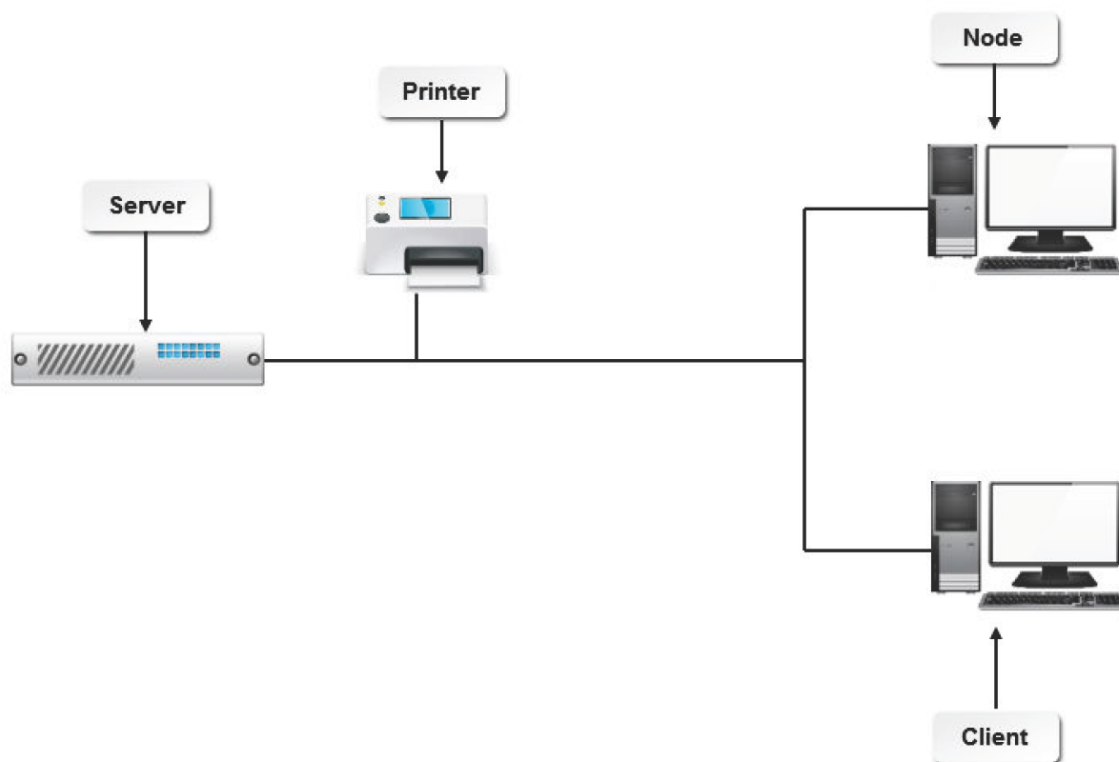


Figure 12-1: An example of a network.

Types of Networks

Networks can be broadly classified into two types based on their size.

Network Type	Description
Local Area Network (LAN)	A network that connects computers in a small geographical area such as a floor or a building. Computers on a LAN are connected through Ethernet at speeds of 10 Mb/s, 100 Mb/s, or 1,000 Mb/s. Compared to other types of networks, such as MAN and WAN, a LAN is a high-speed data network.
Wide Area Network (WAN)	A network that connects computers in a wide geographical area. Computers on a WAN are connected through bridges, routers, hubs, and repeaters. This network can extend across a country or around the world. A WAN may connect LANs and MANs.

Metropolitan Area Networks

A **Metropolitan Area Network (MAN)** is a network that connects computers in a broad geographical area such as a city and its suburbs. Computers on a MAN are connected through switches, access servers, and ISDN terminal adapters. This network is a medium-speed data network and can connect two or more LANs.

Network Protocols

A **network protocol** is a set of rules that enables communication and data transfer among network devices. The rules specify how data should be shared among systems. Network protocols include conventions that specify message acknowledgment or data compression.

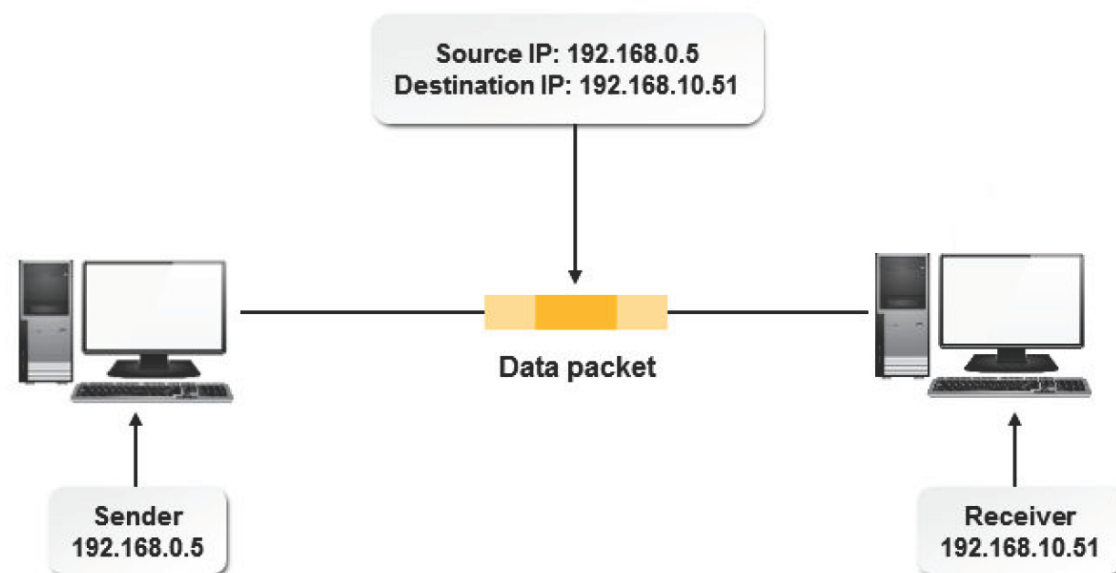


Figure 12-2: Data transmitted between two computers on a network.

A network protocol contains rules that establish the method by which data is transmitted virtually on all networks today. It transmits data to the destination and acknowledges the source to ensure that the data is delivered.

Types of Network Protocols

A network protocol is chosen based on the network setup and network requirements. The protocols range from a high level to a low level, based on the network capacity.

Network Protocol	Used To
Transmission Control Protocol (TCP)	Transfer packets of data from one system to another on a network. TCP binds with Internet Protocol (IP) and acts as a core layer for the Internet. TCP guarantees that the packets are delivered reliably and in the same order in which they are sent.
HyperText Transfer Protocol (HTTP)	Transfer hypertext files across the World Wide Web. HTTP allows web browsers and web servers to communicate with each other to request a file and transfer contents. There are many versions of HTTP.
File Transfer Protocol (FTP)	Send and receive files over the Internet. FTP is based on the client/server architecture. A user with an FTP client has to log on to a remote system, navigate to the filesystem, and upload and download files from that system.
Internet Control Message Protocol (ICMP)	Handle error and control messages. It does not transfer any application data, but transfers information about the status of the network. The ping utility uses ICMP for probing messages. It is useful in Internet protocol network management and administration.
User Datagram Protocol (UDP)	Transmit data in the form of small packets that are sent independently. This protocol cannot determine whether or not the data reached its destination. It is a transport protocol that is part of the TCP/IP suite of protocols

IP Addresses

An [IP address](#) is a unique address that identifies a host on the Internet. It is most commonly a 32-bit binary number that is displayed as four 8-bit decimal numbers, called octets, separated by periods. For example, 155.40.104.49 is an IP address.

The first two octets of the address identify the network on which a host resides and the next two octets identify the host.

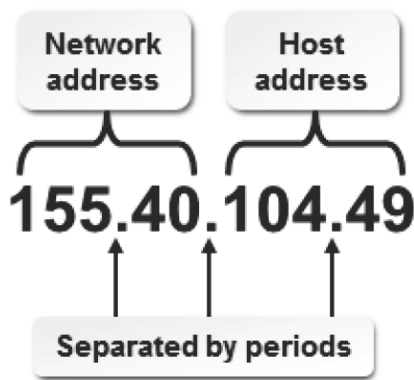


Figure 12-3: An IP address that identifies the network on which a host resides.

IP Versions

IP Version 4 (IPv4) and **IP Version 6 (IPv6)** are the two versions of the Internet protocol that are currently in use. With the number of hosts on the Internet growing at a fast pace, the earlier version, IPv4, which adopts a 32-bit addressing format, has limited unique IP addresses for public Internet access, in addition to reserved and consumed addresses. There is a chance of running out of IP addresses and routing can become complicated. This can restrict future Internet access.

So, a new version of IP, called **IP Next Generation (IPng)** or IPv6, is implemented on the Internet. The proposed Internet standard can increase the available pool of IP addresses by implementing a 128-bit binary address space. IPv6 also includes new efficiency features such as simplified address headers, hierarchical addressing, support for time-sensitive network traffic, and a new structure for unicast addressing.

IPv6 and IPv4 Compatibility

IPv6 is not compatible with IPv4, but the two may coexist on the same networks and more and more academic and production networks are deploying IPv6. Full adoption of the IPv6 standard will require a general conversion of IP routers to support interoperability, which is still underway on a lot of networks.

Comparing IPv4 and IPv6

IPv4 and IPv6 differ drastically in several key areas. The major differences are shown in the following table.

IPv4	IPv6
Uses 32-bit addresses.	Uses 128-bit addresses.
Address Resolution Protocol (ARP) uses broadcast ARP request to resolve IP to MAC/ Hardware address.	Multicast Neighbor Solicitation messages resolve IP addresses to MAC addresses.
Addresses are distributed by Dynamic Host Control Protocol (DHCP) or are manually configured.	Addresses are distributed by Neighbor Discovery Protocol, DHCPv6, or manually.
Headers include a Checksum.	Headers do not include a Checksum.
576 byte packet size.	1280 byte packet size.
Broadcast addresses send to all network nodes.	Does not use Broadcast.
Routers and hosts may fragment packets.	Routers do not perform fragmentation.
Host names are mapped to IP addresses using A records.	Host names are mapped to IP addresses using AAAA (quad-A) records.

Private Networks

Three IP network address blocks are reserved for private networks, or intranets.

- 10.0.0.0 to 10.255.255.255
- 172.16.0.0 to 172.31.255.255
- 192.168.0.0 to 192.168.255.255

These addresses can be used for setting up internal IP networks, such as an organization's LAN.

Routers on the Internet will not forward packets coming from these addresses. These addresses are meaningful only for the network to which they belong. Within these range of addresses, two or more organizations can have the same IP address assigned to a machine. In addition to these three IP network addresses, 127.0.0.0 to 127.255.255.255 is reserved as a loopback network, and specifically 127.0.0.1 is most often used as a loopback address.

Subnet Masks

A **subnet mask** is a 32-bit number that is assigned to each system to divide the 32-bit binary IP address into network and node portions. This makes TCP/IP routable. A subnet mask uses a binary operation to remove the node ID from the IP address, leaving just the network portion. The network portion of an IP address can also be referred to as a netmask. Subnet masks use the value of eight 1s in binary, or 255 in decimal, to mask an entire octet of the IP address. A subnet mask acts as a filter that tells the server whether an IP address is on a local network or on a remote network. Subnet masks help routers identify whether a data packet needs to be retained on the local network or sent to another network.

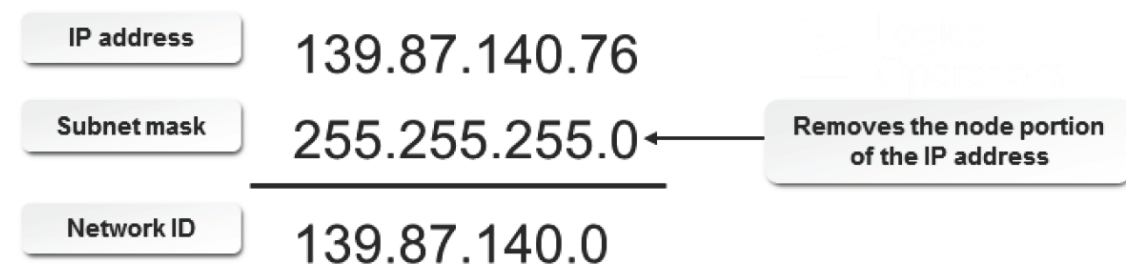


Figure 12-4: The subnet mask removes the node portion of the IP address.

Subnet Mask Values

The following table lists the subnet mask values.

Subnet Mask	Number of Subnets	Number of Hosts in Class A	Number of Hosts in Class B	Number of Hosts in Class C
128	2	8,388,606	32,766	126
192	4	4,194,302	16,382	62
224	8	2,097,150	8,190	30
240	16	1,048,574	4,094	14
248	32	524,286	2,046	6
252	64	262,142	1,022	2
254	128	131,070	510	0
255	256	65,534	254	0

IP Address Classes

The designers of the TCP/IP suite defined five ranges of addresses, called address classes, for specific network uses and sizes. An **IP class** is a block of IP addresses that can be assigned to businesses or governments, based on the size and need. Each IP address belongs to an IP class.

There are five IP classes: Class A, Class B, Class C, Class D, and Class E. The octets in the IP addresses are used to create the IP classes. IP classes are assigned by the Internetwork Information Center, or InterNIC.

Each IP class has different criteria for its usage as listed in the following:

- Class A: Used for very large networks.
- Class B: Used for medium-size networks.
- Class C: Used for small- to medium-size businesses.
- Class D: Used for multicasts. (e.g., Cisco router sending an update to all other Cisco routers.)
- Class E: Used for experimental purposes.

<i>Class and Subnet Mask</i>	<i>Description</i>
Class A 255.0.0.0	<p>Class A subnet masks provide a small number of network addresses for networks with a large number of nodes per network.</p> <ul style="list-style-type: none">• Number of nodes per network: 16,777,214• Network ID portion: First octet• Node ID portion: Last three octets <p>Class A addresses are used only by extremely large networks. Large telephone companies and ISPs leased most Class A network addresses early in the development of the Internet.</p>
Class B 255.255.0.0	<p>Class B subnet masks offer a larger number of network addresses, each with fewer nodes per network.</p> <ul style="list-style-type: none">• Number of nodes per network: 65,534• Network ID portion: First two octets• Node ID portion: Last two octets <p>Most companies leased Class B addresses to use them on Internet- connected networks. In the beginning, there were plenty of Class B addresses to go around, but now there are a few.</p>
Class C 255.255.255.0	<p>Class C subnet masks offer a large number of network addresses for networks with a small number of nodes per network.</p> <ul style="list-style-type: none">• Number of nodes per network: 254• Network ID portion: First three octets• Node ID portion: Last octet <p>Because there can be more Class C networks than any other type, they are the only addresses still available.</p>

Classless Addressing

Changes in the Internet, since the early 90s, have rendered classful addresses obsolete. One of the final remnants of classful addressing is the use of the terms Class A, Class B, and Class C to describe common subnet masks. These traditional IP address classes have limitations on the number of available addresses in each class, there are now various implementations that utilize classless addressing. In these schemes, there is no strict dividing line between groups of addresses, and the network address or node address division is determined entirely by the number of 1 bits in the subnet mask.

CIDR Notation

Classless Inter-Domain Routing (CIDR) is a classless addressing method that considers an IP address as a 32-bit binary word. Mask bits can move in one-bit increments to provide the exact number of nodes and networks required. The CIDR notation combines a network address with a number to represent the number of one bits in the mask. With CIDR, multiple class-based networks can be represented as a single block.

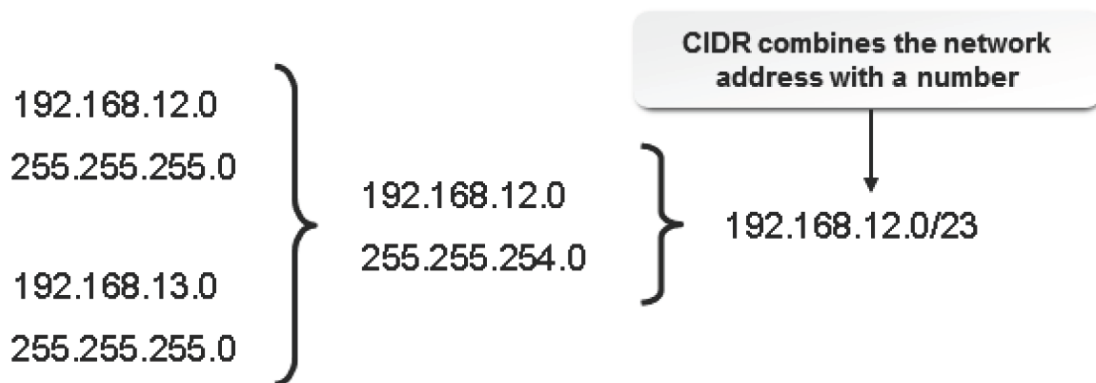


Figure 12-5: A classless addressing method that considers a VLSM as a 32-bit binary word.



Note: CIDR can also be referred to as classless routing or supernetting. Because of its efficiencies, CIDR has been rapidly adopted, and the Internet today is largely a classless address space.

CIDR Subnet Masks

There are different values possible for each CIDR subnet mask. The /24, /16, and /8 CIDR masks correspond with the classful ranges of Class C, Class B, and Class A, respectively.

CIDR Mask (Number of Network Bits)	Number of Possible Nodes	Standard Subnet Mask in Dotted Decimal
/32	N/A	255.255.255.255
/31	N/A	255.255.255.254
/30	2	255.255.255.252
/29	6	255.255.255.248
/28	14	255.255.255.240
/27	30	255.255.255.224
/26	62	255.255.255.192
/25	126	255.255.255.128
/24	254	255.255.255.0 (Class C)
/23	510	255.255.254.0
/22	1,022	255.255.252.0
/21	2,046	255.255.248.0
/20	4,094	255.255.240.0
/19	8,190	255.255.224.0
/18	16,382	255.255.192.0
/17	32,766	255.255.128.0
/16	65,534	255.255.0.0 (Class B)
/15	131,070	255.254.0.0

CIDR Mask (Number of Network Bits)	Number of Possible Nodes	Standard Subnet Mask in Dotted Decimal
/14	262,142	255.252.0.0
/13	524,286	255.248.0.0
/12	1,048,574	255.240.0.0
/11	2,097,150	255.224.0.0
/10	4,194,304	255.192.0.0
/9	8,386,606	255.128.0.0
/8	16,777,214	255.0.0.0 (Class A)
/7	33,554,430	254.0.0.0
/6	67,108,862	252.0.0.0
/5	134,217,726	248.0.0.0
/4	268,435,544	240.0.0.0
/3	536,870,910	224.0.0.0
/2	1,073,741,824	192.0.0.0
/1	N/A	N/A

A CIDR Application

The CIDR address 192.168.12.0/23 applies the network mask 255.255.254.0 to the 192.168.0.0 network, starting at 192.168.12.0. On a modern router, this single routing entry can define a supernet that includes the address range from 192.168.12.0 to 192.168.13.255. Compare this to traditional class-based networking, where this range of addresses would require separate routing entries for each of two Class C networks—192.168.12.0 and 192.168.13.0—each using the default Class C subnet mask of 255.255.255.0.

Broadcast Addresses

A **broadcast address** is a special IP address that is used to send messages to all hosts with the same network address. On IP networks, the general broadcast address is 255.255.255.255.

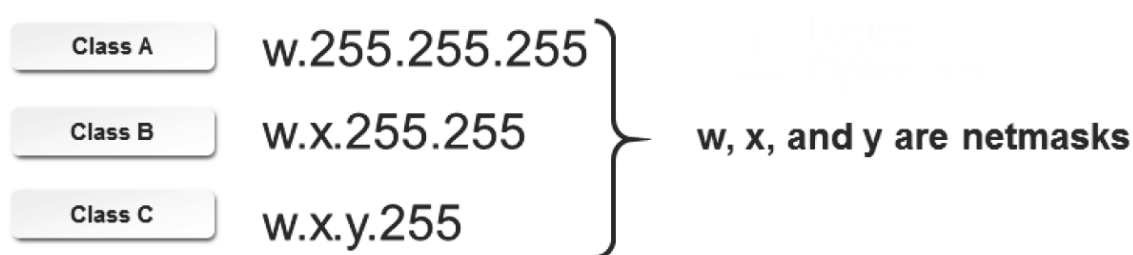


Figure 12-6: Broadcast address formats for the three classes.

Format of Broadcast Addresses

A broadcast address may vary depending on the class of the IP address. The following table lists the format of different broadcast addresses for different classes of octets w.x.y.z.

Class	Subnet Mask	CIDR Mask	Netmask/ Network Part	Host Part	Broadcast Address
A	255.0.0.0	/8	w	x.y.z	w.255.255.255
B	255.255.0.0	/16	w.x	y.z	w.x.255.255
C	255.255.255.0	/24	w.x.y	z	w.x.y.255

Ports

On a network, a **port** is an access point to a logical connection. It serves as a channel through which information can be exchanged directly among networked computers. Many ports can operate simultaneously on a computer to provide services to different applications. A unique port number identifies the type of application that is sending or receiving data. It also informs the computer as to which application program running on the computer should process the data that is being sent or received through a particular port. Ports are identified by numbers between 0 and 65,536.

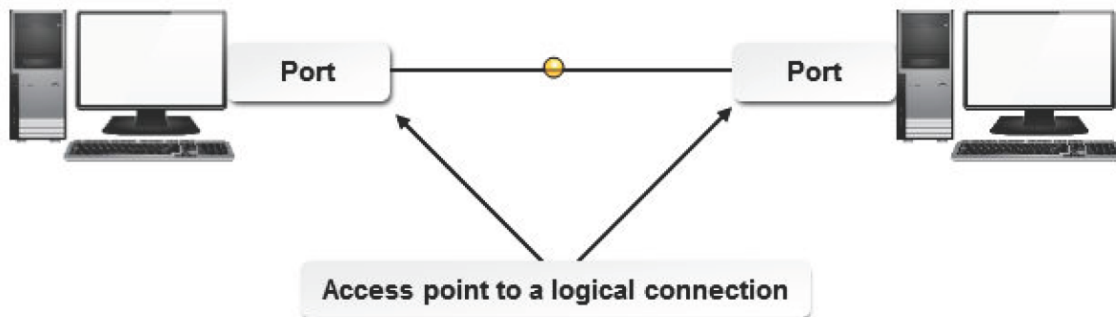


Figure 12-7: Transfer of data through ports on a network.

Allocation of Port Numbers

Just as several people live at the same address, such as in an apartment, multiple network applications may reside at the same IP address. In an apartment, suite numbers may be used in conjunction with the street address to identify which occupant should receive a piece of mail.

Similarly, the IP address along with a specific port number is allocated for different applications.

There is a scheme for identifying specific applications that share an IP address—and that is the addition of a port to the IP address. For example, a web server and an FTP server may both run on the same server, at 24.95.112.13. Web servers typically are set up to run on port 80, and FTP servers run on port 21. To identify the FTP server, you could use the address 24.95.112.13:21. The colon character separates the port address from the rest of the IP address.

Most servers enable the administrator to specify the port on which a service should run. The ability to specify the port number can be useful when multiple services, such as two web servers, are running on the same computer. One server may run on port 80 and the other on port 81.

Ports Allocated for Different Services

Ports can be allocated to different services based on the types of applications supported by a network. The **/etc/services** file contains a list of ports supported in Linux. Some of the common ports are listed in the following table.

Port Number	Description
1	TCP Port Service Multiplexer (TCPMUX)
5	Remote Job Entry (RJE)
7	ECHO
18	Message Send Protocol (MSP)
20	File Transfer [Default Data] (FTP–Data)
21	File Transfer [Control] (FTP–Control)
22	Secure Shell Login (SSH)

<i>Port Number</i>	<i>Description</i>
23	Telnet
25	Simple Mail Transfer Protocol (SMTP)
29	MSG ICP
37	Time
42	Host Name Server
43	Whols
49	Login Host Protocol
53	Domain Name System (DNS)
69	Trivial File Transfer Protocol (TFTP)
70	Gopher Services
79	Finger
80	HTTP
103	X.400 Standard
108	SNA Gateway Access Server
109	POP2
110	POP3
115	Simple File Transfer Protocol (SFTP)
118	SQL Services
119	Network News Transfer Protocol (NNTP)
123	Network Time Protocol
137	NetBIOS Name Service
139	NetBIOS Datagram Service
143	Interim Mail Access Protocol (IMAP)
150	NetBIOS Session Service
156	SQL Server
161	SNMP
179	Border Gateway Protocol (BGP)
190	Gateway Access Control Protocol (GACP)
194	Internet Relay Chat (IRC)
197	Directory Location Service (DLS)
389	Lightweight Directory Access Protocol (LDAP)
396	Novell Netware over IP
443	HTTPS
444	Simple Network Paging Protocol (SNPP)
465	Secure SMTP over TLS
546	DHCP Client
547	DHCP Server
631	Internet Printing Protocol (IPP)
1080	Socket Secure (SOCKS)
3306	MySQL

Port Ranges

The Internet Assigned Numbers Authority (IANA), an international agency, separates port numbers into three blocks: well-known ports, which are preassigned to system processes; registered ports, which are available to user processes and are listed as a convenience; and dynamic ports, which are assigned by a client operating system when there is a request for service.

<i>Block</i>	<i>Description</i>
Well-known ports	Port range: 0 to 1,023. These ports are preassigned for use by common, or well-known, services. Often, the services that run on these ports must be started by a privileged user. Services in this range include HTTP on TCP port 80, IMAP on TCP port 143, and DNS on UDP port 53.
Registered ports	Port range: 1,024 to 49,151. These ports are registered by software makers for use by specific applications and services that are not as well-known as the services in the well-known range. Services in the registered port range include SOCKS proxy on TCP port 1080, Kazaa peer-to-peer file sharing on TCP port 1214, and Xbox Live on TCP and UDP port 3074.
Dynamic or private ports	Port range: 49,152 to 65,535. These ports are set aside for use by unregistered services and by services that need a temporary connection.

Network Interfaces

A **network interface** is a point of connection between two systems. It can be implemented using hardware or software. Different types of network interfaces are available.

<i>Network Interface Type</i>	<i>Description</i>
Physical	A network interface that is implemented using a hardware device. For example, an Ethernet interface (denoted by ethX, where X refers to the number of the interface) is set up using a Network Interface Card (NIC).
Virtual	A network interface that is implemented through software support. For example, a loopback interface (lo) simulates a network interface without the help of a physical device. It is used to test network connectivity and accuracy of data transmission by sending data back to the generating source address.

NICs

A **Network Interface Card (NIC)** is a small circuit board that enables a computer to connect to a network. A network interface is created between two or more computers using NICs. To connect to different networks—such as a wired or a wireless network—more than one NIC can also be installed on a computer. The different NICs connected to a system are numbered. A NIC is usually an internal or external adapter card that is installed into one of the system's expansion slots. NICs can even be built into the motherboard of the system, or connected through a USB port. After the NIC is installed, it has to be configured to connect to a particular network using the required network address and settings.

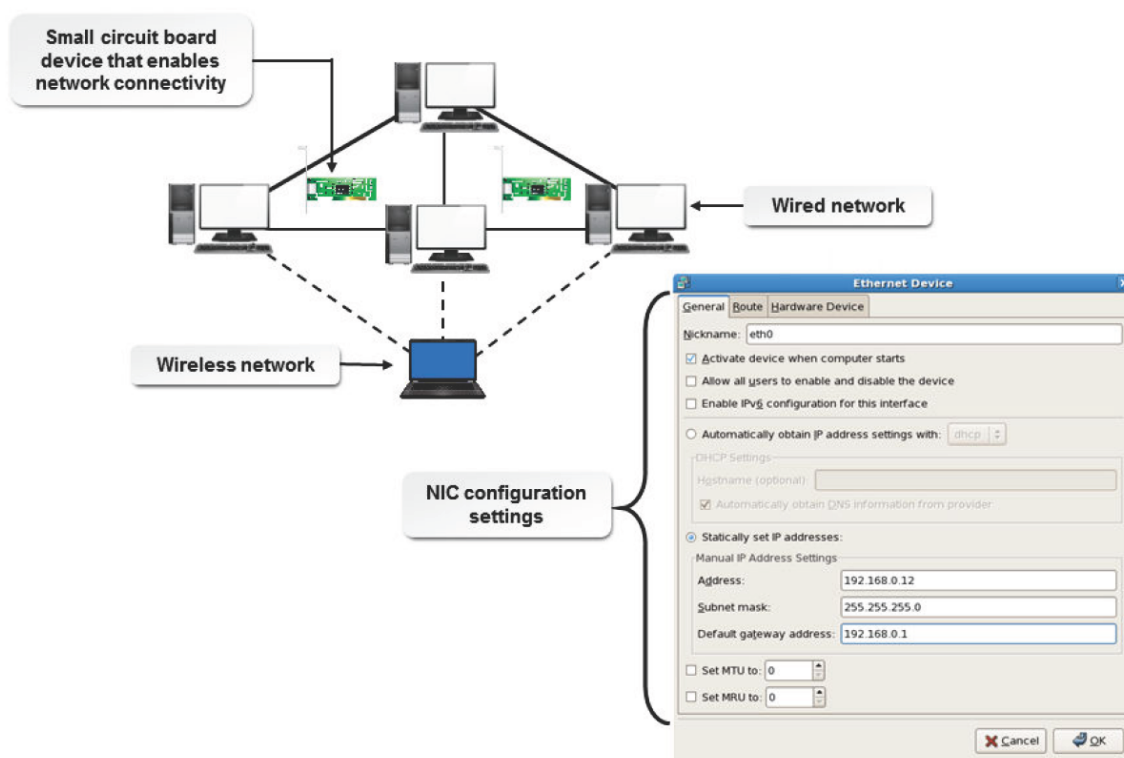


Figure 12-8: A NIC helps to connect computers and other devices on a network.

NIC Characteristics

NICs have some special characteristics that distinguish them from other types of adapter cards.

Characteristic	Description
Network connection port	Network adapter cards will have one or more ports that are configured to connect specifically to a given type of network cable. Some older cards had several types of ports so that they could connect to several different types of network cable. Network connections today are standardized and almost all use one port type.
Physical network address	Each network adapter has a globally unique physical address burned onto the card by the card manufacturer. The physical address uniquely identifies every individual card that connects to the network cable or media. For this reason, the physical address is also called the Media Access Control (MAC) address . MAC addresses are 6 bytes long. A typical MAC address may appear as 00-00-86-47-F6-65 , where the first 3 bytes are the vendor's unique ID and the next 3 bytes uniquely identify that card's vendor.
Status indicator lights	<p>Network adapters, including those built into most network devices, typically have one or more status indicator lights that can provide information on the state of the network connection.</p> <ul style="list-style-type: none"> Most adapters have a link light that indicates if there is a signal from the network. If the link light is not lit, there is a problem with the cable or the physical connection. Most adapters also have an activity light that flickers when packets are received or sent. If the light flickers constantly, the network may be overused or there may be a device generating network noise. Some multi-speed adapters have a speed light to show whether the adapter is operating at 10 Mb/s (Ethernet), 100 Mb/s (Fast Ethernet), or 1,000 Mb/s (Gigabit Ethernet). Some types of equipment combine the functions of more than one light into dual color LEDs (Light Emitting Diode). For example, a green flickering light may indicate normal activity, while an orange flickering light may indicate network traffic collisions.

The ifconfig Command

The **ifconfig** command is used for configuring network interfaces for Linux servers and workstations. It is also used to view the current TCP/IP configuration of a system, including the IP address and the netmask address.

```

root@localhost:~
File Edit View Terminal Tabs Help

[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:C6:ED:E0
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fec6:ede0/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5225659 errors:10 dropped:0 overruns:0 frame:0
          TX packets:1426965 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3301159228 (3.0 GiB)  TX bytes:116861331 (111.4 MiB)
          Interrupt:177 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1358 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1358 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3172706 (3.0 MiB)  TX bytes:3172706 (3.0 MiB)

[root@localhost ~]#

```

Figure 12-9: The output of the ifconfig command.

Syntax

The syntax of the ifconfig command is `ifconfig {interface name} {options | address}`.

ifconfig Command Options

The ifconfig command has many options. The most frequently used options are provided in the following table.

Option	Function
up	Activates the interface.
down	Deactivates the interface.
{address}	Sets the IP address.
netmask {address}	Sets the network mask for the interface.
dstaddr {address}	Sets the remote IP address.

The ifconfig Command Interface

Linux provides an interface to the ifconfig command that makes configuration of a network device very simple. This interface is made up of the ifup and ifdown commands and two or more configuration files. The configuration files are the `/etc/sysconfig/network` file, which specifies the network configuration, and one or more files in the `/etc/sysconfig/network-scripts` directory, which contain device-specific networking information. For a system with a single Ethernet card, device-specific information is stored in the `/etc/sysconfig/network-scripts/ifcfg-eth0` file.

Note: In older versions of Linux, the `/etc/HOSTNAME` or `/etc/hostname` file was used instead of the `/etc/sysconfig/network` file.

The ifup and ifdown commands are used to start and stop specific network devices, respectively.

The syntax of these commands is `[command] {device name}`, where `[command]` is either ifup or ifdown and `{device name}` is the name of the device such as eth0, eth1, and so on.

Because the ifup and ifdown commands control only a single network device, it is often easier to use the `/etc/init.d/network` command with the start or stop parameters. This command starts (or stops) all network devices

simultaneously.

The ip Command

The **ip** command is used for configuring network interfaces for Linux servers and is a more powerful replacement for the older *ifconfig* command. It is also used to view the current TCP/IP configuration of a system, including the IP address and the netmask address.

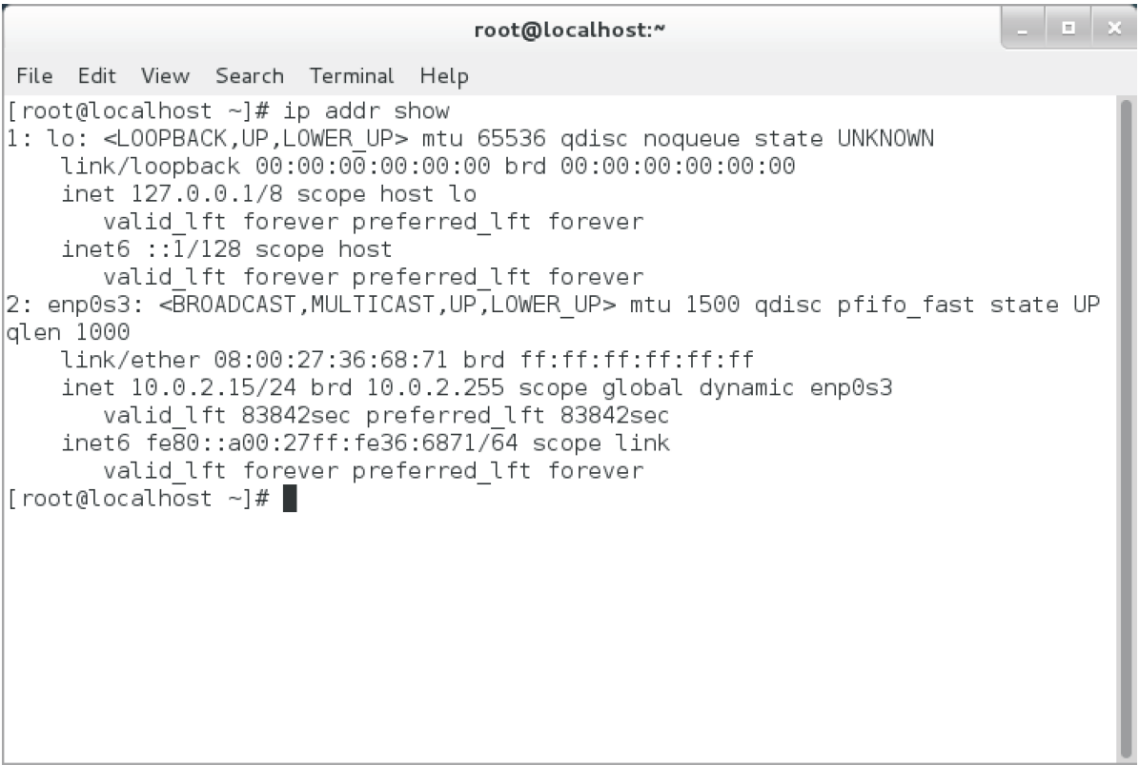


Figure 12-10: The output of the ip command.

Syntax

The syntax of the ip command is `ip {options} {interface name} {command}`.

ip Command Usage

The ip command has many options. The most frequently used commands as compared to their *ifconfig* and *route* versions are provided in the following table.

<i>ip Command</i>	<i>ifconfig/route Command Equivalent</i>
ip addr show	ifconfig
ip link show	ifconfig
ip link set eth0 up	ifconfig eth0 up
ip link set eth0 down	ifconfig eth0 down
ip addr address 192.168.0.50 dev eth0	ifconfig eth0 192.168.0.50
ip addr add 192.168.0.50/24 broadcast 192.168.0.255 dev 0	ifconfig eth0 192.168.0.50 netmask 255.255.255.0 broadcast 192.168.0.255
ip addr del 192.168.0.50/24 dev eth0	N/A
ip addr add 192.168.0.50/24 dev eth0 label eth0:1	ifconfig eth0:1 192.168.0.50/24
ip route show	route -n
ip route replace default via 192.168.0.254	route add default gw 192.168.0.254

The iwconfig Command

The **iwconfig** command is used for configuring wireless network interfaces for Linux servers and workstations. It is similar to the ifconfig command, except that it is used to set up and view the parameters of wireless network interfaces.

Syntax

The syntax of the iwconfig command is `iwconfig {interface name} {options | address}`.

The iwconfig Command Options

The iwconfig command has various options, which are provided in the table.

<i>Option</i>	<i>Function</i>
essid	Sets the ESSID, also called network name or domain ID, which is used to identify cells that are part of the same virtual network.
nwid/domain	Sets the network ID, which differentiates the wireless network from other networks and identifies nodes belonging to the same cell.
nick	Sets the nickname or the station name that is used by some wireless tools.
mode	Sets the operating mode of the device.
freq/channel	Sets the operating frequency or channel of the device.
ap	Registers the access point given by the address.
rate/bit	Sets the bit-rate.
txpower	Sets the transmit power.
sens	Sets the sensitivity threshold.
retry	Sets the maximum number of retries.
rts	Sets the size of the smallest packet for which the node sends Request To Send (RTS).
frag	Sets the maximum size for fragments that can be transferred.
key/enc	Sets the encryption or scrambling keys and security mode.
power	Sets power management parameters.
commit	Applies all pending changes.

Cells

A cell is a network zone covered under a tower or access point.

RTS

Request To Send (RTS) is a signal sent by a communication device to a receiving device, to verify if the receiving device is ready to accept the data that is to be sent to it. For example, a modem sends an RTS to a computer before it transmits data.

Subnets

Subnets are used in large organizations, such as universities and corporations, where it is necessary to divide the network into smaller, more manageable segments. Subnets are logical subsections of a large network. Each segment requires its own network address and host identifiers and is treated as a subnet of the original network.

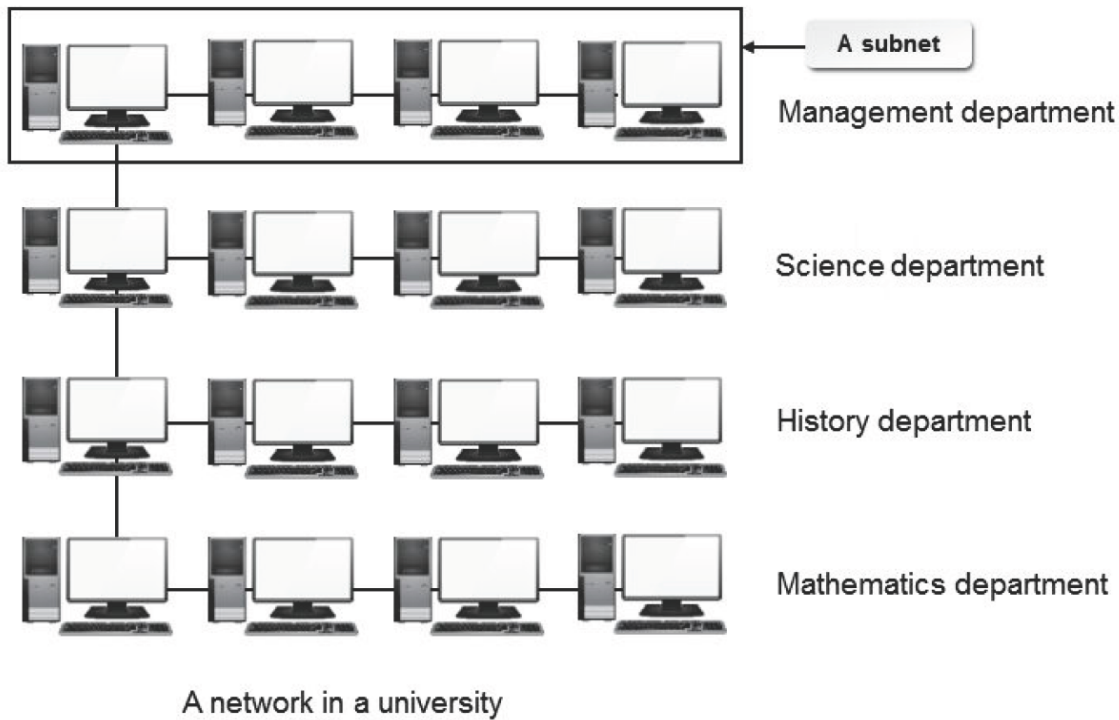


Figure 12-11: A subnet on a university network.

BOOTPROTO

BOOTPROTO is a variable that is used to specify the mode in which a NIC is configured. If **BOOTPROTO=static**, then the NIC will be configured manually. If **BOOTPROTO=dhcp**, then the NIC will contact the DHCP server to obtain the IP information.

The ping Command Options

The ping command is used to test network connectivity. A few frequently used ping command options are described in the following table.

<i>If You Need To</i>	<i>Use This ping Option</i>
Ping the destination IP address for a specified number of times.	-c {number}
Ping the destination IP address at regular intervals.	-i {number}
Broadcast the ICMP packages for a specified network.	-b {broadcast ID}

The ping Command Restriction

Because the ping command generates ICMP traffic to the target hosts, a firewall can be set to block this traffic. Therefore, the ping command should be relied upon only when there are no known firewalls blocking the ICMP traffic.

How to Connect to a Network

Follow these general procedures to connect to a network.

Manually Configure a NIC

To manually configure a NIC:

1. Log in to the CLI as **root**.

2. To stop the **network** service, enter `service network stop`.
3. Enter `cd /etc/sysconfig/network-scripts`.



Note: The `/etc/sysconfig/network-scripts/` directory contains various network scripts, such as `ifcfg-eth0`, that will be executed whenever a system starts up.

4. Enter `vi ifcfg-{device name}`.
5. Make the necessary changes in the file.
 - To change the booting protocol to static, change `BOOTPROTO=dhcp` to `BOOTPROTO=static` and press **Enter**.
 - To specify the IP address of the NIC, enter `IPADDR={IP address}`.
 - To specify the netmask address, type `NETMASK={netmask}`.
6. Save and close the file.
7. To start the **network** service, enter `service network start`.

Automatically Configure a NIC

To automatically configure a NIC:

1. Ensure that the DHCP server is configured and running. If necessary, check with your system administrator.



Note: A DHCP server automatically allocates IP addresses to a client.

2. Log in to the CLI as **root**.
3. Enter `cd /etc/sysconfig/network-scripts`.
4. To open the device file, enter `vi ifcfg-{device name}`.
5. To change the booting protocol to dhcp, modify `BOOTPROTO=static` to `BOOTPROTO=dhcp`.
6. Save and close the file.
7. To restart the **network** service, enter `service network restart`.
8. If necessary, to check the network connectivity, enter `ping -{options} {destination IP address}`.

Set a Temporary IP Address and Establish a Temporary Connection with Other Networks

To set a temporary IP address and establish a temporary connection with other networks:

1. Log in to the CLI as **root**.
2. To view the status of the active devices, enter `service network status`.
3. To view the details of all the configured devices, enter `ifconfig`.
4. To set a temporary IP address to the specified NIC device, enter `ifconfig -a eth{number} {new IP address}`.

5. If necessary, to revert to the original IP address, enter service network restart.

Add a New NIC

To add a new NIC:

1. Install the NIC.
 - a. Shut down the system and disconnect it from the power source.
 - b. Remove the access panel on the system case.
 - c. Insert the NIC card in a free PCI slot.
 - d. Close the access panel on the system case.
 - e. Connect the network cable to the NIC.
 - f. Connect the system to the power source and restart the system.
2. Log in to the CLI as **root**.
3. To navigate to the **network-scripts** directory, enter `cd /etc/sysconfig/network-scripts`.
4. To create a blank file, enter `touch ifcfg-eth{device name}`.
5. Enter `vi ifcfg-eth{device name}`.
6. To specify the device name, enter `DEVICE=eth{device name}`.
7. To automatically activate the device when the system starts, enter `ONBOOT=yes`.
8. Enter `BOOTPROTO=static`.
9. Enter `IPADDR={IP address}`.
10. Type `NETMASK={netmask}`.
11. Save and close the file.
12. To restart the **network** service, enter service network restart.

Disable a NIC

To disable a NIC:

1. Log in to the CLI as **root**.
2. Disable the NIC.
 - Enter `cd /etc/sysconfig/network-scripts` and delete the corresponding `ifcfg-eth{number}` file in the directory.
 - Enter `ifdown eth{number}`.



Note: To enable the NIC, the `ifup eth{number}` command is given.

Delete a NIC

To delete any NIC device:

- Delete the corresponding `ifcfg-eth{number}` file located in the `/etc/sysconfig/network-scripts/` directory.



Note: A NIC does not have to be physically removed from a system to delete it.

Turn Off Network Services Not in Use

To turn off network services not in use:

1. Log in to the CLI or GUI as **root**.
2. To stop unnecessary network services, enter `service {network service name} stop` or `systemctl stop {network service name}`.
3. If necessary, to disable a network service at startup, enter `chkconfig {network service name} off` or `systemctl disable {network service name}`.

TOPIC B Configure Routes

In the previous topic, you configured the IP settings for network interfaces. Routing allows you to manage data transmission traffic on networks. It enables data to be transmitted from a source to its destination through different routes. In this topic, you will configure routes.

Computers on a network interact with each other simultaneously at numerous instances. If one computer on a network communicates with many computers at the same time, and if the data transmission routes or communication paths are not configured, it may lead to a system crash due to flooding of information. Therefore, the routes for information transmission have to be configured to avoid collision in network traffic.

Routers

A **router** is a networking device that connects multiple networks. Routers enable data to be exchanged among networks by examining and determining the best network path for data to travel.

A router can be a dedicated device or can be implemented as a software application running on a network enabling device.

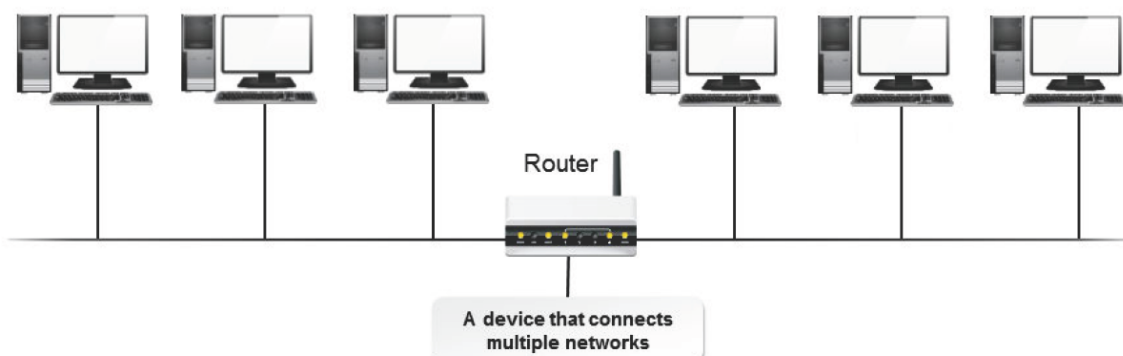


Figure 12-12: A router connecting two networks.

Routing

Routing is the process of selecting the best route for moving data packets from a source to its destination on a network. To assist the process of routing, a router applies appropriate algorithms to generate and maintain an information base about network paths. It considers various metrics, such as the path bandwidth, path reliability, and communication costs, while evaluating the available network paths to determine the optimal route for forwarding a packet. Once the optimal route for a packet is assigned, packet switching is done to transfer the packet from the source host to the destination host.

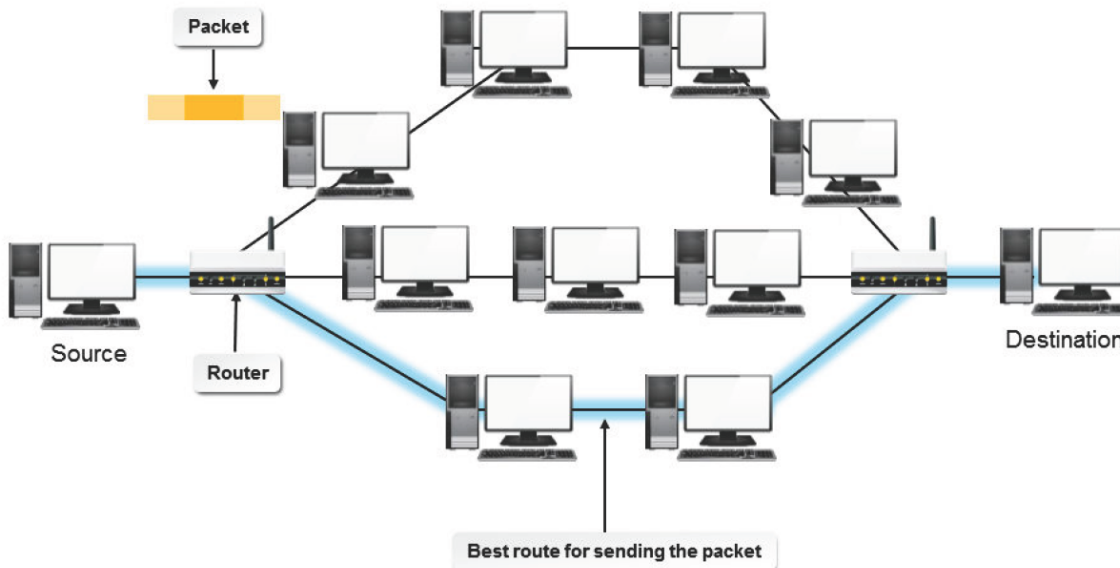


Figure 12-13: Example of routing on a network.

Packets

A **packet** is a formatted unit of data being sent across a network. In addition to the user data, it comprises control information, such as the source and destination addresses, which are required to deliver the user data. A packet is also known as a **datagram**.

Packet-Switching Technology

Packet-switching technology is used for transmitting data among computers on a network. In a packet-switched network, a message is broken into packets, which are transmitted individually or switched to their required destination. During the process, each packet may follow a different path, but at the destination, the packets are reassembled to form the original message sent from the main destination. This technology ensures greater routing and transporting efficiency on a network. The Internet is a packet-switched network.

Benefits of Packet-Switched Networking

The benefits of a packet-switched network lie in the underlying technology of dividing a message to be sent over the Internet into packets. When data is transported in packets rather than in one big stream of data, the packets do not all have to move through the same path. Because the data is broken up into small packets, the packets can be sent across the Internet over various paths, eventually (in a fraction of a second) reaching their destination, where the packets can be reassembled into the original data. This means that one or more of the smaller networks, which make up the Internet, can go out of service without preventing the packets from ultimately reaching their destination, because the packets can simply take a different path to get there. If a few packets never reach their destination, they can be resent over a different path.

If files were not broken up into smaller packets, the entire file will have to be resent if any part of it did not reach the destination intact. Having multiple paths and breaking up files into small packets increases the reliability of the network.

The ip Command

The ip command is used to show or manipulate routing, policy routing, devices, and tunnels. The syntax of the ip command is ip [options] {object} {command | help}.

Routing Tables

Routers exchange information with each other by building a table of network addresses. This information base is called a *routing table*. Routers refer to this table to determine where to forward the packets. If a router that is attached to four networks receives a packet from one of these networks, it will determine which of the other three networks is the best route to send the packet so that it could reach its destination quickly.

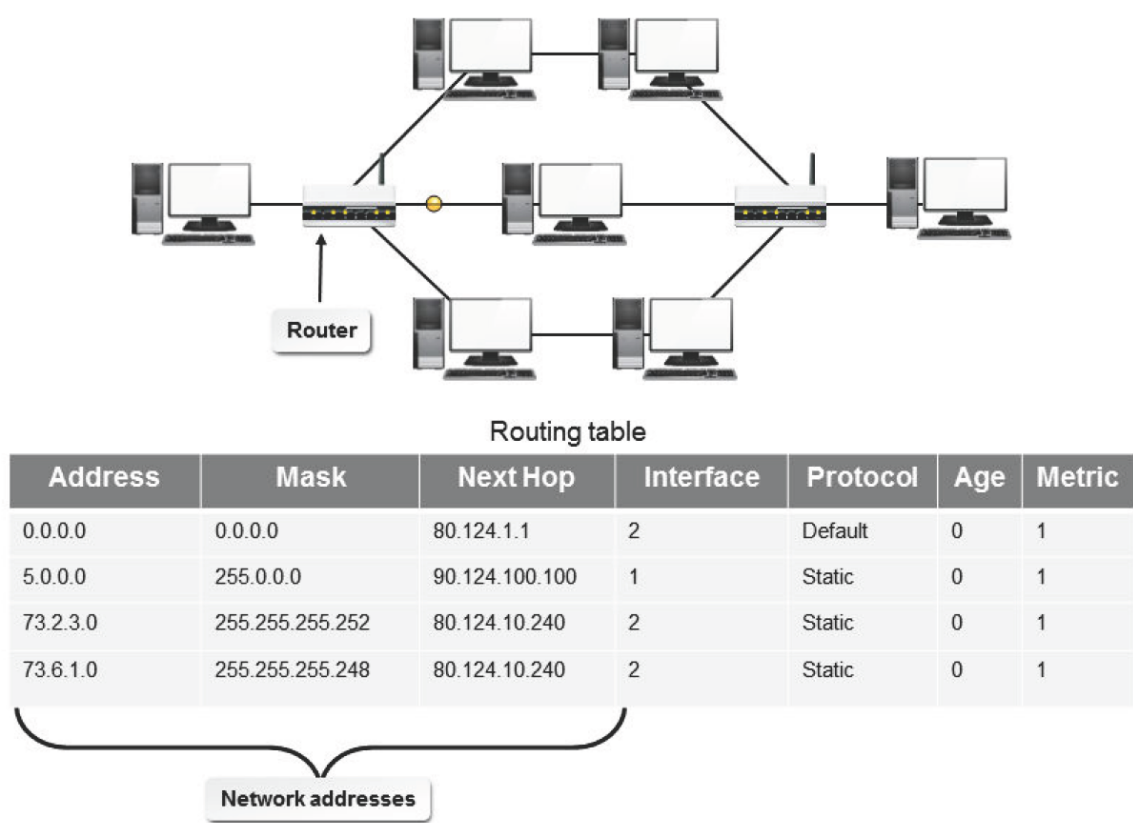


Figure 12-14: A routing table that comprises network addresses.

The route Command

The route command manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks. When the add or del option is used, the route command modifies the routing tables. Without these options, the route command displays the contents of the routing tables.

```

root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# route
Kernel IP routing table
Destination    Gateway      Genmask      Flags Metric Ref    Use Iface
192.168.0.0    *           255.255.255.0 U        0      0      0 eth0
[root@localhost ~]#
  
```

Contents of the routing table

Figure 12-15: The output of the route command.

Routing Examples

The following table displays a few routing examples.

Command	Description
route add -net 127.0.0.0	Adds the normal loopback entry using netmask 255.0.0.0 (class A net, determined from the destination address) and associated with the loopback device, assuming this device was previously set up correctly with ifconfig.
route add -net 192.56.76.0 netmask 255.255.255.0 dev eth0	Adds a route to the network 192.56.76.x via eth0. The Class C netmask modifier is not really necessary here because 192.* is a Class C IP address. The word "dev" can be omitted here.
route add default gw mango-gw	Adds a default route, which will be used if no other route matches. All packets using this route will be gatewayed through mango-gw. The device that will actually be used for that route depends on how mango-gw can be reached—the static route to mango-gw will have to be set up before.
route add -net 192.57.66.0 netmask 255.255.255.0 gw ipx4	Adds the net 192.57.66.x to be gatewayed through the former route to the SLIP (Serial Line Internet Protocol) interface.
route add -net 10.0.0.0 netmask 255.0.0.0 reject	Installs a rejecting route for the private network 10.x.x.x.

Gateways

A **gateway** is a device, software application, or system that converts data between incompatible systems. Gateways can translate data among different operating systems, email formats, or networks.

It can link two dissimilar networks, which operate on varying protocols, enabling them to communicate with each other and exchange information.

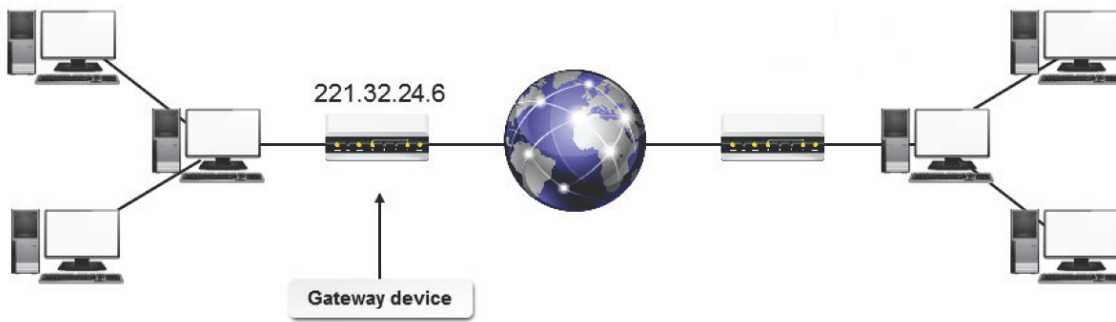


Figure 12-16: A gateway device linking dissimilar networks.

The **default gateway** is the gateway that acts as a network segment's access point to all other external networks and the Internet. The IP address assigned to the default gateway router is called the **default gateway address**. It is important because this address is configured as the access point to all the computers on that network segment. It provides an access path for packets in and out of the network segment.

The traceroute Command

The traceroute command is used to print the route that packets take to reach their destination.

This is useful in troubleshooting some network or Internet connectivity problems. There are several options for the traceroute command.

<i>Option</i>	<i>Description</i>
-d	Sets the socket level debug option.
-n	Prints hop addresses numerically.
-i {interface}	Specifies the interface through which traceroute should send packets.
-g {gateway}	Specifies a source route gateway.
-r	Bypasses the normal routing tables and sends the packets directly to a host on an attached network.
-w {waittime}	Sets the time, in seconds, to wait for a response to a probe.



Note: Like traceroute, tcpdump is another network monitoring package that can be installed on a Linux system.

The netstat Command

The netstat command displays statistics about a network, including socket status, interfaces that are auto-configured, memory statistics, and routing tables. With no arguments, the default netstat command displays open sockets. Some of the frequently used netstat command options are described in the following table.

<i>Option</i>	<i>Displays</i>
-r or --route	The kernel routing tables.
-g or --groups	The multicast group membership.
-i or --interface or --interface={iface}	A table of all network interfaces or the specified interface.
-M or --masquerade	A list of masqueraded connections.
-s or --statistics	A summary of statistics for each protocol.
-e or --extend	Additional details.

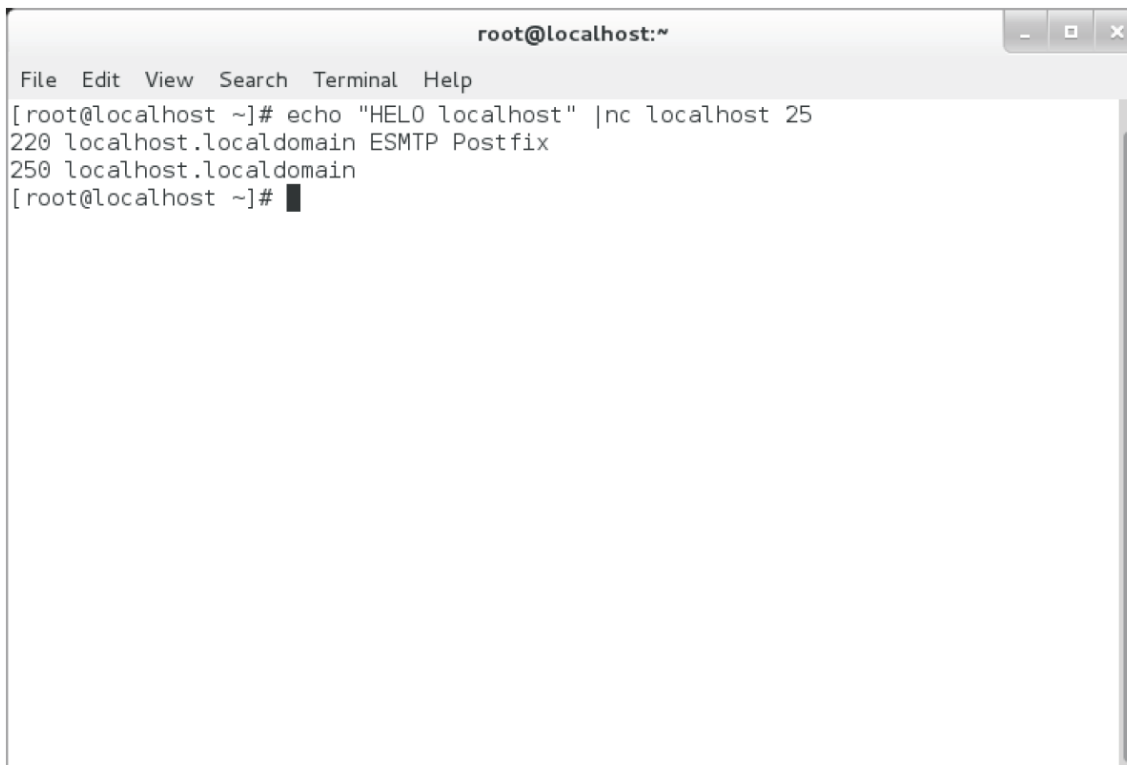
```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# netstat -s  
Ip:  
  28744 total packets received  
  14 with invalid addresses  
  0 forwarded  
  0 incoming packets discarded  
 28717 incoming packets delivered  
 1340 requests sent out  
Icmp:  
 129 ICMP messages received  
 0 input ICMP message failed.  
ICMP input histogram:  
  destination unreachable: 1  
  echo requests: 128  
 136 ICMP messages sent  
 0 ICMP messages failed  
ICMP output histogram:  
  destination unreachable: 8  
  echo replies: 128  
IcmpMsg:  
  InType3: 1  
  InType8: 128  
  OutType0: 128  
  OutType3: 8
```

Figure 12-17: The *netstat* command displaying a summary of statistics for each protocol.

The netcat Command

The GNU netcat command is a networking utility which reads and writes data across network connections using the TCP or UDP protocol. It is often used to test or automate network services and configurations from the command-line or via scripts. Some of the frequently used ncat command options are described in the following table.

Option	Displays
-l {port} or --listen {port}	Listen for connections on <i>port</i> , rather than connecting to a remote system.
-o {file} or --output {file}	Save session data to a file.
-v or --verbose	Verbose output to display useful connection- based details.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command '[root@localhost ~]# echo "HELO localhost" | nc localhost 25' and its output: '220 localhost.localdomain ESMTP Postfix' and '250 localhost.localdomain'. The prompt '[root@localhost ~]#' is shown again at the end.

```
root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# echo "HELO localhost" | nc localhost 25
220 localhost.localdomain ESMTP Postfix
250 localhost.localdomain
[root@localhost ~]#
```

Figure 12-18: The netcat command (also known as "nc") displaying the results of a SMTP protocol test.

How to Configure Routes

Follow these general procedures to configure routes.

Configure a Route for the IPv4 Address

To configure a route for the IPv4 address:

1. Log in to the CLI as **root**.
2. To add a static route, enter `ip route add {network part of IPv4 address}/{length} via {gateway IPv4 address}`.
3. Make the static route persistent.
 - a. To stop the network interface service, enter `ifdown eth{device number}`.
 - b. To open the **network-scripts** directory, enter `cd /etc/sysconfig/network-scripts`.
 - c. To open the NIC configuration file, enter `vi route-eth{device number}`.
 - d. To configure the route, specify `{network part of IPv4 address}/{length} via {gateway IPv4 address}`.
 - e. Save and close the file.
 - f. To start the network interface service, enter `ifup eth{device number}`.
4. View the updated routing table.
 - Enter `route`.
 - Enter `netstat -r`.
 - Enter `ip route`.

Check the IPv4 Connectivity

To check the IPv4 connectivity:

1. Log in to the CLI as **root**.
2. Check the IPv4 connectivity.
 - To check the connectivity between the two systems, enter `ping [options] {IPv4 or hostname of destination system}`.
 - To view the network path to the destination system, enter `traceroute [options] {IPv4 or hostname of destination system}`.
 - To check the connectivity of the network path to the destination system, enter `mtr [options] {IPv4 or hostname of destination system}`.
 - To trace the network path and calculate the associated Maximum Transmission Unit (MTU) to the destination system, enter `tracepath [options] {IPv4 or hostname of destination system}`.



Note: mtr is a network diagnostic utility that combines the functionality of the traceroute and ping commands.

Configure the Default Gateway for the IPv4 Address

To configure the default gateway for the IPv4 address:

1. Log in to the CLI as **root**.
2. Configure the default gateway.
 - Configure the default gateway globally.
 - a. To open the **sysconfig** directory, enter `cd /etc/sysconfig`.
 - b. To open the network settings file, enter `vi network`.
 - c. Switch to insert mode.
 - d. To specify the IP address of the gateway, enter `GATEWAY={IPv4 address of the gateway system}`.
 - e. Save and close the file.
 - f. To stop the device, enter `ifdown eth{device number}`.
 - g. To start the device, enter `ifup eth{device number}`.
 - Configure the default gateway for each NIC.
 - a. To stop the network interface service, enter `ifdown eth{device number}`.
 - b. To open the **network-scripts** directory, enter `cd /etc/sysconfig/network-scripts`.
 - c. To open the NIC configuration file, enter `vi ifcfg-eth{device number}`.
 - d. To set the IP address of the gateway, specify `GATEWAY={IPv4 address of the gateway system}`.
 - e. Save and close the file.
 - f. To restart the network interface service, enter `ifup eth{device number}`.

Configure a Route for the IPv6 Address

To configure a route for the IPv6 address:

1. Log in to the CLI as **root**.
2. To add a static route, enter `ip -6 route add {network part of IPv6 address}/{length} via {gateway IPv6 address}`.
3. Make the static route persistent.
 - a. To stop the network interface service, enter `ifdown eth{device number}`.
 - b. To open the **network-scripts** directory, enter `cd /etc/sysconfig/network-scripts`.
 - c. To open the file containing route settings, enter `vi route6-eth{device number}`.
 - d. To configure the route, specify `{network part of IPv6 address}/{length} via {gateway IPv6 address}`.
 - e. Save and close the file.
 - f. To start the network interface service, enter `ifup eth{device number}`.
4. To view the updated routing table, enter `ip -6 route`.

Check the IPv6 Connectivity

To check the IPv6 connectivity:

1. Log in to the CLI as **root**.
2. Check the IPv6 connectivity.
 - To check the connectivity between the two systems, enter `ping6 [options] {IPv6 or hostname of destination system}`.
 - To view the network path to the destination system, enter `traceroute6 [options] {IPv6 or hostname of destination system}`.
 - To trace the network path and calculate the associated MTU to the destination system, enter `tracepath6 [options] {IPv6 or hostname of destination system}`.

Configure the Default Gateway for the IPv6 Address

To configure the default gateway for the IPv6 address:

1. Log in to the CLI as **root**.
2. To open the **sysconfig** directory, at the command prompt, enter `cd /etc/sysconfig`.
3. To open the network settings file, enter `vi network`.
4. Switch to insert mode.
5. To set the IP address of the gateway, specify `IPv6_DEFAULTGW={IPv6 address of the gateway system}`.
6. Save and close the file.

7. To stop the device, enter `ifdown eth{device number}`.
8. To start the device, enter `ifup eth{device number}`.

TOPIC C Configure Client Network Services

In the previous topic, you configured routers for transmission of data among computers on a network. The settings of the system determine the network resources that it can access. In this topic, you will configure network services on client systems.

On large networks, network administrators will have difficulty in assigning IP addresses to systems manually. It is easier for network users to remember system names instead of IP addresses. Without proper identification, a system will not have access to network resources. As a system administrator, you need to ensure that the IP address is properly assigned to the system and network users can use the system name to communicate with other network systems.

DHCP

The **Dynamic Host Control Protocol (DHCP)** allocates IP addresses on an as-needed basis to a client. Instead of using static IP addressing, DHCP leases a temporary IP address to the client for a specified period of time.

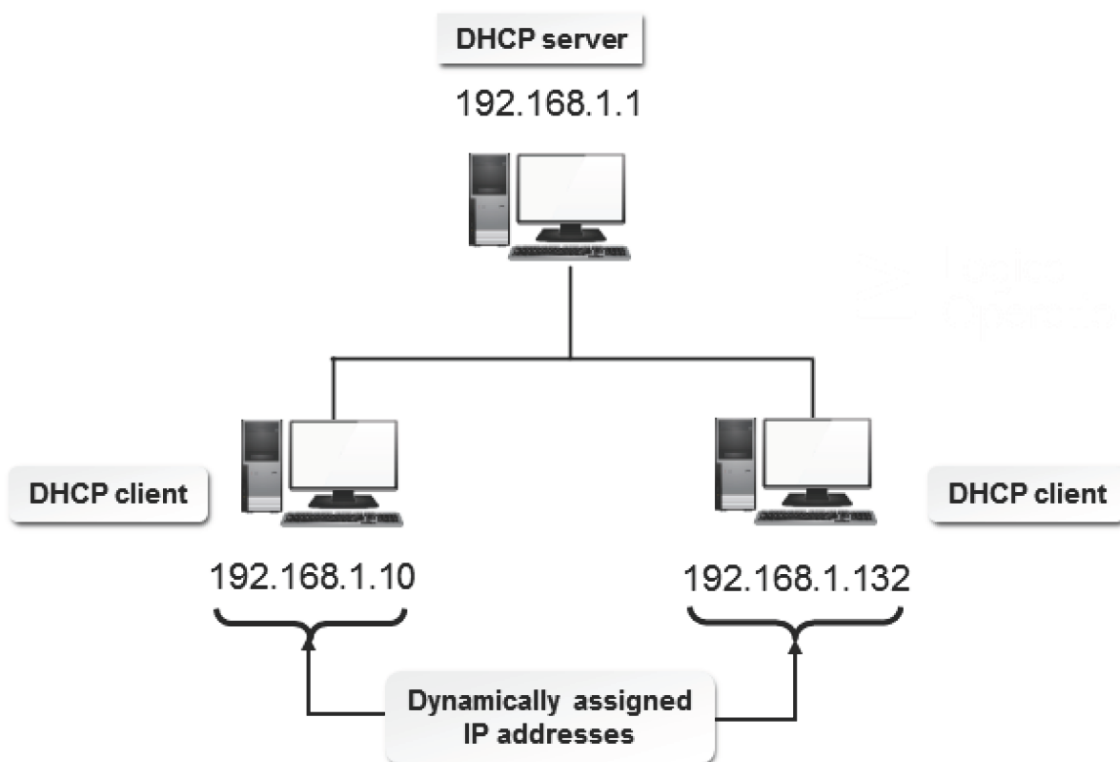


Figure 12-19: A DHCP server assigning IP addresses to clients.

A DHCP client is a system that gets network connectivity information from the DHCP server.

There are a variety of components in a DHCP implementation, but not all have to exist in every DHCP setup.

Component	Description
Options	Items, in addition to an IP address and subnet mask, that may be assigned to client systems, such as default gateways and DNS server addresses.
Scope	The range, or pool, of addresses for a given subnet that a DHCP server will assign.
Reservation	An option whereby a client consistently gets the same configuration information after every initialization.

Component	Description
Lease	The process of assigning an IP address and its associated options to a client for a finite or infinite period of time.

Allocation of IP Addresses

IP addresses can be allocated in three different ways.

Type of Allocation	Description
Manual	To allocate IP addresses manually, the administrator has to visit each host. Hosts can be either workstations or servers. Smaller organizations that have plenty of IP addresses for workstations and servers often use this method. When IP address assignments need to change, this method requires higher maintenance by the administrator.
Automatic	By using automatic allocation, the DHCP server assigns a permanent IP address to the host. The administrator need not visit each system. While this method reduces the amount of administrative time, it requires an adequate supply of IP addresses.
Dynamic	By using dynamic allocation, the DHCP server assigns a temporary IP address to the host. The administrator need not assign the IP addresses, and a limited number of IP addresses serves a larger organization. When a workstation boots, it requests an IP address from the DHCP server along with other information, such as the DNS server IP address, the gateway IP address, and the subnet mask. The DHCP server takes an address from a pool of IP addresses and gives it to the workstation to use temporarily. The administrator can configure how long the address is leased.

The DHCP Process

DHCP is a system-V service that handles client requests on a network and allocates IP addresses.

The service gets activated on the system by installing the DHCP package. The DHCP process can be divided into a number of phases.

1. In the IP request phase, a client broadcasts the IP address request to the DHCP server.
2. In the IP release phase, the DHCP server receives the request and processes it. It responds to the request by sending the IP address, the subnet mask, the duration of lease, and the IP address of the DHCP server to the client.
3. In the client acceptance phase, the client accepts the information and broadcasts it to the network server so that the server ensures that the IP addresses used by the clients are unique.
4. In the server verification phase, the server sends a message to the client stating that it received the acceptance and the client is configured to use TCP/IP.
5. In the lease renewal phase, when half of the lease time has expired, the client sends a request to the server to extend the lease time or sends a request for a new IP address.

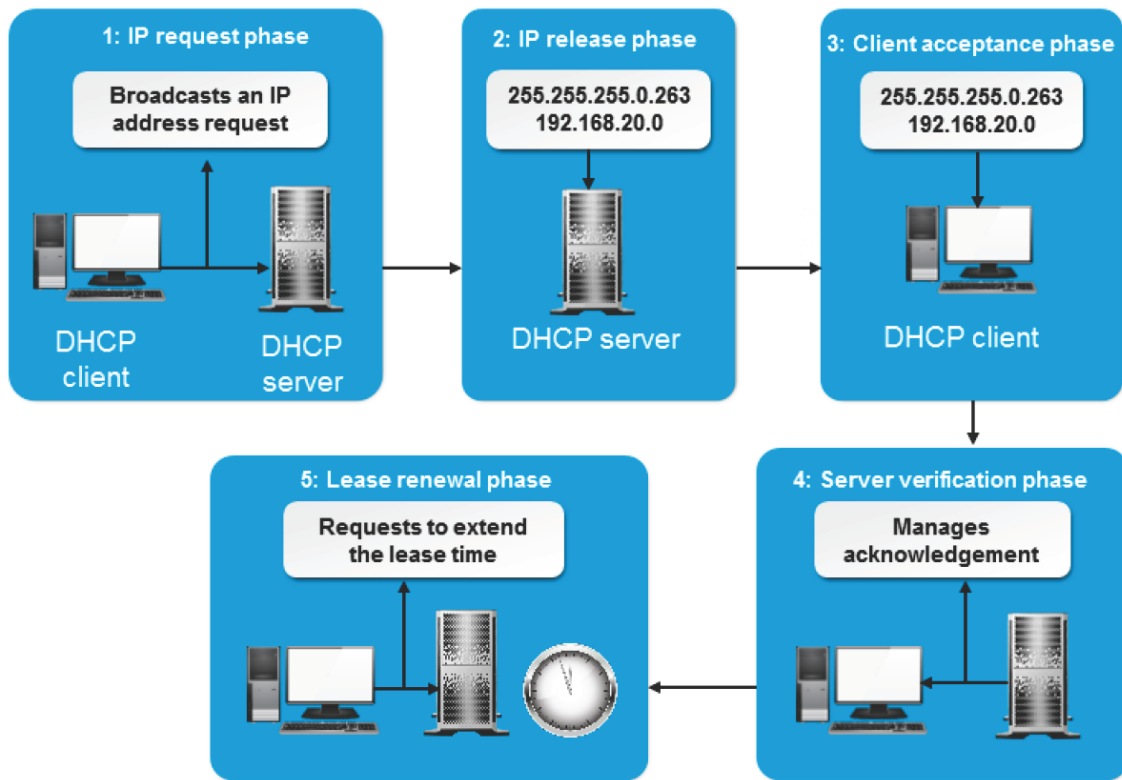


Figure 12-20: Different phases of the DHCP process.

DNS

The **Domain Name System (DNS)** is a distributed, hierarchical database system that maintains information about domain names and their equivalent IP addresses on a network. It uses this information to translate a fully qualified domain name into its numeric IP address or vice versa. IP addresses are used by networked computers to locate, connect, and communicate with each other.

DNS translates IP addresses to their corresponding domain names. It works like a central system ensuring that there are no duplicate domain names and IP addresses on the network.

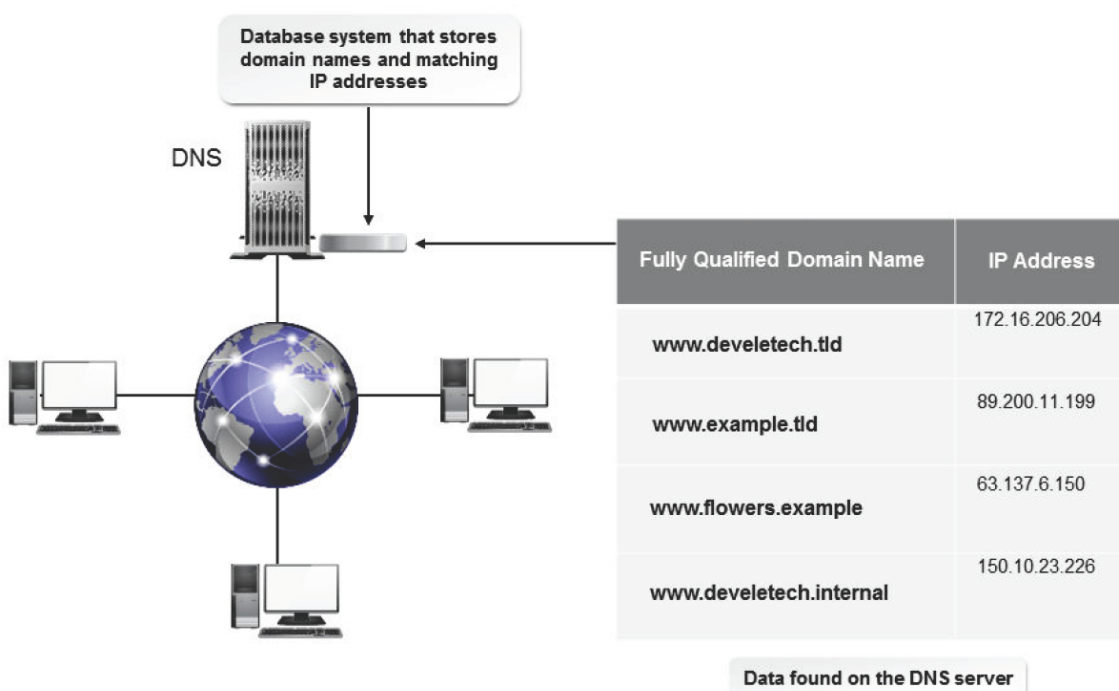


Figure 12-21: Allocation of domains using DNS.

DNS Utilities

Various utilities are used to resolve DNS hostnames. Some of the resolving utilities are dig, host, and nslookup.

Domain Names

A **domain name** is a label given to a **domain**, which is a node in the hierarchical structure of data stored in the DNS. It is the concatenation of all labels from the node to the root node, not including the node name (if any). A domain name is represented by a string, and each node is separated by a period. Each domain name is unique within its parent domain.

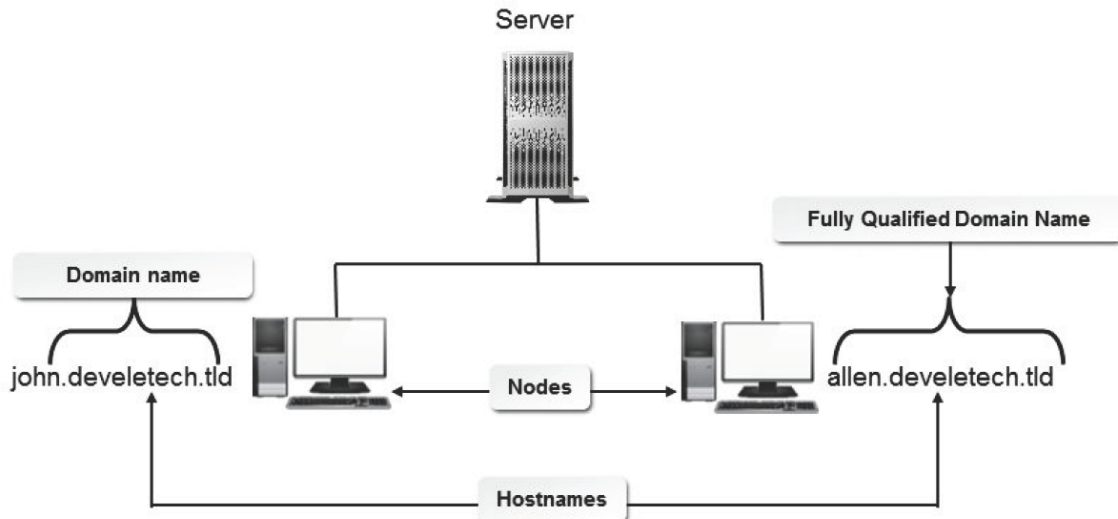


Figure 12-22: Assigning domain names to systems.

Hostname

A hostname is the name of the node itself (i.e., "john" and "allen" in the example figure), and may or may not include the domain name.

Subdomains

A subdomain is a part of a larger domain name. A DNS hierarchy comprises a root-level domain, followed by top-level domains, second-level domains, and subdomains. Each top-level domain contains subdomains, which are referred to as child domains.

The FQDN

A **Fully Qualified Domain Name (FQDN)** is a method by which systems are uniquely identified on the worldwide network. A complete domain name consists of a hostname, one or more subdomains, and the top-level domain.

Zones

A **zone** is a point of delegation in a DNS tree structure that maps to a domain. A zone can map to an entire domain with all of its child domains or to a specific portion of a domain. Each zone will have one authoritative name server or one or more secondary name servers.

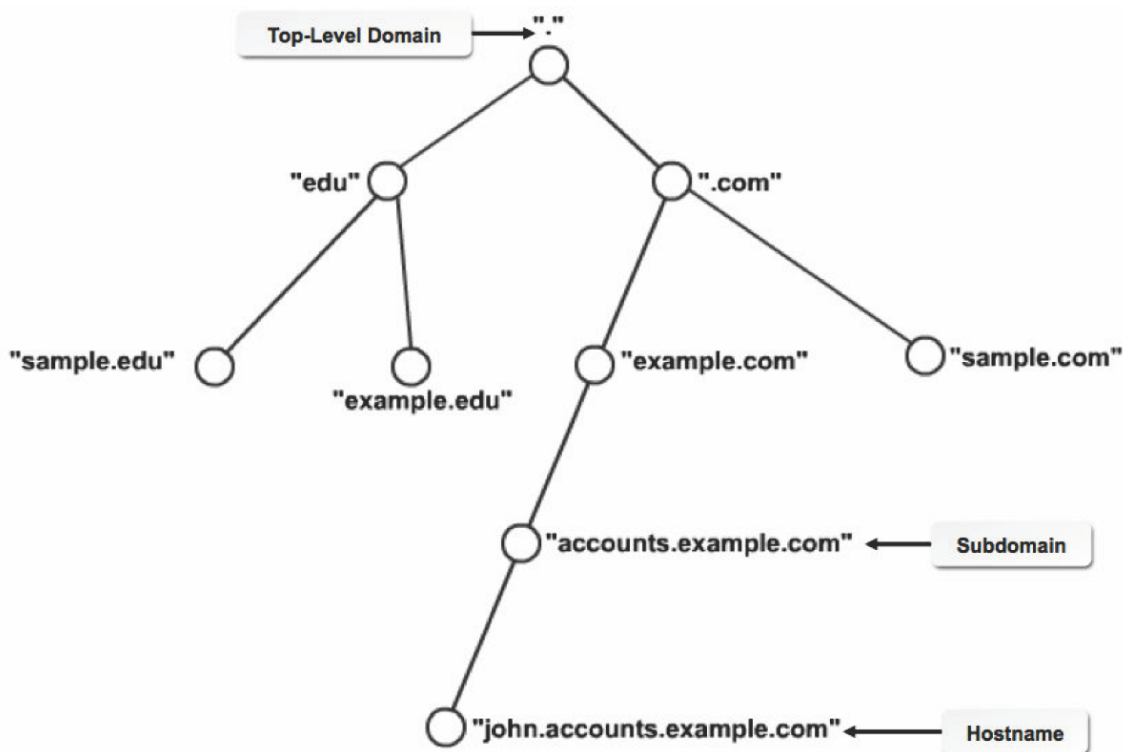


Figure 12-23: Delegation of domains using zones.

Forward and Reverse Zones

There are two types of zones: forward and reverse.

Zone	Description
Forward zone	A zone that is used for mapping hostnames to IP addresses. It contains information on the time allocated for the DNS server to get updated.
Reverse zone	A zone that is used for mapping IP addresses to hostnames. A reverse zone can be used to resolve the IP address of a domain to trace unauthorized users.

The Domain Name Resolution Process

The process of domain name resolution involves several phases.

1. The DNS query containing the domain name is sent by an application to the resolver, requesting an IP address.
2. The resolver searches its cache for matching domain names. If any entry is found, then the respective IP address is forwarded to the client application. In case no entries are found, then the query is forwarded to the name server.
3. The name server, if authoritative for the zone, sends the reply to the resolver. If the name server is nonauthoritative, then the secondary name server forwards the query to the primary or authoritative name server, which sends the reply to the resolver.
4. The resolver then resolves the IP address and sends the reply to the client.

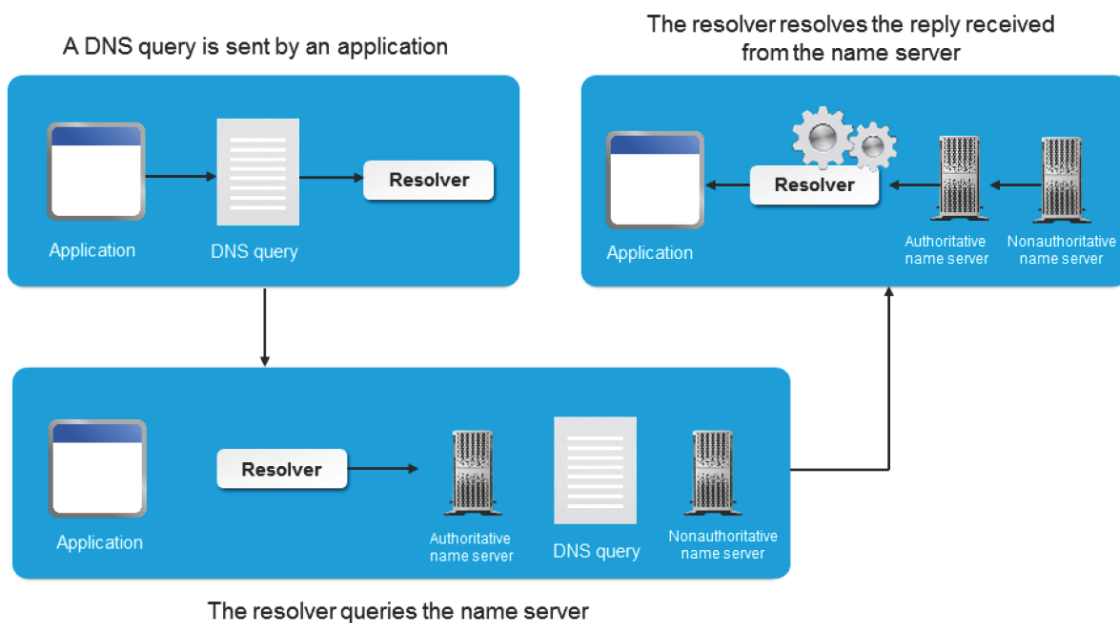


Figure 12-24: The various steps in the domain name resolution process.

The dig Utility

The dig utility interacts with name servers and displays the results to users. It may be configured to query a single name server or multiple name servers in the form of batches. It references the **/etc/resolv.conf** file for the list of name servers to be queried.

Various options can be used to configure the dig utility based on user requirements. Some of the frequently used command options are listed in the following table.

Command Option	Description
-b	Sets the source IP address of the query. The utility will query name servers using this IP address.
-f {file name}	Enables the utility to query name servers based on the processes listed in the specified file.
-p {port#}	Specifies the port to be used to send queries.
-4	Forces the utility to use only the IPv4 protocol for querying.
-6	Forces the utility to use only the IPv6 protocol for querying.
-t {query type}	Sets the query type. Some of the valid query types are soa, axfr, ixfr, and mx.

Various query options can be used to configure the dig utility based on user requirements. Some of the frequently used query options are listed in the following table.

Query Option	Enables You To
+tcp	Set the resolver to use the TCP service to query the name server.
+domain={somename}	Create a search list for the desired domain.
+search	Configure the resolver to use the search list specified in the /etc/resolv.conf file or through the +domain option.
+nssearch	Configure the resolver to search for name servers based on the zones in which they are defined.
+identify	Display the IP address and port number of the name server that answers the query.
+trace	Trace the DNS query for root name server information from the /etc/resolv.conf file.

Syntax

The syntax of the dig utility is `dig [options] {query options} {Fully Qualified Domain Name | IP address}`.

The host Utility

The host utility is a DNS lookup utility, similar to the dig utility. It is used to convert system names to IP addresses and vice versa. The host utility has various options.

<i>Option</i>	<i>Enables You To</i>
{name}	Set the domain name to be looked up by the host utility.
-t {query type}	Set the type of query to be sent to the name server.
-W {seconds}	Set the time the resolver must wait for a reply before quitting.
-s	Force the resolver to terminate the querying process once it fails without retrying.

Syntax

The syntax of the host utility is `host [options] {FQDN | IP address}`.

The nslookup Utility

The nslookup utility is used to query name servers over the Internet and check whether the name-to-IP address mapping is correct in the DNS configuration files. It operates in two modes—interactive and noninteractive. Interactive mode allows a client to query the name server for specific information. Noninteractive mode displays only standard information on the host.

Syntax

The syntax of the nslookup command is `nslookup {host name or FQDN}`.

Resolver Files

Various files are used to configure resolvers for resolving domain names and hostnames.

<i>File</i>	<i>Description</i>
/etc/hosts	Contains the hostname to IP address mapping information for systems on a network. In older versions of Linux, the /etc/networks file was used for this purpose.
/etc/host.conf	Contains information on how the hostname lookups are to be performed. For example, if the /etc/host.conf file contains the line "order hosts,bind," the hostname lookup will be performed first in the local /etc/hosts file and then in the DNS. The default entry in the /etc/hosts file is "order hosts,bind."
/etc/nsswitch.conf	The name server switch configuration file, contains information about each and every database and the order in which they work. The first column contains information about the database and ends with a colon; the remaining columns specify the order in which the database should use the service. For example, in the file entry <code>hosts: files dns</code> , <code>hosts</code> refers to the hosts database. This means that the host entries in the local files will have higher priority than the entries in the DNS server. In case the hostname entries are not found in the local files, the search will continue in the DNS.
/etc/resolv.conf	The resolver configuration file is a set of routines in the C library that provide access to the Internet DNS. The resolver configuration file contains a list of keywords with values that are read by the resolver routines, the first time they are invoked by a process. The three different configuration options are name server, domain, and search.

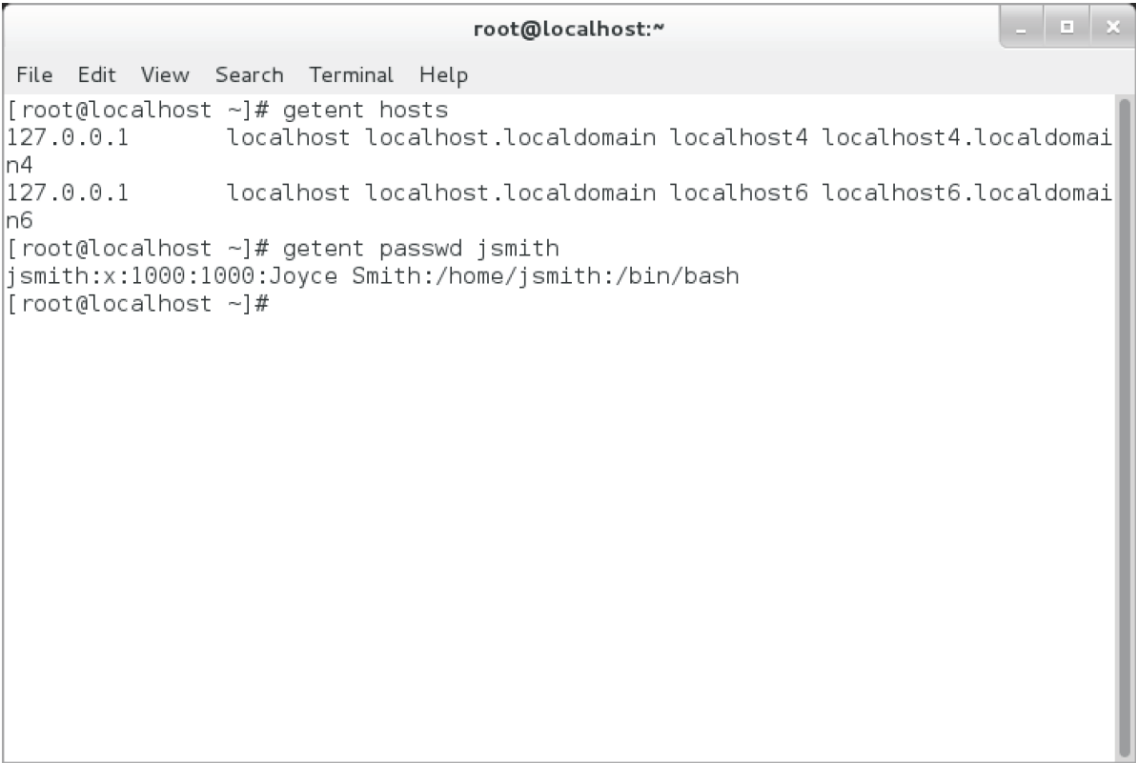
The getent Utility

The **getent** utility is a database lookup utility. It is used to display entries from databases supported by the Name Service Switch (nss) libraries, which are configured via the **/etc/nsswitch.conf** resolver file. The utility can be used to look up user, group, hostname, and service information via the nss-configured databases. The getent utility has just one required option, which is the database to use for the lookup.

<i>Option</i>	<i>Enables You To</i>
{database}	Specify the database to query.
[key ...]	Optionally specify one or more keys to look up. If no key is specified, the entire database may be displayed if that database supports enumeration.

Syntax

The syntax of the `getent` utility is `getent {database} [key ...]`.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
[root@localhost ~]# getent hosts
127.0.0.1      localhost localhost.localdomain localhost4 localhost4.localdomai
n4
127.0.0.1      localhost localhost.localdomain localhost6 localhost6.localdomai
n6
[root@localhost ~]# getent passwd jsmith
jsmith:x:1000:1000:Joyce Smith:/home/jsmith:/bin/bash
[root@localhost ~]#
```

Figure 12-25: Using the `getent` command to display host information and information about the `jsmith` user.

The `named.conf` File

The [`named.conf`](#) file is a user-defined configuration file that is used to manage the Berkeley Internet Name Domain (BIND) service. This file is invoked when the `named` service starts. It contains statements and comments. Statements define zone settings and comments contain messages or descriptions about the statements inside a file. Comments can be either a single-line or multi-line text.



Note: Single-line comments start with `//` and multi-line comments start with `/*` and end with `*/`.

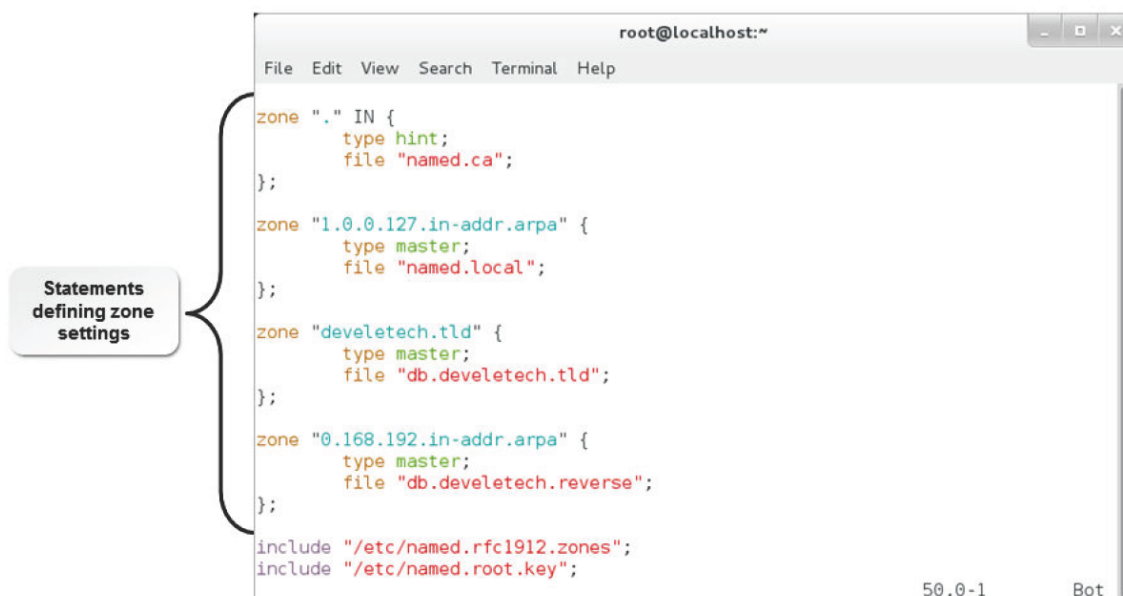


Figure 12-26: Various declarations in the named.conf file.

DNS Resource Records

A **DNS resource record** defines parameters for a zone. It contains five components: the fully qualified domain name, the TTL, the record class, the record type, and the record data. The format of a resource record is defined by the Request for Comments (RFC). The record data in a resource record depends on the record type.

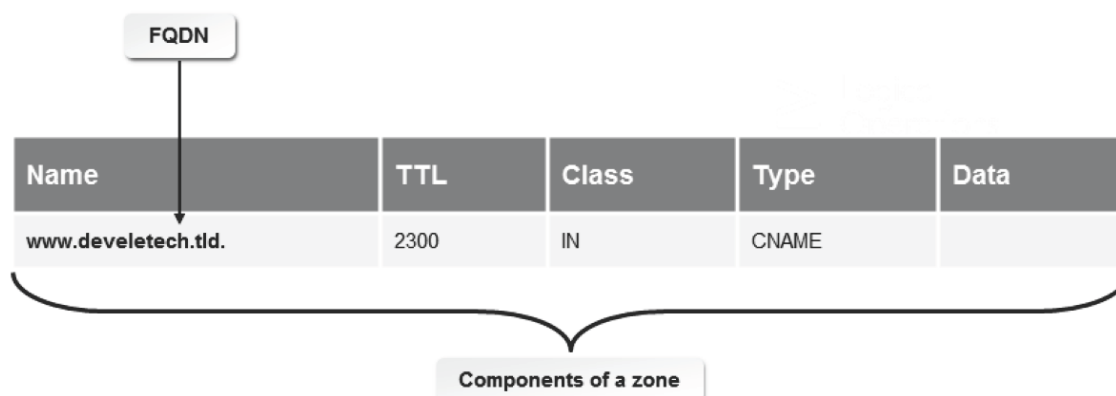


Figure 12-27: Interpretation of a DNS resource record.

Various DNS records and the format used for each are given in the following table.

DNS Record	Description
SOA	The Start Of Authority (SOA) record is used to specify information about a zone in the string of fields format. The SOA record tells the server to be authoritative for the zone. Each zone will contain only one SOA record. The format of an SOA record is: @ IN SOA primary nameserver{hostmaster email}{ {serial number}{time to refresh}{time to retry}{time to expire}{minimum TTL} }.
NS	A Name Server (NS) record is used to define the authoritative name server for a specific zone. The format of the NS record is IN NS {nameserver}. The name server should be an FQDN. It can either be a primary server or a slave server.
A	An Address (A) record is used to assign an IP address to a name. The format of the A record is {hostname} IN A {IP address}. The IP address should not be terminated with a period (.). If the hostname is omitted, the A record will point to the default IP address at the top of the namespace.
PTR	The Pointer (PTR) record is used for reverse name resolution mapping. It is used in the reverse map zone files to map an IP address to a name. The format of the PTR record is {last IP digit} IN PTR {FQDN of system}. The {last IP digit} specifies the last number in an IP address, which should point to a particular system's domain name. The PTR record should always end with a period. For example, 253 IN PTR srv##.example.com.

DNS Record	Description
MX	A Mail eXchange (MX) record is used to specify the relative preference of mail servers for a zone. The MX record format is IN MX {priority value} {mail server name}. The highest priority value is 0, which is assigned to a host where the mail is destined. Two hosts can have the same priority to distribute mail equally between them. Any number of MX records can be defined for a domain.
CNAME	A Canonical Name (CNAME) record is used to map an alias name to the real name. The CNAME record is also referred to as an alias record. The format of the CNAME record is {alias name} IN CNAME {real name}. CNAME records are generally used to point to another domain.
TXT	A Text (TXT) record is used to map text with a hostname. It is used to validate genuine email sources from a domain.

How to Configure DNS Services

Follow these general procedures to configure DNS services.

Configure the DNS Service

To configure the DNS service:

1. Log in as **root**.
2. To use the **named.cachingnameserver.conf** file as a template, enter `cp /etc/named.caching-nameserver.conf /etc/named.conf`.
3. Configure the DNS.
 - a. Specify the global configuration options.
 - b. Add zone statements for the root, loopback domain, and forward and reverse zones.
 - c. Create forward and reverse zone files.
4. To check the named syntax, enter `named-checkconf -f /etc/named.conf`.
5. Start the **named** service.
 - a. To start the **named** daemon, enter `service named start`.
 - b. If the named service is already running, to reload the named daemon, enter `service named reload`.
 - c. To reload the zone files, enter `rndc reload`.
 - d. To start the named service at system startup, enter `chkconfig named on`.

TOPIC D Manage Remote Network Systems

Previously, you configured network services to allow a system to access network resources. In situations such as adding new systems to a network, you will directly communicate with the system and modify data on the system to connect your computer to the other system. In this topic, you will explore Secure Shell (SSH) and Virtual Network Computing (VNC) and examine their functions to communicate with remote systems.

As a system administrator, you will address the needs of users who are scattered across different locations. There may be some meetings or conferences that require you to connect to the server remotely. Your capability as a system administrator will increase if you know how to connect to remote systems. You can access data remotely and troubleshoot all systems from one location.

PKC

Public Key Cryptography (PKC) is an encryption method that uses a public and private key pair.

Data is encrypted using the public key and then transmitted through the network. When the data reaches its destination, it is decrypted using the private key.

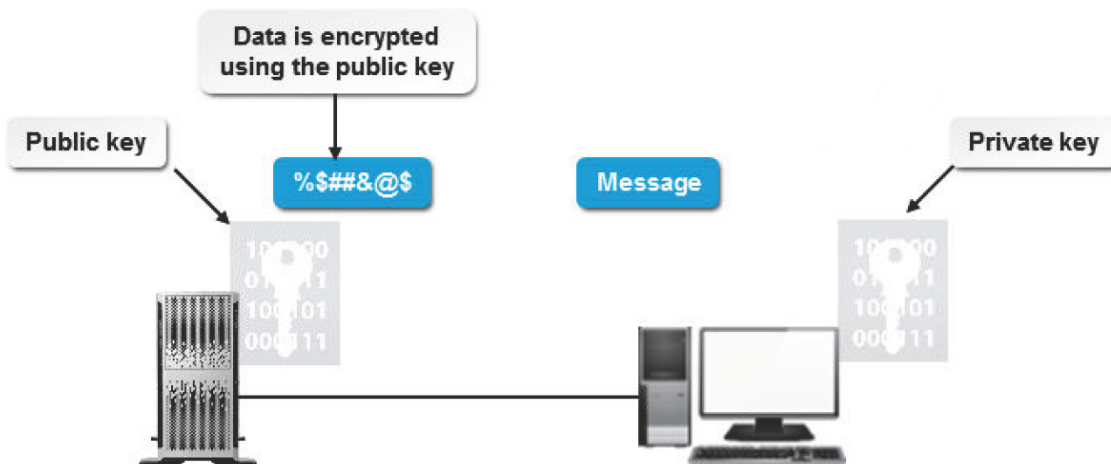


Figure 12-28: Encrypted data that uses both the keys.

SSH

Secure Shell (SSH) is a network protocol that securely controls the flow of data among computers on a network. SSH architecture contains the transport layer, the user authentication layer, and the connection layer. The client places a request that is authenticated by the user authentication layer.

This layer transfers the request, which is authenticated by the transport layer, to the server through the connection layer. By making use of public-key cryptography to encrypt data, this architecture makes SSH flexible and secure. Many versions of SSH, such as SSH1 and SSH2, are available.

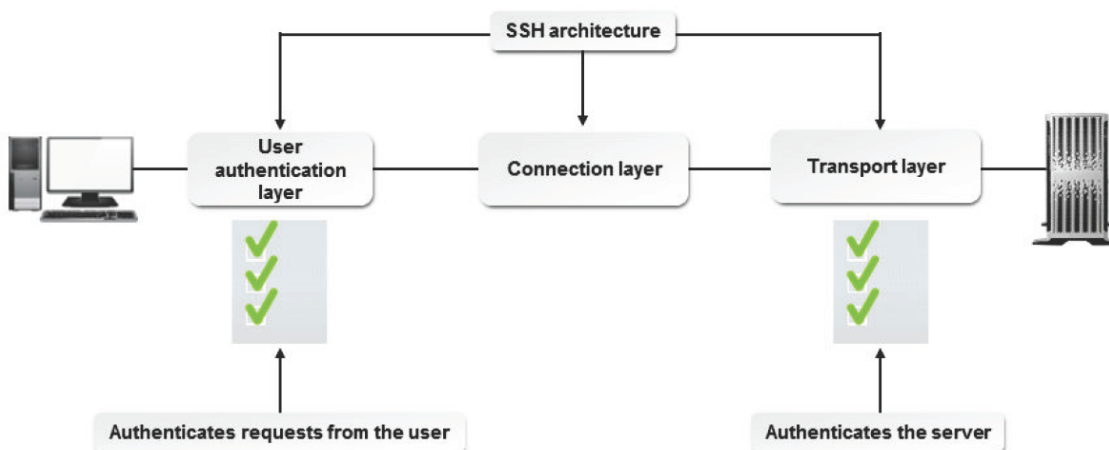


Figure 12-29: SSH controlling secure communication on a network.

OpenSSH

OpenSSH is an open source implementation of the SSH protocol that is included with most Linux distributions. Data to be transmitted passes through a secure tunnel that is formed between two systems. Telnet transmits data, which includes passwords, that can be easily intercepted by any system on the network. OpenSSH provides a strong client-server authentication method for data transmitted by Telnet and similar applications.

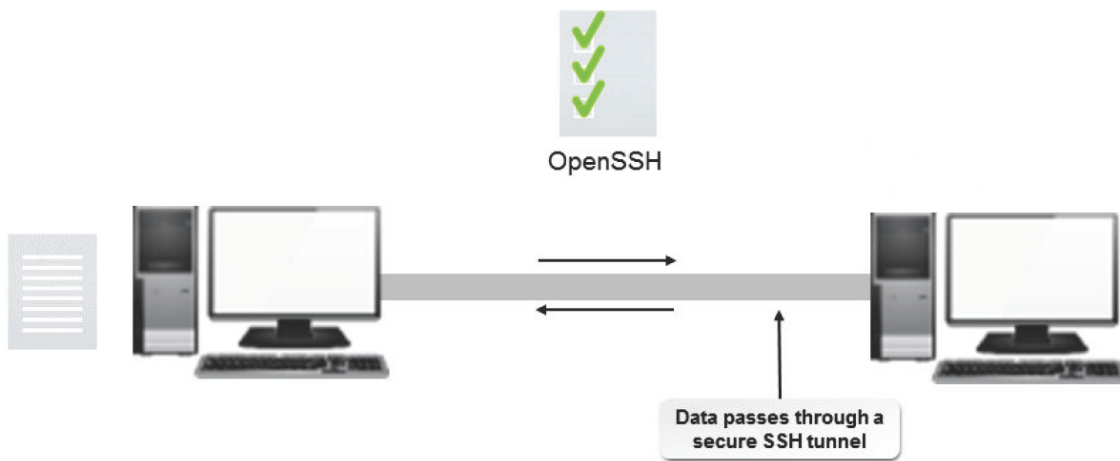


Figure 12-30: Data transfer using the OpenSSH tunnel.

The ssh-keygen Command

The [ssh-keygen](#) command generates, manages, and converts authentication keys. The following table lists some options of the ssh-keygen command.

Use This ssh-keygen Command Option	If You Need To
-b {bits}	Specify the number of bits to be created in the key.
-c	Change the comment in the public and private key files.
-f {file name}	Specify the file name of the key file.
-l	Show the fingerprint of the specified public key file.
-p	Change the passphrase of a private key file instead of creating a new private key.

Public and Private Keys

Both private and public keys are involved in an authentication process. Each key is a collection of alphanumeric and special characters that uniquely identify each system. A private key is involved in public key authentication that is retained on the local system. A public key is involved in public key authentication that is made known to remote systems. The private key is retained on the local system. The public key is made known to remote systems. The private key is never transmitted to the destination server. A meaningful message will result only when the destination's private key is combined with the public key of the original server. While logging in to a remote system, the private and public keys are combined by the remote server for verification. The keys need to match if a user has to log in to the system and transfer files. Authenticity is established by the remote server, which then grants the necessary permissions.

Key Files

The key pairs that you create using SSH are stored in different files depending on the algorithm you use.

File Created	Algorithm
<i>id_dsa</i> and <i>id_dsa.pub</i>	Digital Signature Algorithm (DSA)
<i>id_rsa</i> and <i>id_rsa.pub</i>	RSA with SSH protocol version 2 (RSA stands for Rivest, Shamir and Adleman)
<i>identity</i> and <i>identity.pub</i>	RSA with SSH protocol version 1

OpenSSH Key File Names

When you install OpenSSH, typically two pairs (RSA and DSA authentications) of public and private keys are created in the **/etc/ssh** directory. The two pairs created are: **/etc/ssh/ ssh_host_rsa_key** and **/etc/ssh/ssh_host_rsa_key.pub** and **/etc/ssh/ssh_host_dsa_key** and **/etc/ssh/ssh_host_dsa_key.pub**. In newer, modern systems the Elliptic Curve Digital Signature Algorithm (ECDSA) authentication protocol is sometimes used in addition, or instead of, DSA. If ECDSA is configured, typically a pair of public and private keys are created named **/etc/ssh/ ssh_host_ecdsa_key** and **/etc/ssh/ssh_host_ecdsa_key.pub**.

The `/etc/ssh/sshd_config` File

The **`/etc/ssh/sshd_config`** file is the SSH server configuration file. Most lines in this configuration file are commented, indicating the default settings that have been applied. You can remove the comment and change the default settings. A few SSH server configuration file options are listed in the following table.

<i>Use This sshd Option</i>	<i>If You Need To</i>
X11Forwarding yes	Run or stop running a program on one system and display the X window output on another system.
X11Forwarding no	Stop running a program on one system and display the X window output on another system.
PermitRootLogin	Allow the root user to login via SSH.
MaxStartups {number}	Specify the maximum number of connections that can be made to a host.
LoginGraceTime {time in seconds}	Drop connections if the connection is not established within the login grace time.
AuthorizedKeysFile {file name}	Specify the location of the file that contains the authentication keys.
PermitEmptyPasswords yes/no	Specify whether a null password is allowed or denied.

The ssh-agent Program

The **ssh-agent** program is a program that holds private keys for public key authentication. This program starts with an X-session or login session, and acts as a parent program, while all other programs or windows are its clients. The ssh-agent does not send a private key through channels; instead, the task that requires a private key is performed by the agent and the result is returned to the client. Initially, the agent does not have a private key. Keys are added using the **ssh-add** utility.



Figure 12-31: Role of ssh-agent.

Server Keys

When you install an SSH server, public host and server keys are automatically created for authentication purposes. The server keys are not stored anywhere in the disk; they are automatically regenerated every hour to ensure security.

The `known_hosts` File

When you connect to a remote host, the host sends you public host and server keys for authentication. Your system looks up the **~/.ssh/known_hosts** file to locate an entry for the host's keys, and if an entry is found, you will be allowed to access the host. Otherwise, a message is displayed, stating that the authenticity of the remote host is not yet established. You need to type "yes" to trust the remote host and connect to it. Whenever a SSH connection is made to a remote terminal, the public key of that system or host is added to the **~/.ssh/known_hosts** file.



Note: The `~/.ssh/known_hosts` file was known as the `/etc/ssh_known_hosts` file in older versions of Linux.

The `~/.ssh/authorized_keys` File

The public key for a user may be added to the `~/.ssh/authorized_keys` file on the remote server to enable password-less authentication. In recent versions of OpenSSH, transferring these keys to remote systems is best accomplished via the `ssh-copy-id` command.

SSH Protocol Versions

By default, both SSH protocol versions 1 and 2 are compatible with OpenSSH. Although using SSH 2 offers enhanced security benefits, you can use SSH 1 based on the client. To change the configuration, you can modify the Protocol option in the `sshd_config` file.

You can also specify OpenSSH to use SSH 2 by default and fall back on SSH 1, whenever needed, by modifying the Protocol option as Protocol 2,1.

Multiple SSH Connections

If your network or firewall settings change while you are connected to the SSH server, you may lose the connection. To prevent such loss of connection, SSH allows multiple simultaneous connections from multiple hosts.

You can use the MaxStartups option to specify the maximum number of connections that can be made to a host. However, additional connections will be dropped only until the login grace time expires. The login grace time is specified using the LoginGraceTime option.

The SCP Command

SCP stands for secure copy. The `scp` command enables you to transfer secure copies of files over an encrypted remote network connection. The command uses SSH to provide security for data transfer and authentication, using passwords and passphrases.

Using the `scp` command, you can copy files from a remote system to your local host and vice versa.

Furthermore, you can also transfer files between two remote systems, without involving your local system.

`.rhosts` and `.shosts`

Using `.rhosts` or `.shosts`, SSH allows users to log into another remote host with just a user name.

A password or passphrase is not required for authentication. If the remote machine consists of the `.rhosts` or the `.shosts` files in the home directory, users are allowed to log in and access the machine. However, this method is a threat to security because if users are allowed to access a remote machine without authentication, users can further access other machines that trust the first remote machine.

The `sftp` Command

The `sftp` command, or SSH File Transfer Protocol (SFTP), is used to transmit data files over a secure, encrypted SSH connection. Thus, it provides the functionality of FTP with the security and privacy of SSH. The SFTP command is compatible with most normal FTP commands in interactive mode.

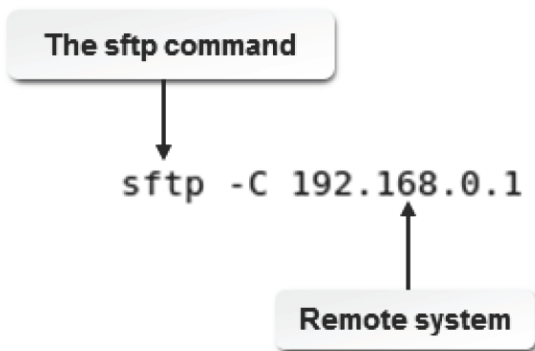


Figure 12-32: Connecting to a remote system using `sftp`.

Syntax

The syntax of the `sftp` command is `sftp {hostname}` or `sftp {user}@{hostname}`. The

command can also be used with the `-C` flag to allow file compression.

Tunneling

Tunneling is a layered protocol model in which one protocol is layered over another. The inner protocol, called the payload protocol, is encapsulated within another protocol, which is called the delivery protocol. This provides security and flexibility to the connection. Some of the tunneling protocols are Generic Routing Encapsulation (GRE), GPRS Tunneling Protocol (GTP), and Multiprotocol Label Switching (MPLS).

An SSH tunnel is created when an SSH protocol connection is made. SSH tunneling enables users to access websites and bypass firewalls by setting up **proxy servers**. A protocol that is blocked by a firewall is encapsulated within a different protocol that is not blocked by the firewall, thus establishing the connection.

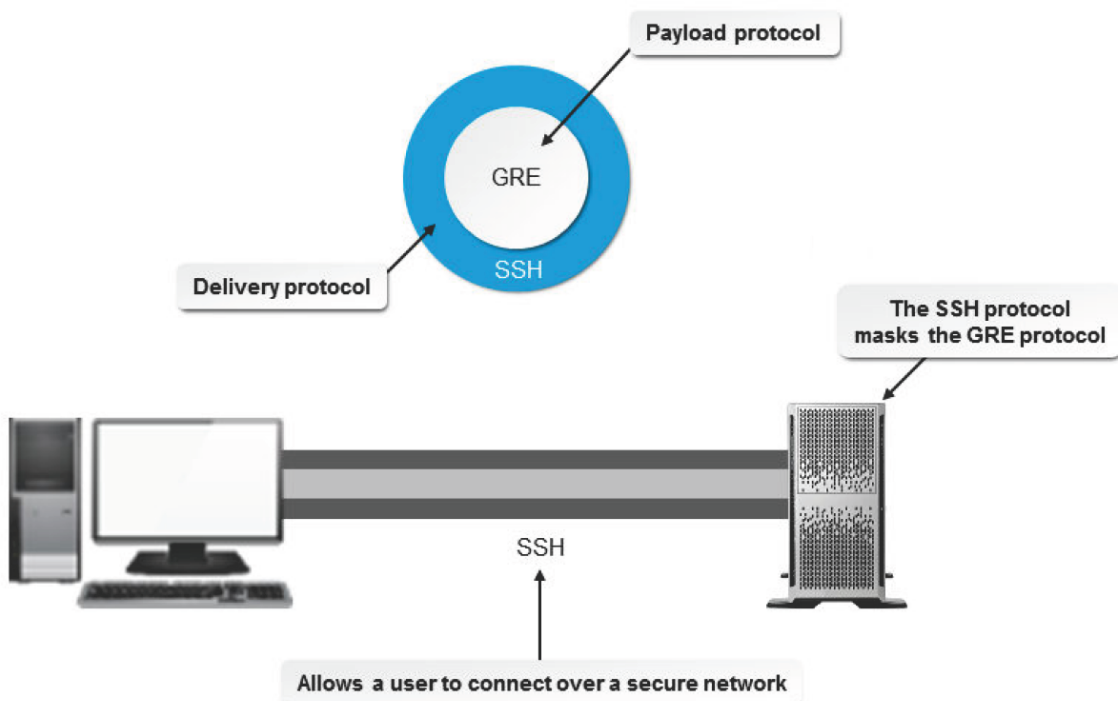


Figure 12-33: The tunneling protocol and its architecture.

X Forwarding

X forwarding is a mechanism by which programs are run on one machine and the X window output is displayed on another machine. X forwarding can be enabled or disabled by setting the `X11Forwarding` option to yes or no in the `/etc/ssh/sshd_config` file. This allows X11 tunnelling over an SSH connection.

```
root@localhost:/etc/X11
File Edit View Terminal Tabs Help
# Accept locale-related environment variables
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL
#AllowTcpForwarding yes
#GatewayPorts no
#X11Forwarding no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#UseLogin no
#UsePrivilegeSeparation yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#ShowPatchLevel no
#UseDNS yes
#PidFile /var/run/sshd.pid
#MaxStartups 10
```

Figure 12-34: The X11 forwarding option enabled in the configuration file.

Port Forwarding

SSH secures the TCP/IP protocol using port forwarding. SSH can map a local port of the client to a remote port on the server. The following command is used to create a TCP/IP port forwarding channel:

```
ssh -L local-port:{remote-hostname}:{remote-port} {username}@{hostname}
```

Port forwarding is also used to transfer information securely through network firewalls.

VNC

Virtual Network Computing (VNC) is a platform-independent system through which a user can control a remote system. The virtual network is made up of the VNC client, the VNC server, and the VNC protocol. The client views the output that is displayed by the server through the VNC protocol. The user can run multiple VNC sessions at any given time. However, the display for each VNC client may differ from the display of the VNC server.

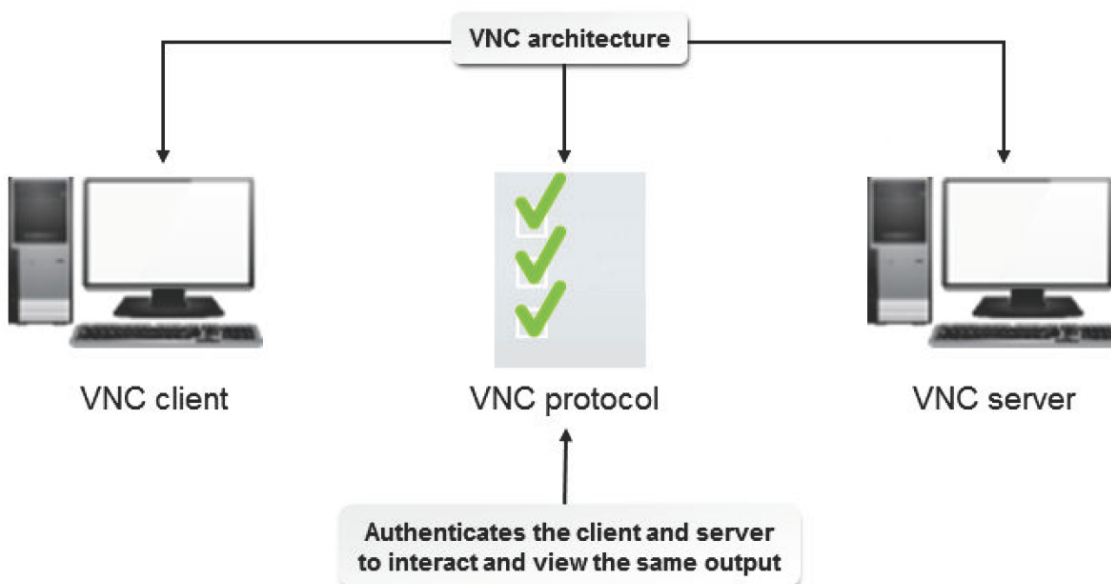


Figure 12-35: The VNC protocol enables the client to view the output displayed by the VNC server.

The vncserver Command

The vncserver command is used to start a system with VNC. The `$HOME/.vnc/xstartup` file allows a user to control applications running on a remote system. You can specify the display number that the VNC server will use when it is started.

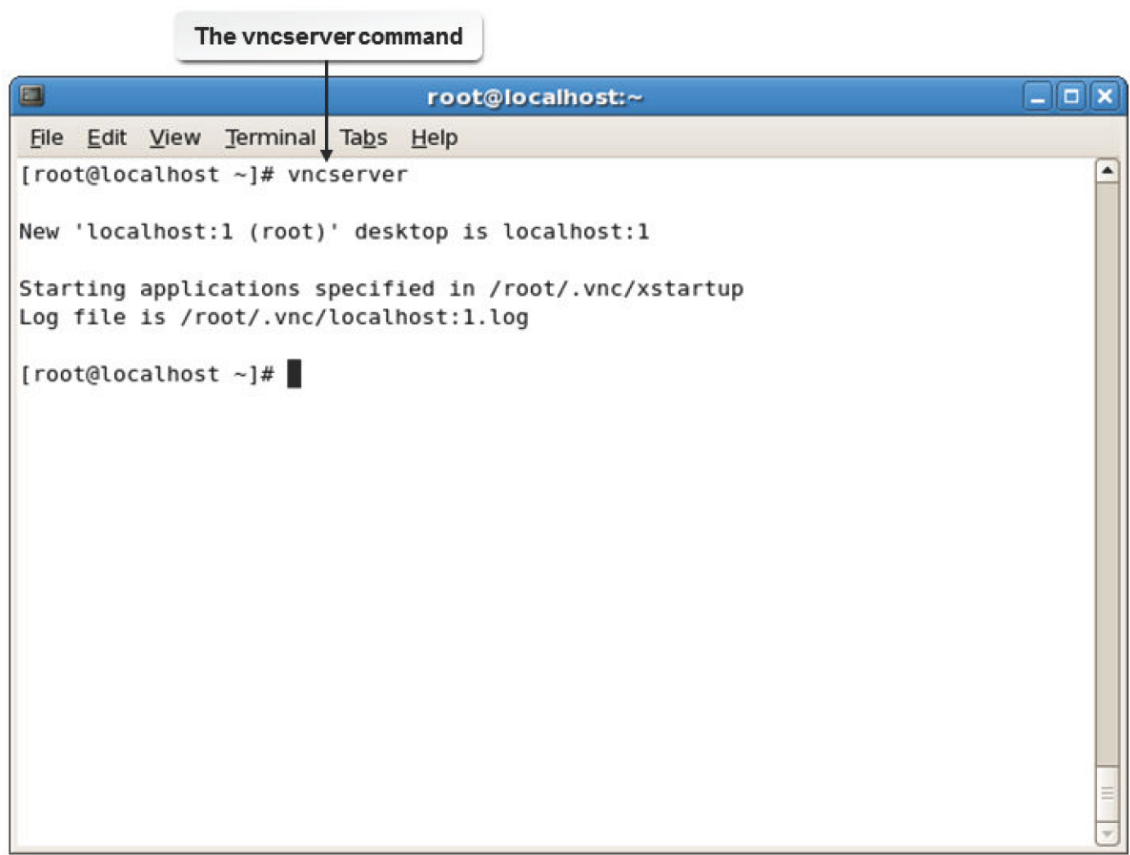


Figure 12-36: VNC enabled on a Linux system.

The vncserver command has various options.

Option	Enables You To
-name {desktop name}	Specify the desktop name.
-geometry {resolution}	Specify the screen resolution of the remote desktop.
-depth {depth}	Specify the pixel depth of the desktop. The accepted values are 8, 15, and 24.
-pixelformat {format}	Specify the pixel format such as RGB and BGR.

Syntax

The syntax of the vncserver command is `vncserver {:display number} {options}`.

The vncviewer Command

The vncviewer command is used to view the VNC client. Various options are available for specifying vncviewer parameters.

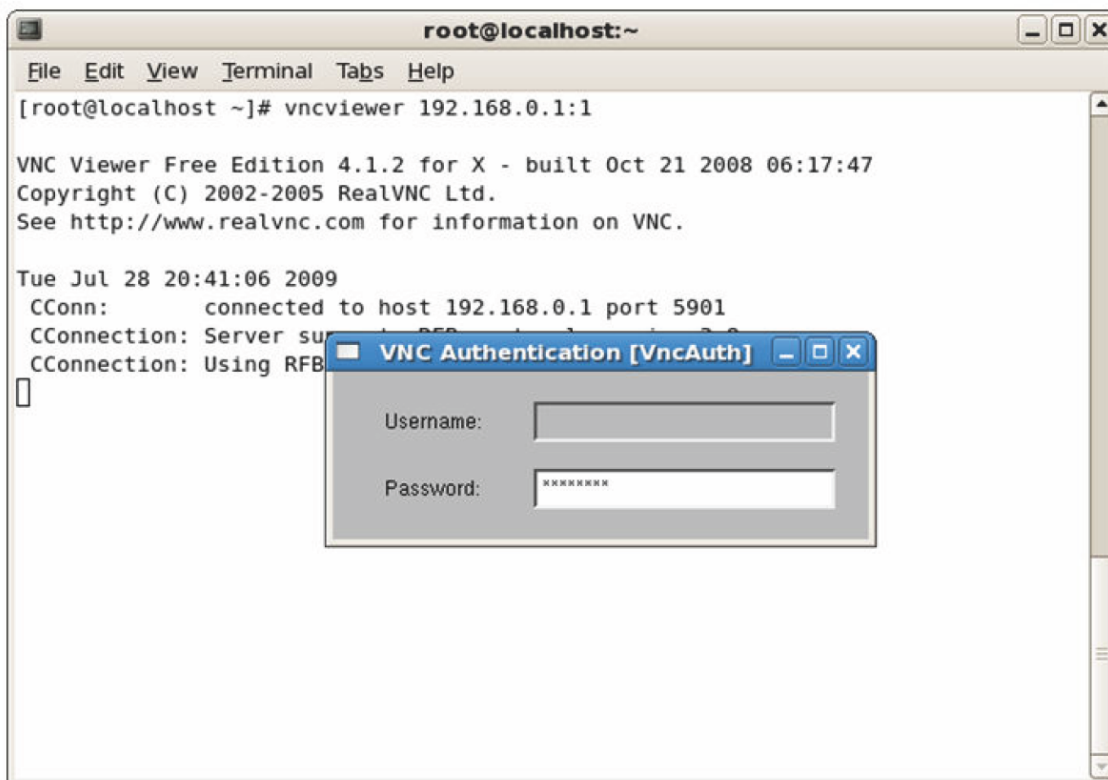


Figure 12-37: The `vncviewer` command is used to connect to a VNC server.

The `vncviewer` command has various options.

Option	Enables You To
<code>-display {Xdisplay}</code>	Specify the X display.
<code>-listen [port]</code>	Search for reverse connections from the VNC server.
<code>-Shared</code>	Keep multiple VNC connections open.
<code>-FullScreen</code>	Start the VNC client in full-screen mode.
<code>-via {gateway}</code>	Create a tunnel to a gateway system and then connect the client to the host.

The rdesktop Utility

`rdesktop` is an open source utility, released under General Public License (GPL). It enables a client system running Linux to log in to a system running Microsoft® Windows® on a network. It supports Microsoft's Remote Desktop Protocol (RDP). The `rdesktop` command can be used to log in to a remote Windows system.

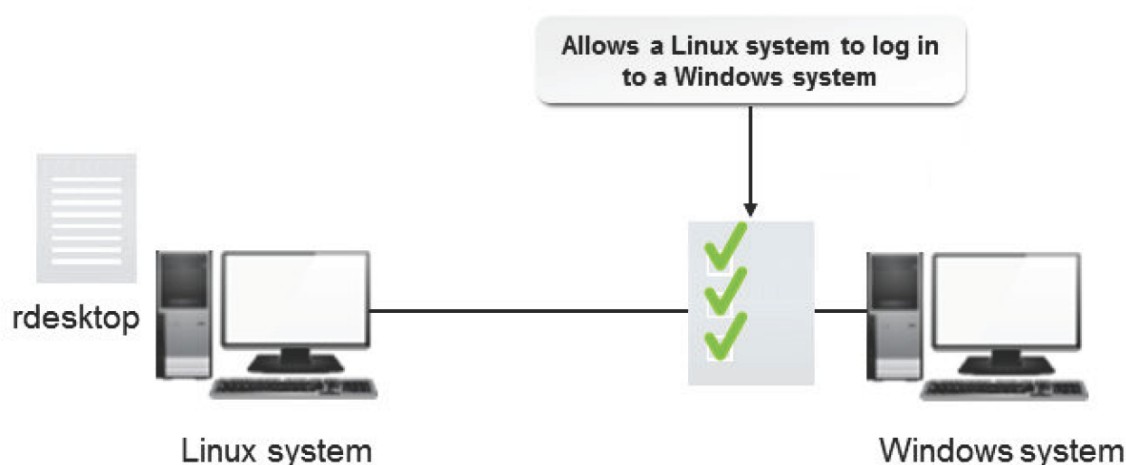


Figure 12-38: A Linux system logging in to a Windows system with the help of `rdesktop` and the RDP protocol.

Syntax

The syntax of the rdesktop command is rdesktop [options] server[:port].

The rdesktop Command Options

Some frequently used rdesktop command options are listed in the following table.

Option	Used To
-u {user name}	Specify the user name for authentication on the server.
-d {domain name}	Specify the domain name for authentication.
-s {application name}	Start a specific application instead of Explorer.
-c {directory}	Specify the initial working directory for the user.
-p {password}	Specify a password for authentication.

RDP

RDP is a multi-channel protocol that allows users running various other operating systems to connect to a system running Microsoft Windows and vice versa, on a network.

SNMP

The [Simple Network Management Protocol \(SNMP\)](#) enables you to remotely monitor and configure network components such as bridges, routers, network cards, and switches.

SNMP management requires two primary elements: a network manager and an SNMP agent.

Element	Description
Network manager	The software running on a workstation through which the network administrator monitors and controls the different hardware and software systems that comprise a network.
SNMP agent	A piece of software running on network equipment that implements the SNMP. The SNMP defines exactly how a network manager communicates with an SNMP agent.

RMON

[Remote Monitoring \(RMON\)](#) is a network management protocol that allows network information to be gathered at a single workstation. For RMON to work, network devices, such as hubs and switches, must support it.

How to Manage Remote Network Systems

Follow these general procedures to manage remote network systems.

Communicate Using Secure Shell

To communicate using secure shell:

1. Log in as a user.
2. Connect securely to another computer.
 - a. To connect to the remote host, enter ssh {user name}@{hostname} | {IP of the destination}.
 - b. If prompted, add the host as a trusted host.

- c. To log in to the system, enter the password.
3. Execute commands securely on another computer.
 - a. To connect to the remote host, enter `ssh {user name}@{hostname} | {IP of the destination}`.
 - b. If prompted, add the host as a trusted host.
 - c. To log in to the system, enter the password.
 - d. To execute the required action, enter the command.
4. Create a tunnel using SSH.
 - a. To create a tunnel using SSH, enter `ssh -L {port number}:{remote server IP} | {FQDN}:{port number} {user name}@{remote server IP} | {FQDN}`.
 - b. If prompted, add the host as a trusted host.
 - c. To log in to the system, enter the password.
5. Authenticate the tunnel with SSH keys.
 - a. To generate a key, enter `ssh-keygen`.
 - b. To generate the keys `id_rsa` and `id_rsa.pub` in `/root/.ssh`, press **Enter** three times.
 - c. Log in as **root** in the second system with which you want to establish an SSH connection.
 - d. Enter `ssh-keygen -d`.
 - e. To generate the keys `id_rsa` and `id_rsa.pub` in `/root/.ssh`, press **Enter** three times.
 - f. To copy the public key from the second system to the first system, enter `ssh-copy-id {user name}@{remote server IP} | {FQDN of the first system}`.
 - g. If prompted, add the host as a trusted host.
 - h. To log in to the system, enter the password.
 - i. To copy the public key from the first system to the second system, enter `ssh-copy-id {user name}@{remote server IP} | {FQDN of the first system}`.
 - j. If prompted, add the host as a trusted host.
 - k. To log in to the system, enter the password.

Configure ssh-agent

To configure ssh-agent:

1. Open the `/etc/skel/.bash_profile` file.
2. Add the `eval `ssh-agent`` statement to provide the same ssh-agent whenever any user logs in.
3. Save and close the file.
4. Open the `/etc/skel/.bash_logout` file.
5. Add the `ssh-agent -k` statement to kill the ssh-agent when the user logs out.
6. Save and close the file.

Transfer Files Securely to Another Computer

To transfer files securely to another computer:

1. Log in as a user in the CLI.
2. To transfer files using the scp utility, enter `scp {options} {source file or folder name} {user name}@{hostname} | {IP of the destination}:{destination file} or {folder name}`.
3. If prompted, add the host as a trusted host.
4. To transfer the file, enter the password.

Run the VNC Server

To run the VNC server:

1. Log in as **root** in the GUI.
2. On the terminal, enter `vncserver` to start the VNC server.
3. Enter the VNC server password, which will be used by clients when connecting to this server.
4. Confirm the password.
5. Write down the **{server name}:{screen number}** that is displayed.

Connect to the VNC Server Using the VNC Viewer

To connect to the VNC server using the VNC viewer:

1. Log in as **root** in the GUI of the client system.
2. To view the VNC server, on the terminal, enter `vncviewer {server name}:{screen number}`.
3. To connect to the VNC server, in the **VNC Authentication** window, in the **Password** text box, enter the password of the server.

ACTIVITY 12-1

Configuring Network Services Review

Scenario

Answer the following review questions.

1. What networking schemes are in use in your organization, and based on what you've learned in this lesson would you change anything?
2. When you need to securely access remote systems, which tools will you use and why?

Summary

In this lesson, you configured and managed various network services and remote network systems.

You will now be able to disseminate information, administer systems remotely, enable communication through mail or chat systems, facilitate technology sharing, manage software licenses, and control unauthorized access.

13 Configuring Basic Internet Services

Lesson Time: 2 hours

Lesson Introduction

In the last lesson, you configured network services. Now, you want to implement a business-oriented service that allows you to share resources and communicate across various platforms on a network. In this lesson, you will configure basic Internet services.

The Internet offers various services such as email, file sharing, downloading, and web browsing. Employees of an organization will often need to access the Internet and communicate with clients across various platforms. As a system administrator, you want to implement a simplified service that provides interoperability among applications and involves cost-effective communication. Internet services facilitate the ability to share and transfer resources across various networks securely.

Lesson Objectives

In this lesson, you will configure basic Internet services. You will:

- Configure email services.
- Control Internet services.

TOPIC A Configure Email Services

In the previous lesson, you configured basic network services. Now, you need to share information through the Internet because this is the most simplified and standardized way of communication and provides client and server email services by defining the email process. In this topic, you will configure email services.

It wasn't long ago that organizations conducted business communications using only the internal mail room and the postal service. Today, however, we communicate electronically with email and instant messaging. Understanding how mail clients operate is critical to implement an email service.

Mail Protocols

A [mail protocol](#) is a set of rules that enable distribution of email messages from a mail server.

Using a mail protocol, an email message may be stored on a server or transmitted to a client's computer when read. Mail protocols enable users to create and manage folders on a server, search for messages, or delete messages. Post Office Protocol (POP), Internet Message Access Protocol (IMAP), and Simple Mail Transfer Protocol (SMTP) are the most frequently used mail protocols.

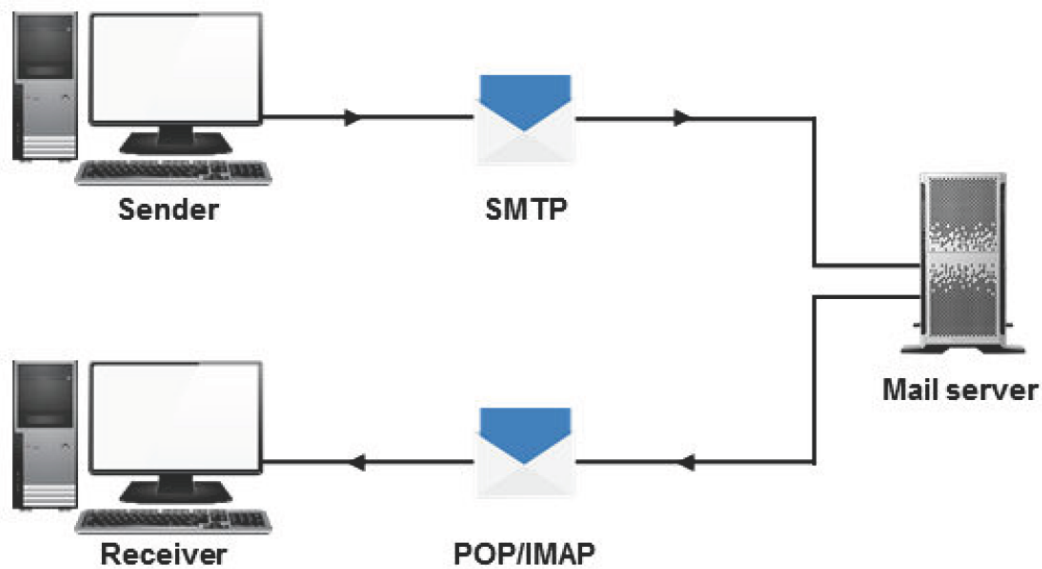


Figure 13-1: Email messages are distributed using the POP, IMAP, and SMTP protocols.

SMTP

The **Simple Mail Transfer Protocol (SMTP)** is a protocol that defines a set of rules to enable interaction between a program sending an email message to a server and a program receiving an email message from a client. The Mail Transfer Agent (MTA) or the Mail User Agent (MUA) acts as an SMTP client. The SMTP server listens to port 25 for client responses. SMTP uses Transmission Control Protocol (TCP) for transmitting messages and Internet Protocol (IP) for routing purposes.

It contains a number of status codes that are used to set specific conditions for communication between the server and the client. It also contains a set of commands that are used for communication between the server and the client.

Figure 13-2: Communication between clients using SMTP.

In addition to text messages, graphics and attachments can also be transferred through email using Extended SMTP (ESMTP). The Extended HELLO (EHLO) command is used by ESMTP clients to communicate with the server, and the server responds with one of the three status codes: success, failure, or error.

Mail Spooling

Spooling is a method of handling delays in delivering email messages. If there is an email delivery delay, SMTP at the originating server spools the message. If there is no delay, SMTP at the originating server sends the email

message to the destination server using the TCP connection.

SMTP on the destination server receives the email message and puts it in the user's mailbox. The same process occurs in reverse when a user at the destination server sends a reply to a user at the originating server.

POP3

Post Office Protocol version 3 (POP3) is used to retrieve email messages over the TCP connection on port 110. The POP client connects to the server and retrieves all messages. It then stores them on the client PC as new messages, deletes them from the server, and disconnects from the server. It supports Multi-Purpose Internet Mail Extensions (MIME) formatted email messages.

In general, POP3 supports the transmission of messages and passwords in clear text format. It also provides authentication by encrypting the messages using the Secure Socket Layer (SSL) protocol over TCP on port 995.

IMAP

The **Internet Message Access Protocol (IMAP)** is used to retrieve email messages over the TCP connection on port 143. The client retrieves all messages from the server, which retains them until they are deleted by the user. Although the protocol transmits messages and passwords in clear text format, it supports SSL encryption of messages over TCP on port 993. In addition, it supports MIME formatted email messages.

Mail Queues

A **mail queue** is a waiting area for email messages that need to be processed by a computer. Mail queues are organized in such a way that the first item added to a queue is also the first item that is sent out of the queue.

Figure 13-3: Email messages are arranged in a queue.

The MTA

The **Mail Transfer Agent (MTA)** is a program on the Internet for sending email messages using SMTP. They use SMTP to indicate the success or failure in the delivery of messages and to form separate queues for failed messages. MTAs often use the Local Mail Transport Protocol (LMTP), a derivative of SMTP, when mail queues are not allowed to be stored on the server end. MTAs allow clients to handle mail queues instead of the server and do not provide mailbox handling features.

Figure 13-4: Routing of email messages through the MTA.

Common Linux MTAs

MTAs support features such as virtual hosting, automatic resending of messages in case of failed delivery, and SpamAssassin (a mail filter that is used to identify spam messages). Each MTA supports various features providing default access control on the server.

MTA	Description
Sendmail	A standard MTA that supports various UNIX-based operating systems. It is designed to function in such a way that it runs as a single entity. It supports various MTAs and MDAs (Mail Delivery Agents). It is an MTA that is still used from earlier days.
Postfix	A fast and secure MTA that can be administered easily. It is similar to Sendmail but varies in its internal functionality. Unlike Sendmail, Postfix supports a modular functionality. It is a free, open MTA and can be used as an alternative to Sendmail.
Exim	A flexible and a freely available MTA. Unlike other MTAs, it has extensive features that allow the administrator to control the mail transfer through the system. The latest version has been developed into an ACL-based system, which provides more detailed and flexible controls. This helps in the integration of antivirus and anti-spam measures in the MTA.
qmail	The first secure mail transfer agent that manages large mailing lists. It is made up of several modules, which can be replaced by any of the new modules that contain the same interface. It introduces the concept of wild cards, which allows users to publish multiple mail addresses for mailing lists.

Sendmail Configuration Files

When you install Sendmail, configuration files are created in the **/etc/mail** directory. The **sendmail.cf** file is the main configuration file for Sendmail. Because the configuration file is large, it is better to avoid editing the file directly. Instead, you can make changes to the **sendmail.mc** file and later update the **sendmail.cf** file using the **m4** utility.

Postfix Configuration Files

Postfix configuration files are created in the **/etc/postfix** directory. The **main.cf** and the **master.cf** are the two main Postfix configuration files. The **main.cf** file defines the configuration parameters and the **master.cf** file defines the daemon processes. Most of the configuration parameters are set to their default values. The **master.cf** file primarily defines the process of a client program connecting to a service and a daemon running when a service is requested.

qmail Configuration Files

All qmail configuration files, except the **.qmail** file, are stored in the **/var/qmail/control** directory.

The **.qmail** file resides in the **~alias** directory of the user's home directory. The **.qmail** files are used to control the delivery of mail. They contain a list of delivery instructions represented by some special characters.

Exim Configuration Files

The configuration files of Exim are created in the **/etc/exim** directory. The **exim.conf** file is the main configuration file for Exim.

Sendmail Emulation Layer Commands

The smrsh program is a shell utility that restricts users from performing malicious actions and limits the programs that such users can execute. It acts as a replacement for **/bin/sh** in the program mailer definition for Sendmail and is installed in the **/usr/sbin/smrsh** directory. When the smrsh program is used along with Sendmail, Sendmail only executes the set of programs specified in the **smrsh** directory. The set of commands that the smrsh program allows Sendmail to execute are referred to as Sendmail emulation layer commands. Some of the interpreter programs prohibited from execution are sh, csh, perl, and sed.

The MDA

A **Mail Delivery Agent (MDA)** is a program that delivers incoming email messages to the intended recipient's mailbox. An MDA sends new mail messages using SMTP, and retrieves messages using POP3 or IMAP. It also distributes and sorts messages on a local machine so that an MUA can access them.

Figure 13-5: Email messages are delivered to clients using the MDA.

The MUA

The **Mail User Agent (MUA)** is a program used for reading and composing email messages. Also referred to as an email client application, the MUA acts as an interface between a user and an MTA and contains mailboxes for storing messages. MUAs can have either a graphical interface, such as Thunderbird® and Evolution Mail, or a text-based interface such as Mutt.

Figure 13-6: MUA acts as an interface between a user and an MTA.

Mail Forwarding

Mail forwarding is a feature that automatically forwards email messages. It can also redirect mail to one or more addresses.

The Email Process

The email process describes the sequence of steps involved in creating, transmitting, and storing messages. There are five stages involved in the electronic mailing process.

1. The sender composes the message to be sent. The sender's MUA formats the message in an email format and uses SMTP to send the message to the sender's MTA.
2. The sender's MTA checks for the destination address provided by SMTP. Based on the destination address, it sends a request to the Domain Name System (DNS) server to look for the specific domain name in the DNS server using User Datagram Protocol (UDP). The DNS server sends a response with a Mail Exchange (MX) record that lists the mail exchange servers supported by the domain.
3. Based on the response from the DNS server, the sender's MTA sends the message to the recipient's MTA using SMTP.
4. The recipient's MTA again sends the message to the recipient's MDA, which delivers the email to a spool where all the recipient's messages are stored.
5. The recipient's MUA retrieves the message from the spool through a retrieval agent known as the **Mail Retrieval Agent (MRA)** using protocols such as POP3 and IMAP. The recipient then opens the received message.

Figure 13-7: The process of sending an email message from one system to another.

How to Configure Email Services

Follow these general procedures to configure email services.

Install Postfix

To install Postfix:

1. Log in as **root**.
2. Navigate to the **/opt/linuxplus/Packages** directory.

Note: Normally, you would mount the CentOS installation media containing Postfix or access an online Yum repository, but

	the necessary files have already been copied to the /opt/linuxplus/Packages directory.
--	---

3. To install Postfix, at the command line, enter `yum localinstall postfix-{version}. {release}.x86_64.rpm`.
4. To verify that Postfix has been installed, enter `rpm -qi postfix`.

Configure Postfix for Incoming Email Messages

To configure Postfix for incoming email messages:

1. Log in as **root**.
2. Set the Fully Qualified Domain Name (FQDN).
 - a. Enter hostname *{Fully Qualified Domain Name}*.
 - b. Open the **/etc/sysconfig/network** file and ensure that the FQDN is updated.
3. Using the vi editor, open the **/etc/postfix/main.cf** file.
4. To receive mail messages from remote servers, set the system.
 - a. Locate the "inet_interfaces = all" directive and uncomment the line.
 - b. Locate the "inet_interfaces = localhost" directive and comment the line.
5. To specify a list of domains controlled by the Postfix mail server, locate the "mydestination" directive and uncomment the line. `mydestination = $myhostname, localhost, $mydomain, localhost, $mydomain, mail.$mydomain, www.$mydomain`.
6. Save and close the file.
7. To start the postfix service, enter `service postfix start`.
8. To verify that the postfix server is listening to all interfaces, enter `netstat -plt | grep master`.
9. To enable the postfix service during system startup, enter `chkconfig postfix on` or `systemctl enable postfix.service`.

Configure Postfix for Outgoing Email Messages

To configure Postfix for outgoing email messages:

1. Log in as **root**.
2. Navigate to the **/etc/postfix** directory.
3. Using the vi editor, open the **main.cf** file.
4. Masquerade the domain for outgoing email messages.
 - a. Locate the "myorigin" directive and uncomment it.
 - b. To masquerade the **root** user, type `masquerade_exceptions = root`.
5. Save and close the file.

6. To restart the **postfix** service, enter `service postfix restart` or `systemctl restart postfix`.

Configure Email Aliases Using Sendmail or Exim

To set an email alias using Sendmail or Exim:

1. Using the vi editor, open the **/etc/aliases** file.
2. To add email aliases, enter `{user name}:{alias1}, [alias2, alias3,...]`. An alias can be a local user name or a file name, a command, an include file, or an external address.
 - To add an alias for a user name that exists on the local system, enter `{user name}:{alias name for the user}`.
 - To add an alias for an external email address to which you want the message to be forwarded, enter `{alias name}:{email address}`.
 - To add an alias for a list of users to whom you want the mail to be forwarded, specify the list of users in a separate file, `{user name} :include: {path to the file containing user names}/{file name}`.
 - To add a file name to which you want to append the messages received by a user, enter `{user name}:{path to the file}/{file name}`.
 - To add an alias to enable Sendmail process commands that receive messages from the standard input, enter `{user name}:<{command}`.
3. Save and close the file.
4. To update the aliases database file, enter `newaliases`.

Configure Email Aliases Using Postfix

To configure email aliases using Postfix:

1. Using the vi editor, open the **/etc/postfix/main.cf** file.
2. Verify whether the `alias_maps` variable is defined with the location of the aliases file as follows, `alias_maps = hash:/etc/aliases`.
3. Save and close the file.
4. Using the vi editor, open the **/etc/aliases** file.
5. Add an alias for a user name that exists on the local system by entering `{user name}:{alias name for the user}`.
6. Save and close the file.
7. To update the aliases database file, enter `newaliases`.
8. To restart the postfix service, enter `systemctl restart postfix`.

Configure Local Aliases

To configure local aliases:

1. Log in as **root**.
2. Using the vi editor, open the **/etc/postfix/main.cf** file.
3. Locate the `alias_maps` directive and verify whether the directive is defined with the location of the aliases file, in the format `alias_maps = hash:/etc/postfix/aliases`.
4. Locate the `alias_database` directive and verify whether the directive is defined with the location of the aliases file, in the format `alias_database = hash:/etc/postfix/aliases`.
5. Save and close the file.
6. Using the vi editor, open the **/etc/postfix/aliases** file.
7. To add email aliases, enter `{user name}:{alias1}, [alias2, alias3]`.
 - To add an alias to a user name that exists on the local system, enter, `{user name}:{alias name for the user}`.
 - To add an alias to an external email address to which the mail is to be forwarded, enter `{user name}:email address`.
 - To add an alias to a file that contains the list of users to whom the mail messages are to be forwarded, enter `{user name}:include:{path to the file containing user names}{{file names}}`.
 - To add an alias to the Sendmail command that receives messages from the standard input, enter `{user name}:{command}`.
8. Save and close the file.
9. To update the alias database file, enter `postalias /etc/postfix/alias`.
10. To restart the postfix service, enter `systemctl restart postfix`.

Configure Virtual Aliases

To configure virtual aliases:

1. Log in as **root**.
2. Configure the prerequisite settings for virtual alias.
 - a. Using the vi editor, open the **/etc/postfix/main.cf** file.
 - b. Locate the `mydestination` directive and define the domains that will be controlled by the mail server by entering `mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain, mail.$mydomain, www.$mydomain, /etc/postfix/ mydestination`.
 - c. Save and close the file.
 - d. Using the vi editor, open the **/etc/postfix/mydestination** file.
 - e. To specify the list of domains that are controlled by the postfix, type *List of domains*. A sample **mydestination** file would be `develetech.tld develetech.test`
`develetech.example`.
 - f. Save and close the file.
3. Using the vi editor, open the **/etc/postfix/main.cf** file.

4. Define the `virtual_alias_maps` variable below the `alias_maps` variable by entering `virtual_alias_maps = hash:/etc/postfix/virtual_maps`.
5. Save and close the file.
6. Using the vi editor, open the **`/etc/postfix/virtual_maps`** file.
7. To map email addresses to user names, enter `{user name@domain name}{user name}`.
8. To map email messages destined to a domain to a specific user in another domain, enter `@{domain name}{user name@domain name}`.
9. To map email messages destined to a domain to another domain, enter. `{user name}@{domain name1}{user name@domain name2}`.
10. Save and close the file.
11. To rehash the virtual alias file, enter `postmap virtual_maps < virtual_maps`.
12. To restart the postfix service, enter `systemctl restart postfix`.

Configure Outbound Address Rewriting

To configure outbound address rewriting:

1. Log in as **root**.
2. Navigate to the **`/etc/postfix`** directory.
3. Using the vi editor, open the **`main.cf`** file.
4. To enable outbound address rewriting, below the definition of the `alias_maps` directive, type `smtp_generic_maps = hash:/etc/postfix/generic_map`.
5. Save and close the file.
6. Using the vi editor, create a file, **`generic_map`**.
7. Type `{user name@domainname1} {user name@domainname1}`.
8. Save and close the file.
9. To rehash the file, enter `postmap generic_map < generic_map`.
10. To restart the postfix service, enter `systemctl restart postfix`.

Forward Incoming Mail Messages to a Different Address

To forward incoming mail messages to a different address:

1. Log in as user.
2. Enter vi `.forward`.
3. Type the email address to which the incoming mail messages should be forwarded.
4. Save and close the file.
5. Verify that the messages have been forwarded to the address specified in the **`~/forward`** file by sending an

email to the user who has the forward file.

- a. Enter mail *{user}*.
- b. Specify a subject for the email and press **Enter**.
- c. Specify content for the email. In the last line, type a period (.) and press **Enter**.
- d. Specify the Carbon Copy recipients and press **Enter**.
- e. Log out and log in as the user mentioned in the *~/forward* file.
- f. Enter mail to view the mailbox.
- g. Enter the mail number to read the email.
- h. If necessary, to delete the email, enter d.
- i. To save and quit the mailbox, enter q.

Manage a Mail Queue

To manage a mail queue:

1. Send an email message.
 - a. Log in to the CLI as **root**.
 - b. Enter mail *{user name}@{domain name}*.
 - c. Enter a subject for the message.
 - d. Enter content for the message.
 - e. On a separate line, enter just a period.
2. Enter mailq to check whether the email message has been sent or is in the queue.
3. If necessary, enter `rm -f /var/spool/mqueue/*` to remove the mail messages in the queue.

Note: The mailq command is used to display the mail queue.

Configure Postfix Restrictions

To configure postfix restrictions:

1. Log in as **root**.
2. Using the vi editor, open the */etc/postfix/main.cf* file.
3. To define restrictions for the sender, at the end of the file, enter `smtpd_sender_restrictions = check_sender_access hash:/etc/postfix/ sender_access, [comma separated list of zero or more mail sender restrictions]`.
4. To define restrictions for the client, enter `smtpd_client_restrictions = check_sender_access hash:/etc/postfix/client_access, [comma separated list of zero or more mail client restrictions]`.

5. To define restrictions for the recipient, enter `smtpd_recipient_restrictions = check_sender_access hash:/etc/postfix/recipient_access, [comma separated list of zero or more mail client restrictions]`.
6. Save and close the file.
7. Navigate to the **/etc/postfix** directory.
8. Open the sender, recipient, or client access file.
9. Define the restrictions.
 - To allow relaying, type `{domain name} RELAY`.
 - To accept mail messages from the domain specified, type `{domain name} OK`.
 - To reject the mail messages sent from the domain specified, type `{domain name} REJECT`.
 - To reject the mail from the sender and display an arbitrary message, type `{domain name} ERROR:###message}`.
 - To discard messages after accepting them, type `{domain name} DISCARD`.
10. Save and close the file.
11. To build the respective access database, enter `postmap {sender or recipient or client access file} < postmap {sender or recipient or client access file}`.
12. To restart the **postfix** service, enter `systemctl restart postfix`.

TOPIC B Control Internet Services

In the last topic, you configured email services. There are numerous services and benefits that the Internet offers. This has made the Internet a necessity in almost all organizations. In this topic, you will control Internet services.

Internet services are crucial for the effective functioning of businesses. Any disruption or problems in Internet services could paralyze regular work and cost organizations dearly. As a Linux system administrator, it is your primary responsibility to ensure that Internet services are running without any problems all the time.

The xinetd Daemon

The **xinetd** daemon controls system services on a network and is based on the client-server architecture. Various services are listed in the configuration files of xinetd. When the daemon receives a request from a port for a particular service, it checks with the configuration files and then starts the appropriate server containing the service. This enables faster service management over the network because requests are handled immediately. Services that are managed by xinetd include file sharing, Telnet, rsync, and time management. The settings for these services can be controlled using the `/etc/xinetd.conf` file or the service-specific files found in the `/etc/xinetd.d` directory.

Figure 13-8: The xinetd daemon controls services on clients and servers.

Telnet

Telnet is a terminal emulation protocol that enables a user on a site to simulate a session on a remote host. In other words, it allows a user to log in to another computer over a network. It does this by translating keystrokes from the user's terminal into instructions recognized by the remote host, and then carrying the output back to the user's terminal and displaying it in a format native to the remote host. This service is transparent; it gives users the impression that their terminals are directly attached to the remote host. The remote computer needs to have a Telnet server, and the user's computer needs to have a Telnet client.

Figure 13-9: A Telnet client communicating with a remote Telnet server.

Disadvantages of Telnet

Telnet is not a very secure protocol because it transmits passwords as plain text. Telnet has largely been replaced by Secure Shell (SSH) because it offers better security, and is no longer recommended for remote access.

The /etc/xinetd.conf File

The **/etc/xinetd.conf** file is the configuration file for the xinetd daemon and contains a list of services managed by the daemon. It is referenced by the daemon each time a request is sent for a service to be started or stopped. The file is divided into different parts, one for each service, and each containing the service's settings. Services can be single threaded or multithreaded.

Figure 13-10: The xinetd.conf file with its various configuration settings.

The `/etc/xinetd.conf` file can be configured according to user requirements using a number of options. Some of the frequently used options are listed in the following table.

Option	Enables You To
service	Specify the service name.
wait	Specify whether the service is single threaded or multithreaded.
user	Specify the User ID (UID) for the process running on the server.
server	Specify the executable that is to be launched on the server when the service is invoked.
disable	Specify whether a service is enabled or disabled.

Syntax

The syntax for specifying services in the xinetd configuration files is `{attribute} {assignment operator} {value}`.

The libwrap.so Libraries

The `libwrap.so` files are libraries that are linked to services controlled by the xinetd daemon. There are three main library files: `libwrap.so`, `libwrap.so.0`, and `libwrap.so.0.7.6`. These files, found in the `/usr/lib` directory, control the TCP services and additional network access settings, including the settings found in the network access configuration files.

The /etc/xinetd.d Directory

The `/etc/xinetd.d` directory contains configuration scripts for services managed by xinetd. Some of the services listed in this directory are FTP, Kerberos, and rsync. Each service can be individually configured using the configuration file found in this directory. The options in the service configuration files are similar to those found in the `/etc/xinetd.conf` file.

Figure 13-11: The *xinetd.d* directory contains configuration files for *xinetd*-managed services.

xinetd Access Controls

The *xinetd* daemon has a set of functions for controlling access to services managed by it. These access controls are of two types: host-based and time-based.

Access Control Type	Description
Host-based	Host-based access controls are implemented by restricting hosts in the <i>xinetd</i> service configuration file through the <i>no_access</i> and <i>only_from</i> options. Host Pattern Access Controls are used to specify host patterns when host-based access controls are implemented. Hosts may be specified in the form of their IP addresses, netmask ranges, network names, or hostnames.
Time-based	Time-based access controls are implemented by adding the <i>access_times</i> option to the service configuration file.

Service and Application Access Controls

Service access controls and *application access controls* are daemons used for restricting access to certain important services and applications, such as those that control network connections and security policies. Examples of these daemons are *squid* and *httpd*. These daemons restrict access by referring to the hostnames or IP addresses of systems listed in the *libwrap.so* file or the *xinetd* configuration files. If systems have the required permission, the daemons will permit them to access the protected services and applications.

How to Control Internet Services via xinetd

Follow these general procedures to control Internet services by using *xinetd*.

Configure Services Managed by xinetd

To configure services managed by *xinetd*:

1. Log in as **root**.
2. To list the services managed by the *xinetd* daemon, enter `ls -l /etc/xinetd.d/`.

<p>Note: In the older versions of Linux, the <i>/etc/xinetd.d/*</i> file was referred to as the <i>/etc/inetd.d/*</i> file.</p>
--

3. Manage the *xinetd* daemon at system startup.

- To make sure that the xinetd daemon is configured to start in the desired runlevel, enter `chkconfig --list xinetd`.
- To update the runlevel configuration of the xinetd daemon, enter `chkconfig --level {levels} xinetd {on|off}`.
- To manage the status of the xinetd daemon, enter `service xinetd {start|stop| status|restart|reload}`.

4. Manage services controlled by the xinetd daemon.

- To enable or disable the specified service at system startup, enter `chkconfig {service name} {on|off}`.
- Manage the service using configuration files found in the **/etc/xinetd.d** directory.
 - a. To edit the service-specific configuration file, enter `vi /etc/xinetd.d/{service configuration file}`.
 - b. To disable or enable the service, edit the "disable = {yes|no}" line.
 - c. To define the service-specific settings that override the global settings for all services managed by the xinetd daemon, edit `<attribute> <assignment operator> <value> <value>`.
 - d. Save and close the file.
 - e. To reload the xinetd daemon with the applied changes, enter `service xinetd reload`.

Define the Global Settings in the /etc/xinetd.conf File

To define the global service settings in the **/etc/xinetd.conf** file:

1. Log in as **root**.
2. To edit the xinetd daemon configuration file, enter `vi /etc/xinetd.conf`.
3. To define the global settings for all services managed by xinetd daemon, edit `{attribute} {assignment operator} {value} {value}`.
4. Save and close the file.
5. To apply the changes, enter `service xinetd restart`.

Configure Access Control for xinetd Managed Services

To configure access control for xinetd managed services:

1. Log in as **root**.
2. Define service-specific access control in the **/etc/xinetd.conf** file.
 - a. To edit the service-specific configuration file, enter `vi /etc/xinetd.d/{service configuration file}`.
 - b. To define the remote hosts for which the particular service is available, type `only_from = {list of IP addresses}`.
 - c. To define the remote hosts for which the particular service is unavailable, type `no_access = {list of IP addresses}`.
 - d. To define the time intervals when the service can be accessed, type `access_times = {hour:min-hour:min}`.
3. Save and close the file.
4. To apply the changes and restart the xinetd daemon, enter `service xinetd restart`.

Monitor Sendmail, Postfix, or Qmail Mail Servers Using Telnet

To monitor Sendmail, Postfix, or Qmail mail servers using Telnet:

1. To connect to the SMTP server on port 25, enter telnet *{server's IP address}* 25.
2. Send an email message using Telnet and observe the results to ensure that the SMTP server is properly functioning.
 - a. To specify your hostname, enter HELO *{server name}*.
 - b. To specify the sender's email address, enter MAIL FROM: *email address*.
 - c. To specify the recipient's email address, enter RCPT TO: *email address*.
 - d. Enter DATA.
 - e. Enter the subject and message. Terminate the message with a period.

ACTIVITY 13-1

Configuring Basic Internet Services Review

Scenario

Answer the following review questions.

1. Which mail retrieval protocol would you configure in your organization? Why?
2. What services might you configure using Systemd in your environment?

Summary

In this lesson, you configured email services and controlled Internet services. Now, you will be able to implement a cost-effective solution for communication, sharing, and transfer of resources across a network.

14 Securing Linux

Lesson Time: 1 hour

Lesson Introduction

In the last lesson, you configured basic Internet services. All networked services, no matter how basic, need to be secured. Poor security can lead to damage or loss from disgruntled employees, hackers, or competitors. In this lesson, you will secure a Red Hat® Enterprise Linux® system connected to a network.

To properly secure a Linux system, an administrator has to understand how different threats affect the system. Specific security measures that allow the administrator to control the transfer of sensitive data and restrict unauthorized users from accessing the network need to be implemented.

Lesson Objectives

In this lesson, you will implement measures to secure a Linux system. You will:

- Implement basic system security and encryption services.
- Secure user accounts.

TOPIC A Implement Basic System Security

In the previous lesson, you worked with the Internet and email services. These services are prone to threats from hackers, which can cause serious problems. A security breach in an organization's network cannot be compromised because sabotage or data theft could affect business badly. These instances can be avoided by securing the systems and sending data in a secured format that can be comprehended only by the user to whom the data is directed. In this topic, you will examine the basics of system security.

Computer security is a critical part of business strategy, and organizations continually demand new levels of protection. On a network, there are various web and mail services that can be implemented.

Though these services facilitate data transfer, there is always the risk of data theft associated with them. Without proper security and encryption mechanisms governing these services, transferring sensitive data securely is impossible.

Keys

While logging in to a remote machine, the private and public keys are combined by the remote server for verification. A user can log in and transfer files only if the keys match. Authenticity is established by the remote server, which then grants the necessary permissions.

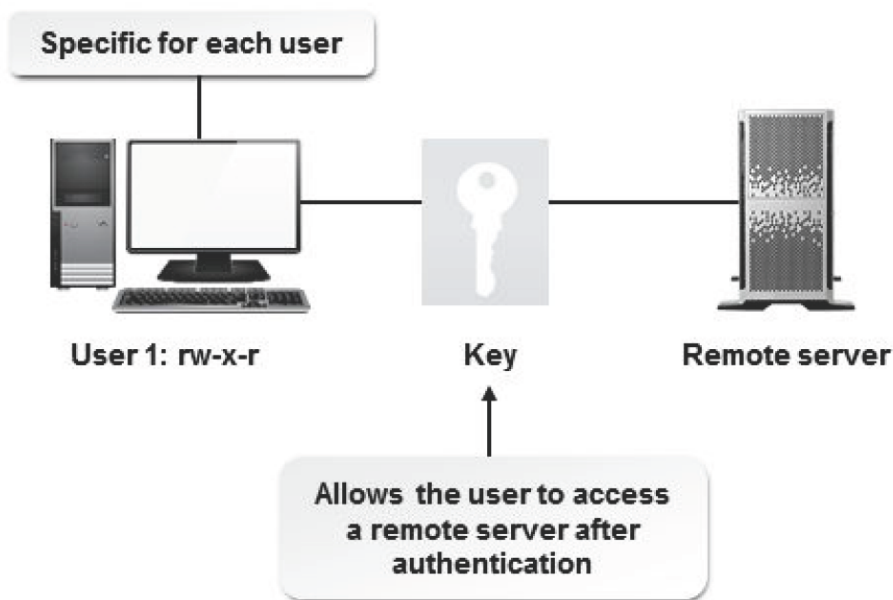


Figure 14-1: Securing data using a key.

Keys are of two types: public and private.

Key Type	Description
Public	A public key is the key that is transmitted to the destination server along with the request. This is compared with the private key of the destination. Only when these two are related, the user is authenticated.
Private	A private key is retained on the local system and is not transmitted to the destination server. Data is sent to the user along with the sender's public key. The user can access data only when the private key of the user is authenticated by the public key of the sender.

Authentication

Authentication verifies that users are who they say they are. In data communication, authenticating a sender is necessary to verify that the data came from the right source. A receiver is authenticated as well, to verify that the data is going to the right destination.

There are several methods for ensuring authentication.

Method	Description
The known_hosts file	When you connect to a remote host, the host sends your public host and server keys for authentication. Your system looks up the known_hosts file to locate an entry for the host's keys, and if an entry is found, you will be granted access.
The SSH server	A server that automatically generates public host and server keys for authentication purposes.
Kerberos	A network authentication service that is used by client/server applications. Kerberos creates a key, or ticket, for each user logging in to the network. The tickets are embedded along with the message to identify the sender.

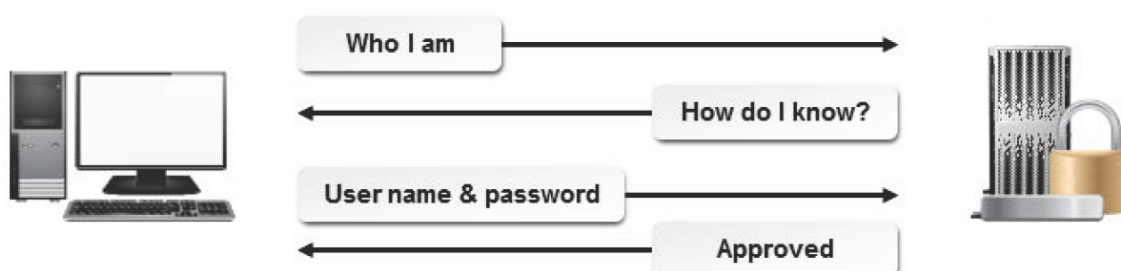


Figure 14-2: The authentication process.

Authentication Factors

The number of factors that are used to show the identity of a user through authentication determines how effective authentication can be. The three types are:

- One-factor authentication—which provides what you know, such as a password or Personal Identification Number (PIN). It is based on recalling a piece of information from your memory or by writing the information down. This type of authentication is the least effective.
- Two-factor authentication—which combines what you have with what you know. For example, your ATM card combined with the PIN provides access to your bank account.
- Three-factor authentication—which also provides proof of the user’s identity through biometrics. It uses physiological identification characteristics such as fingerprints, voice recognition, or signature recognition. This is the best type of authentication.

Biometric Authentication

To increase the level of reliability of systems and ease of use to users (beyond password authentication), biometric authentication can be introduced. When this type of system is added to an authentication scheme, it is considered to be a strong authentication. The designation of strong is given because a user is not only identified digitally, but by his or her physiological characteristics, through fingerprint scanning, iris scanning, or hand geometry.

Encryption

Encryption is a method of controlling user access to information by configuring data to appear as codes that cannot be interpreted by unauthorized users. To authorized users, though, this data appears in its original form. Passwords can also be used to protect data along with encryption.

Various protocols, such as Secure Shell (SSH), Secure Socket Layer (SSL), and SECURENET, are used to implement encryption on a network.

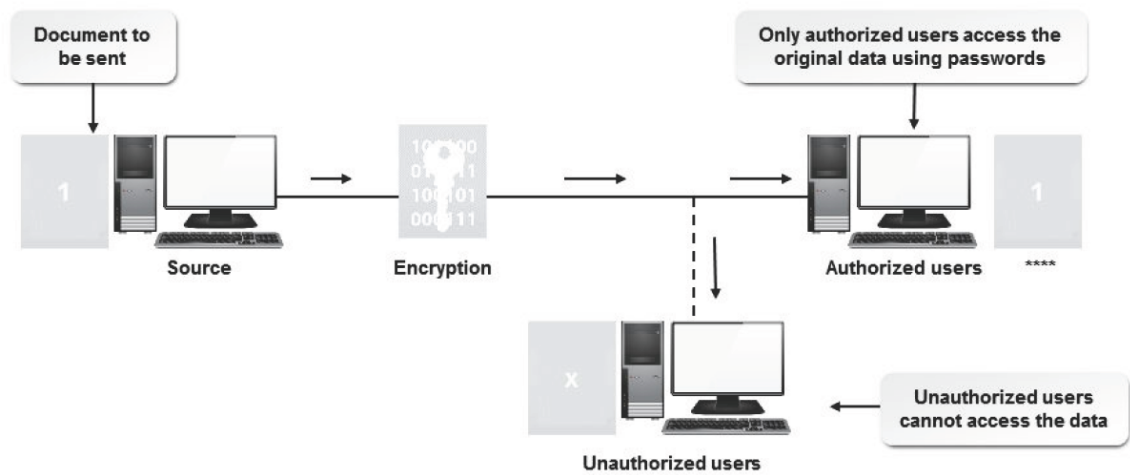


Figure 14-3: Securing data using encryption.

Encryption Solutions

Authorized users of encrypted computer data must have the key that was used to encrypt the data in order to decrypt it. Different solutions are available for encrypting data using specific **algorithms**.

Solution	Description
----------	-------------

Solution	Description
Blowfish	<p>A symmetric block cipher that provides strong encryption and uses key sizes up to 56 bytes (a 448-bit key). Its features include:</p> <ul style="list-style-type: none"> • Strong key support, handling, and cryptography. • Security to wipe files and clear empty disk space.
3DES	<p>A block cipher algorithm (pronounced "triple dez") that can encrypt and decrypt data using a secret key. 3DES uses three stages of Data Encryption Standard (DES), making it a very secure option. Anything encrypted by DES encryption has 72,000,000,000,000,000 (or 72 quadrillion) possible keys.</p>
MD5	<p>Message-Digest algorithm 5 (MD5) is a command-line utility that generates and verifies message digests (digital signatures) using the MD5 algorithm.</p> <p>The MD5 algorithm is intended for digital signature applications, where a large file must be compressed in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem. MD5 is also used to check the integrity of files.</p>
SHA-2	<p>Secure Hash Algorithm version 2 (SHA-2) is a set of cryptographic hash functions designed by the U.S. National Security Agency (NSA) that includes the commonly used SHA-256 and SHA-512 hash functions. The SHA-2 hash functions are also intended for digital signature applications.</p>

Random Number Generation

Random number generation is an encryption method in which the kernel is used to generate random numbers that are assigned to files before transfer. Only when the numbers are matched by the recipient is the transfer completed. The algorithm that governs random number generation is the **Pseudo Random Number Generation (PRNG)** algorithm. In Linux, the kernel files, **/dev/random** and **/dev/urandom**, act as random number generators. Using the concept of permutations and combinations, these kernels are able to generate numbers with millions of digits from a single source number.

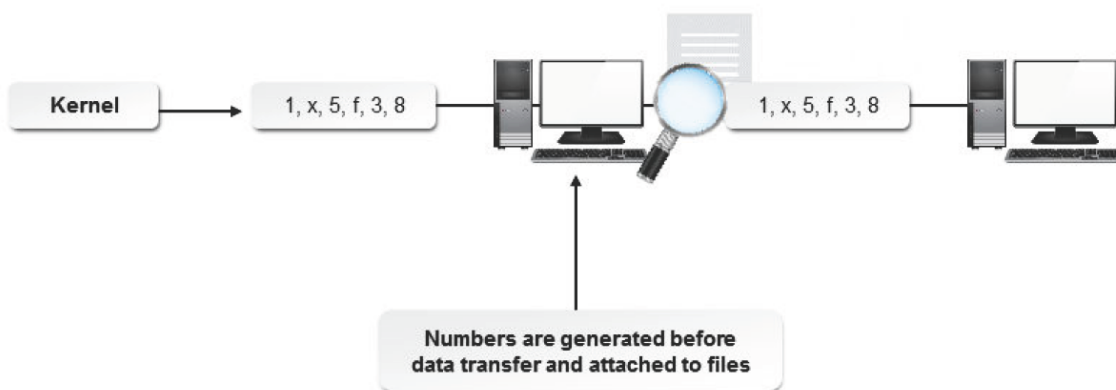


Figure 14-4: Securing data using the random number generation method.

Cryptographic Hashes

Cryptographic hashes are used in an encryption method in which arbitrary data is encapsulated within a fingerprint that is attached to a file. A fingerprint is a fixed string called the hash value, checksum, or message digest. The data in the file can be checked for authenticity by verifying the hash value. When modifications are made to a file, its hash value also changes. Various algorithms, such as MD5, SHA-1, and SHA-2 (SHA-256 and SHA-512), are used to implement cryptographic hash functions.

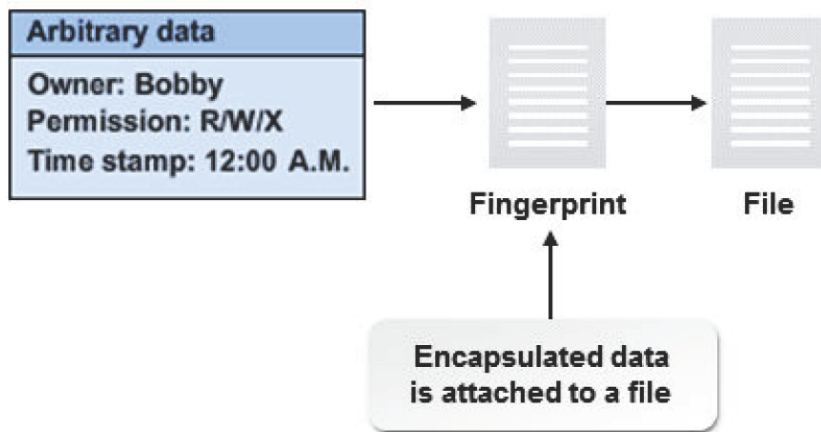


Figure 14-5: Securing data using the cryptographic hashing method.

Various utilities are used to check the hash values of files.

<i>Utility</i>	<i>Enables You To</i>
sha1sum	Compute and check files with SHA-1 checksums. The syntax of this utility is <code>sha1sum --check {file name}</code> .
sha256sum	Compute and check files with SHA-256 checksums (SHA-2 family with a digest length of 256 bits). The syntax of this utility is <code>sha256sum --check {file name}</code> .
sha512sum	Compute and check files with SHA-512 checksums (SHA-2 family with a digest length of 512 bits). The syntax of this utility is <code>sha512sum --check {file name}</code> .
md5sum	Check files for improper MD5 checksums. The syntax of this utility is <code>md5sum --check {file name}</code> .

Symmetric Encryption

Symmetric encryption is carried out using only a single key, which is used for both encryption and decryption.

Various utilities are used to perform symmetric encryption.

<i>Utility</i>	<i>Used To</i>
passwd	Change the login password. Users who are logged in can change only their login password, and not that of other users. However, this does not apply to the root user. When you type the <code>passwd</code> command at the command prompt, you are asked to enter your current password and then the new password you want to set. The new password is effective the next time you log in to the system.
gpg	Encrypt messages using the GNU Privacy Guard (GnuPG) encryption system. This utility has various commands and options. The syntax of this utility is <code>gpg [options] {command} {arguments}</code> .
openssl	Encrypt and decrypt messages using the SSL protocol through creation of keys, certificates, and signatures. The syntax of this utility is <code>openssl {command} [options] {arguments}</code> .

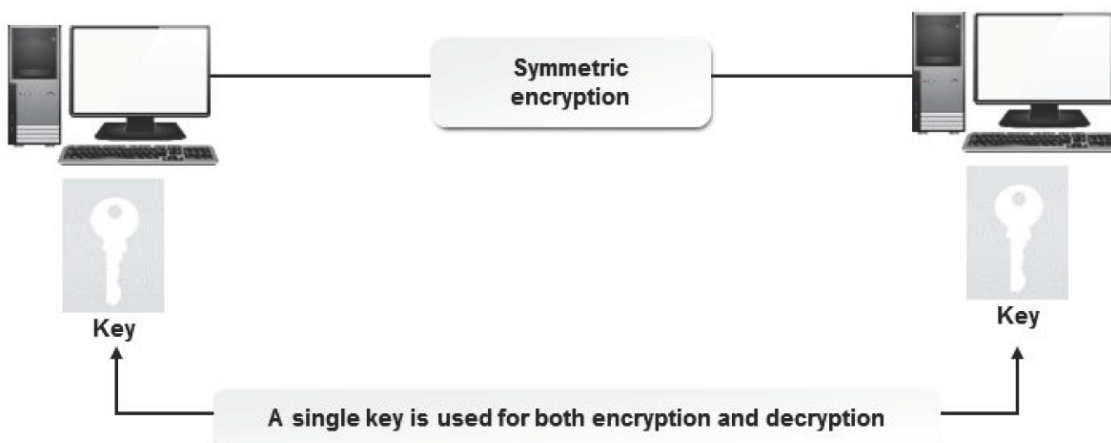


Figure 14-6: Key generation in symmetric encryption.

Asymmetric Encryption

Asymmetric encryption is carried out using two keys in the form of key pairs. While one key is used for encryption, the other key is used for decryption.

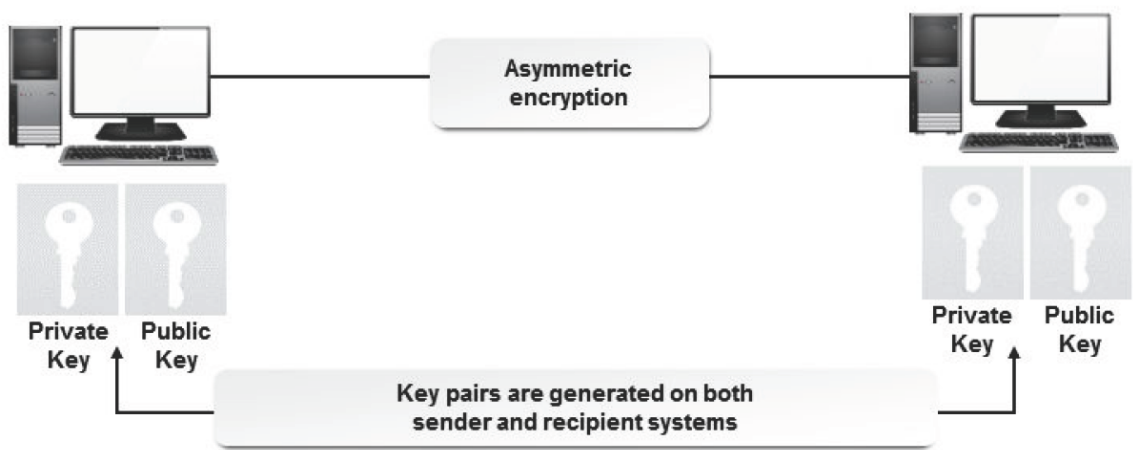



Figure 14-7: Key generation in asymmetric encryption.



Note: The SSH protocol is a common example of asymmetric encryption.

There are two main protocols involved in asymmetric encryption: protocol 1 and protocol 2. Only one protocol can be implemented at a time.

Protocol	Description
Protocol 1	<p>Only one key is exchanged between a sender and a recipient.</p> <ol style="list-style-type: none">The recipient first generates a key pair, which contains a public key and a private key.The request is then sent to the sender in an encrypted form along with the public key of the recipient. The private key is retained with the recipient.The sender receives the request, authenticates the public key sent by the recipient, and then returns the requested information in the encrypted form along with the public key of the recipient.The recipient receives the information along with its public key and decrypts it by authenticating the public key with its private key.
Protocol 2	<p>Digital signatures are used along with key encryption. A digital signature is a unique ID created when a message digest of the sender is encrypted.</p> <ol style="list-style-type: none">The sender first generates a key pair, which contains a public key and a private key. Using the private key and the message digest, the sender generates a digital signature.The public key is transmitted to all the systems on a network. The request is then sent by the recipient to the sender.The sender receives the request and returns the information in an encrypted form along with its digital signature.The recipient receives the information and decrypts it by authenticating the digital signature of the sender with its public key.

Rogue Public Keys


Rogue public keys are generated by unauthorized users to bypass public key cryptography. These are used to decrypt information that they are not supposed to access. Generation of rogue keys can be prevented by using

public key fingerprints, forming trusted groups, and issuing digital certificates through trusted certificate authorities.

Digital Certificate Types

A digital certificate is a method of symmetric encryption. Two main types of digital certificates are available.

Certificate Type	Description
Certificate Authority	Certificate Authority certificates are generated by a common and trusted Certificate Authority (CA) on receiving a certificate signature request (csr). The advantage of using this method is that generation of rogue digital certificates can be prevented.
Self-signed	Self-signed certificates are generated by users themselves and contain the public key of the user as the signature. Therefore, any user can create a self-signed certificate. However, this certificate does not provide guarantee about the identity of the user or the organization.



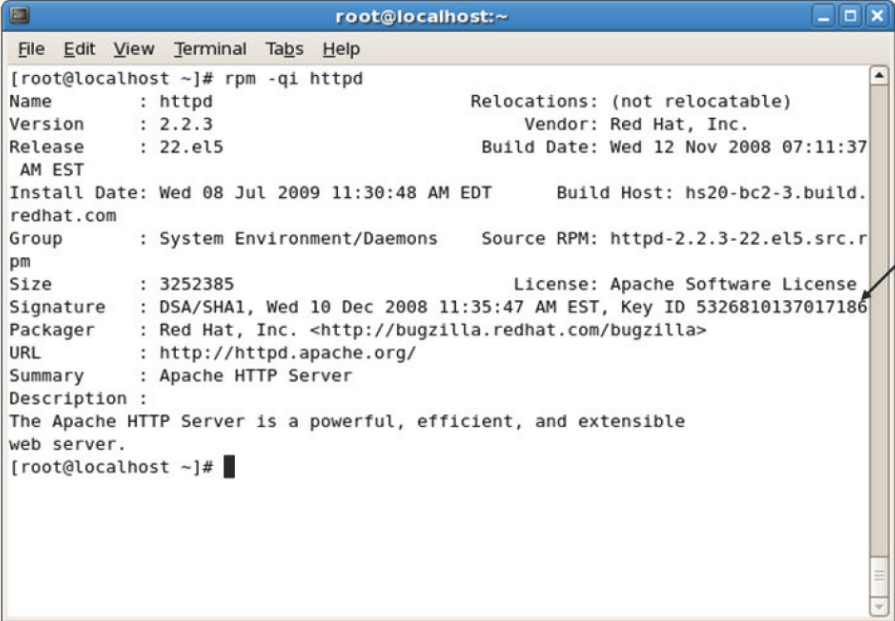
Note: The X.509 format is a standard format for public key certificates.

Package Integrity

Each package in Linux is assigned a public key, which is installed along with the package. **Package integrity** is the method of checking packages for these public keys to ensure that the package has come from a trusted vendor. It is necessary to perform a package integrity test before installing a package because installing a package from an unreliable source may lead to improper installation and virus attacks. The yum command always installs packages along with their public keys from the Red Hat online repository.

The rpm command can be used to check file integrity.

Command	Enables You To
rpm --verify {package name}	Check whether or not the package is installed.
gpg --import /etc/pki/rpm-gpg/RPM-GPG- KEY-*	Import public keys to the rpm database.
rpm --checksig {package name}	Check whether the package has valid signatures.
rpm --addsign {package name}	Assign valid signatures to the package.



```
root@localhost:~  
[root@localhost ~]# rpm -qi httpd  
Name       : httpd                      Relocations: (not relocatable)  
Version    : 2.2.3                      Vendor: Red Hat, Inc.  
Release    : 22.el5                     Build Date: Wed 12 Nov 2008 07:11:37  
          : AM EST                       Build Host: hs20-bc2-3.build.  
Install Date: Wed 08 Jul 2009 11:30:48 AM EDT  
redhat.com  
Group      : System Environment/Daemons  Source RPM: httpd-2.2.3-22.el5.src.r  
pm  
Size       : 3252385                     License: Apache Software License  
Signature  : DSA/SHA1, Wed 10 Dec 2008 11:35:47 AM EST, Key ID 5326810137017186  
Packager   : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>  
URL        : http://httpd.apache.org/  
Summary    : Apache HTTP Server  
Description:  
The Apache HTTP Server is a powerful, efficient, and extensible  
web server.  
[root@localhost ~]#
```

Public key installed with the package

Figure 14-8: Package information displaying the installed public key ID.

RADIUS

Remote Authentication Dial In User Service (RADIUS) is a client/server protocol that facilitates centralized authentication, authorization, and accounting over a network. User information is stored in a central database, which is shared by all remote servers. When a user sends a request to access a service or resource, RADIUS enables remote access servers to communicate with a central server to verify the authenticity of the user. It usually utilizes the user's login and password, stored in the `/etc/passwd` file on the server, to verify the user's credentials. It allows secure transmission of passwords by encrypting them with the MD5 algorithm. The RADIUS server is widely used by Internet Service Providers (ISPs).

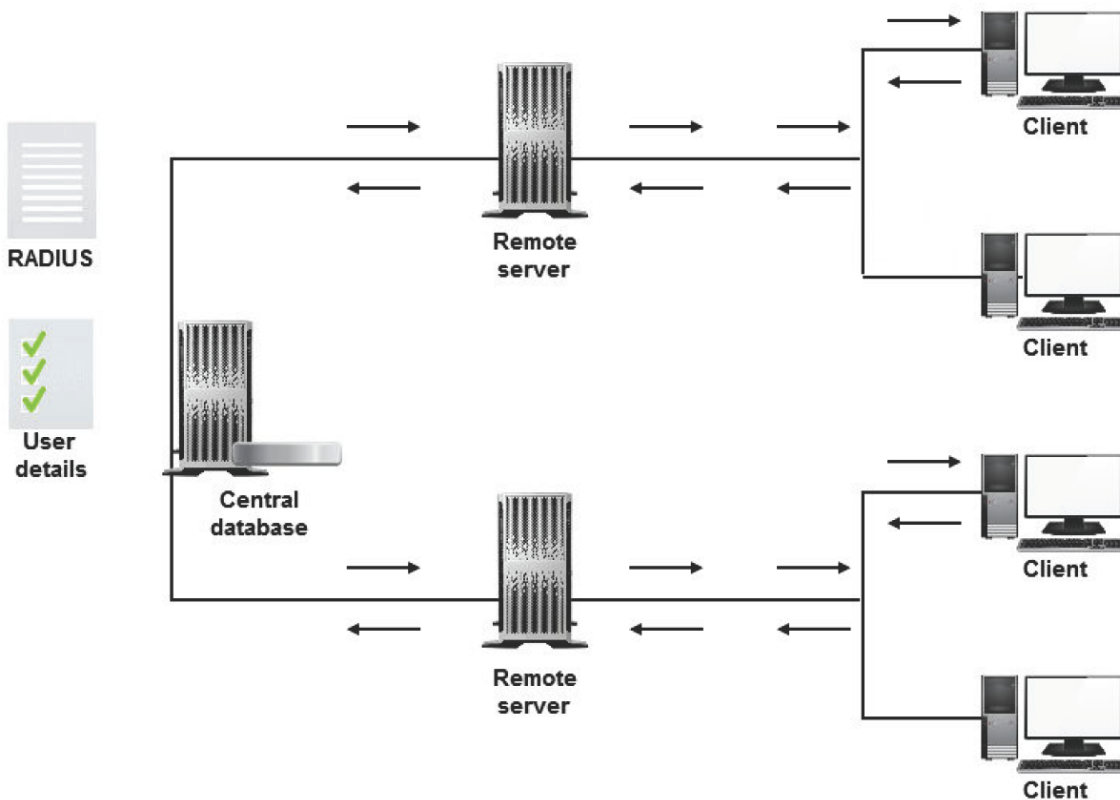


Figure 14-9: The RADIUS protocol providing authentication services on a network.

The RADIUS Server as a Network Access Server

When a network contains several remote access servers, you can configure one of the servers as a RADIUS server and all other servers as RADIUS clients. The RADIUS clients will pass all authentication requests to the RADIUS server for verification. User configuration, remote access policies, and usage logging can be centralized on the RADIUS server.

TCP Wrappers

TCP wrappers are protection layers that define the host computers that are allowed to connect to some network services and those that are not. The TCP wrappers package consists of the `/lib/ libwrap.so.0` library. A TCP wrapped service is compiled using the `libwrap.so.0` library. TCP wrappers operate separately from the network services protected by them.

Figure 14-10: TCP wrappers with protection layers.

How to Implement Encryption Services

Follow these general procedures to implement encryption services.

Configure Encryption

To configure encryption:

1. Log in as **root**.
2. To open the file, enter `vi /etc/{application}/{configuration file}`.
3. Locate the line that contains the encryption details.
4. If necessary, uncomment the line.
5. Modify the line to change the type of encryption.
6. Save the file and exit from the editor.
7. Restart the daemon service of the application or device to apply the changes.

Implement Cryptographic Services

To implement cryptographic services:

1. Log in as **root**.
2. Implement cryptographic services.
 - To generate a random number, enter `openssl rand {options}{number of pseudo-random bytes}`.
 - Generate a message digest.
 - To compute and check the SHA-1 message digest, enter `sha1sum [options] {file name}`.

- To compute and check the MD5 message digest, enter `md5sum [options] {file name}`.
- To output the message digest of a supplied file, enter `openssl dgst {message digest options} {file name}`.
- Implement symmetric encryption using the `gpg` command.
 - To encrypt a file with symmetric encryption, enter `gpg --symmetric {file name}`.
 - To decrypt an encrypted file using the `gpg` command, enter `gpg -d {file name}`.

	Note: <code>~/.gnupg</code> is the directory that stores the private and public keys created using the <code>gpg</code> command.
--	---

- Implement encryption using the `openssl` command.
 - a. To encrypt the file and store the output in another file, enter `openssl enc -e -salt -{bf|des3|cast5-cbc} -in {absolute or relative path of file to be encrypted} -out {absolute or relative path of encrypted file}`.
 - b. Enter the symmetric encryption password when prompted.
 - c. To decrypt the file and store the decrypted data in another file, enter `openssl enc -d -{bf|des3|cast5-cbc} -in {absolute or relative path of encrypted file} -out {absolute or relative path of decrypted file}`.
 - d. Enter the encryption password when prompted.

Verify Package Integrity Using the `rpm` Command and GPG

To verify package integrity using the `rpm` command:

1. Log in as **root**.
2. Verify package integrity.
 - To report files that differ from the original `rpm` version, enter `rpm --verify {package name}`.
 - To import GPG keys to add them to packages, enter `rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS*`.
 - To check whether the `rpm` package was packaged by CentOS, enter `rpm --checksig {package filename}`.

Generate Digital Certificates

To generate digital certificates:

1. Log in as **root**.
2. Generate keys using the `openssl` command.
 - a. To generate a public or private key pair, enter `openssl genrsa -out {absolute or relative path of key file} {key size}`.
 - b. To generate a certificate signature request, enter `openssl req -new -key {absolute or relative path of key file} -out {absolute or relative path of certificate request file}`.
 - c. Enter required information such as the country name, state, city, organization name, organizational unit name, server name, email, and if necessary, a challenge password and the optional company name.

- d. To generate a self-signed certificate, enter `openssl req -new -key {absolute or relative path of key file} -out {absolute or relative path of certificate file} -x509`.
 - e. Enter required information such as the country name, state, city, organization name, organizational unit name, server name, email, and if necessary, a challenge password and the optional company name.
 3. Generate digital certificates using the make command.
 - a. To generate a public or private key pair, enter `make -C /etc/pki/tls/certs {file name}.key`.
 - b. To generate a certificate signature request, enter `make -C /etc/pki/tls/certs {file name}.csr`.
 - c. To generate a certificate, enter `make -C /etc/pki/tls/certs {file name}.crt`.
 - d. To generate a key and a certificate in one file, enter `make -C /etc/pki/tls/certs {file name}.pem`.

Configure TCP Wrappers to Allow or Deny Services

To configure TCP wrappers to allow or deny services:

1. Log in to the CLI as **root**.
2. Allow services to a particular IP or domain address.
 - a. Enter `vi /etc/hosts.allow`.
 - b. Enter `{service name}: {ip/domain address}`. For example, `sendmail: 192.168.1.1` means that the TCP wrapper should allow the sendmail service request from the specified IP address.
 - c. Save and close the file.
3. If desired, deny services to a particular IP or domain address.
 - a. Enter `vi /etc/hosts.deny`.
 - b. Enter `{service name}: {ip/domain address}`. For example, `sendmail: 192.168.1` means that the TCP wrapper should deny the sendmail service request from the specified IP network.
 - c. Save and close the file.

TOPIC B Secure User Accounts

You are now familiar with encryption and basic system security. The next step is to provide users with a secure computing environment. In this topic, you will secure user accounts.

Given the reputation that Linux has as a secure operating system, there may be an inherent tendency to take a casual approach to user security. However, as a Linux administrator, it is important to institute organizational policies that help establish best practices in your Linux user community. By doing so, you will limit the potential for disasters, especially as your company user base grows.

Environment Config Files

In Linux, several environment files can be customized.

File	Description
------	-------------

File	Description
/etc/hosts.allow	Allows access to certain services and hosts.
/etc/hosts.deny	Denies access to certain services and hosts.
/etc/limits	Limits users' resources when a system has shadow passwords installed.
/etc/login.defs	Sets user login features on systems with shadow passwords.
/etc/passwd	Displays the user name, real name, home directory, encrypted password, and other information of a user.
/etc/securetty	Identifies secure terminals from which the root user is allowed to log in.
/var/log/secure	Tracks user logins. It is recommended to check this file periodically.

The /etc/login.defs File

The **/etc/login.defs** file is used with shadow passwords to set the initial path and other parameters, including how often a user must change passwords and what is acceptable as a password.

Login Levels

In Linux, you can provide root-level or user-level access to resources. By default, the root user has login privileges to all information on the system, but other users have limited login privileges.

Login Level	Description
Root login	<p>Root user is considered a specific user account, with a UID of 0. It has privileges that no other user on the system has.</p> <p>Specifically, the root user can navigate anywhere on the system, change any file, and manipulate system controls, including user accounts, storage devices, and kernel parameters.</p> <p>The system administrator(s) will generally have his or her own user accounts with normal user privileges, and the root account. As a rule of thumb, you should do as much as possible under your UID before working from the root user account.</p>
User login	<p>User accounts must be created with security in mind. The user name, or login ID, and the password are stored in two different files, /etc/passwd and /etc/shadow, and are set up with restricted access rights for added security.</p> <p>The user account file, /etc/passwd, is set up to be read-only by everybody except the root user.</p>

Command Aliases for the Root User

If you are using the Bash shell, you can create an alias for root as a precautionary measure for selected commands. An alias, in this case, is an entry in the **.bashrc** file where you can define additional actions for specific default commands. For example, the **rm** command, which is used for removing or deleting files, can be given an alias that prompts you for additional confirmation (**rm -i**).

The su Command

The **substitute or switch user (su)** command is used to change the ownership of a login session without logging out. It is generally used to switch ownership between an ordinary user and a root user, to change access permissions for administrative work.

Figure 14-11: Switching users using the su command.

The Login Shell vs. the Non-Login Shell

A login shell is a shell that is created during a user login. On the other hand, a non-login shell is a shell that you can invoke from within a login shell. For example, running the su command from a login shell, invokes the non-login shell. However, the su - command can be used to run commands in the login shell. GNOME terminals and executed scripts are non-login shells. The logout command can be run only in login shells, whereas the exit command can be run in both the login and non-login shells.

The id Command

You can use the id command to view user identities. This allows you to identify the owner of the current login session.

The sudo Command

The **super user do (sudo)** command allows users to run programs with the security privileges of the root user. It prompts you for your password and confirms your request to execute a command by checking the **/etc/sudoers** file, which is configured by the system administrator. The sudo command allows system administrators to give certain users or groups access to some or all commands without users knowing the root password. It also creates a log of all commands and arguments used, to maintain a record.

Figure 14-12: Using the sudo command to perform tasks with root privileges.

Syntax

The syntax of the sudo command is `sudo {command name} {options}`. For example, `sudo shutdown -h now` will begin a system shutdown, if a user has permission to access the shutdown command via sudo.

After executing the sudo command, the user will be prompted for his or her password. This is an extra security measure to ensure that unauthorized users cannot access the sudo command from authorized users' login sessions without their knowledge.

Password Policies

A **password policy** is a set of guiding principles that help form effective passwords. A password policy divides users of a system into various categories with differing levels of access to resources.

A password policy may have rules such as the following:

- Passwords need to be at least ten characters long.
- Passwords must be made up of numbers, upper and lowercase letters, and special characters such as punctuation.
- Passwords must not be a recognizable English word.
- Passwords must be changed every three months.

Password policies encourage users to avoid writing down their passwords. Password policies may also be enforced by training.

A Good Password Policy

Develetech Industries follows a strict password policy. Employees are instructed to choose passwords that use a minimum of twelve characters with a combination of numbers, upper and lowercase letters, and special characters such as punctuation. Employees are prompted by the system to change their passwords every 6 months. This has helped enhance the security of company data.

The Shadow Password File

The *shadow password file* is a highly protected file that is used for storing each user's encrypted password. Unlike other password files, the shadow password file is readable only by the root user.

This file is `/etc/shadow`, and can be accessed only by those processes that run at the root level.

All modern Linux distributions, including RHEL, CentOS, and Ubuntu®, have shadow passwords enabled by default.

Memory Usage

Memory usage is the sum of all the programs in the memory of an operating system. It also includes cached data. When more processes begin, the memory available for cache is reduced. If a limit is exceeded, Linux swaps out virtual memory processes that are idle most of the time.

Ways to Improve User-Level Security

There are a number of ways to improve user-level security. The following table lists some of the ways to improve user-level security.

<i>Method</i>	<i>Description</i>
Disable root login	By disabling root logins to a server, all access must be made via a non-privileged user that then executes commands via the <code>su</code> or <code>sudo</code> commands.
Disable remote login by password	Disallows all remote access to the server, except through OpenSSH key-based authentication, eliminating password security concerns.
Limit the number of users	Prevents unauthorized users from accessing the system.
Limit the number of user logins	Specifies the maximum number of sessions a user can log in simultaneously.
Limit user accounts	Specifies the date when a user account should expire.
Limit hard disk and CPU memory usage	Sets quotas for individual users to limit memory usage on storage devices and the CPU so that the system performance is not affected.
Limit processes	Limits the number of simultaneous processes that a user can run so that the system performance is not affected.

Number of Logins

You can specify the maximum number of sessions a user can log in simultaneously. For example, if you specify `username - maxlogins 4`, it means that the user will be able to log in and run four different sessions simultaneously. If `username` is replaced by `*`, it means that a maximum of four logins will be permitted simultaneously for all the users.

Limiting User Account

You can limit a user account by specifying its expiry date using the `usermod` command. For

example, `usermod -e {yyyy-mm-dd} {login name}`.

The ulimit Utility

The ulimit utility sets or gets the file-size writing limit of files written by the shell and its descendants (files of any size may be read). Only a process with appropriate privileges can increase the limit. Limits are categorized as either soft or hard. With the ulimit command, you can change your soft limits, up to the maximum set by the hard limits. You must have root user authority to change resource hard limits. The following table lists the ulimit command options.

<i>If You Need To</i>	<i>Use This ulimit Command Option</i>
List all of the current resource limits.	-a
Specify the size of core dumps, in number of 512-byte blocks.	-c
Specify the size of the data area, in number of K bytes.	-d
Specify that the hard limit for the given resource is set.	-H
Set the maximum size of files created by the shell.	-f
Specify the size of physical memory, in number of K bytes.	-m
Specify the maximum number of processes available to a single user.	-u

How to Secure User Accounts

Follow these general procedures to secure user accounts.

Execute Commands as a Superuser

To execute commands as a superuser:

1. Log in as a user other than root.
2. To run the command as the superuser, enter `sudo {command which needs root access}`.
3. Enter the password of the superuser or the user.
4. Observe that the command has been executed.

Limit User Logins

To limit the number of user logins:

1. Log in to the CLI as **root**.
2. Enter `vi /etc/security/limits.conf`.
3. To limit the user logins, type `Username - maxlogins Number of logins`.
4. Save and close the file.

Disable Root Access for Telnet and SSH Services

To disable root access for Telnet and SSH services:

1. Log in as **root**.
2. Disable root access for the **Telnet** service.
 - a. Open the `/etc/securetty` file in any editor.

- b. Remove the entries `tty0–tty9`.
 - c. Save the file.
 - d. If necessary, restart the **Telnet** service.
- 3. Disable root access for the **SSH** service.
 - a. Open the `/etc/ssh/sshd_config` file in any editor.
 - b. To disable root access, replace the line `#PermitRootLogin yes` with `PermitRootLogin no`.
 - c. Save the file.
 - d. Restart the **SSH** service.

ACTIVITY 14-1

Securing Linux Review

Scenario

Answer the following review questions.

1. Which encryption method will you use to secure data? Why?
2. What password policies do you have currently in your organization and how do you think they can be improved?

Summary

In this lesson, you examined various options available to secure a Linux system connected to a network. You configured encryption and secured user accounts. You will now be able to allow or restrict user access to network resources and safeguard your network by effectively protecting confidential data.

15 Managing Hardware

Lesson Time: 1 hour, 45 minutes

Lesson Introduction

In the last lesson, you configured various services and ensured the security of your network.

While working with Red Hat® Enterprise Linux®, hardware-related issues will arise. In this lesson, you will manage hardware.

As a Linux administrator, you will need to upgrade hardware components regularly. By adding, removing, managing, and troubleshooting hardware, you can ensure the efficient working of your system.

Lesson Objectives

In this lesson, you will manage hardware associated with Linux systems. You will:

- Identify common hardware components and resources.
- Configure removable hardware.
- Configure disk quotas.

TOPIC A Identify Common Hardware Components and Resources

There are various computer hardware and peripheral components available in the marketplace.

However, not all are compatible with Linux. In this topic, you will identify common hardware components and resources for Linux.

Before you install any operating system, you need to have all of your hardware and peripheral components in place. However, not all hardware is compatible with all systems. Having a thorough understanding of each component and how it works in a Linux system is critical for its successful installation.

Hardware Components

Basic hardware components of a Linux system include a Central Processing Unit (CPU), hard drives, memory, network adapters, and video cards. Additional components may include CD/DVD drives, sound cards, modems, USB devices, FireWire® devices, PC cards, printers, or scanners.

Hardware Resources

Hardware resources, such as Interrupt ReQuests (IRQs), Direct Memory Address (DMA), memory addresses, and *Small Computer Systems Interface (SCSI)* IDs, can cause system configuration conflicts in a Linux system. To prevent such conflicts in your system, check the documentation before configuring the settings for a device resource. And when all devices are configured, do not keep changing their settings because it can be difficult; therefore, set the devices in such a way that they are not conflicting.

Types of SCSI

SCSI, pronounced as "skuzzy," is a set of standards for connecting peripheral devices to a computer.

It is a popular format for hard disks and CD-ROM drives. It has a fast transfer rate and can handle multiple devices. There are many SCSI types, each with their own specifications.

<i>Type of SCSI</i>	<i>Bus Width and Maximum Throughput</i>
SCSI-1	8 bits at 5 MB/sec
Wide SCSI	16 bits at 10 MB/sec
Wide Ultra SCSI	16 bits at 40 MB/sec
Fast SCSI	8 bits at 10 MB/sec
Fast Wide SCSI	16 bits at 20 MB/sec
Ultra SCSI	8 bits at 20 MB/sec
Ultra2 SCSI	8 bits at 40 MB/sec
Ultra3 SCSI	16 bits at 160 MB/sec
Ultra-320 SCSI	16 bits at 320 MB/sec
Ultra-640 SCSI	16 bits at 640 MB/sec

SCSI IDs

Each SCSI device has a unique SCSI ID, also known as a SCSI address, assigned to it so that the host adapter will be able to identify the device it is communicating with. The lowest SCSI ID is 0 and the highest is 7 or 15; 0 to 7 for narrow SCSI and 0 to 15 for wide SCSI. Priority is given based on the drive ID, 7 being the highest priority.

Normally, the Host Bus Adapter (HBA), which provides the interface between the system and SCSI, is assigned the ID of 7. Slower devices should have higher-priority IDs so that faster devices do not monopolize the bus. So, the primary hard drive should be assigned a lower number and a slow device, such as a tape drive, should be assigned a higher ID number. Priority for narrow SCSI is 7, 6, 5, 4, 3, 2, 1, 0; for wide SCSI, it is 7, 6, 5, 4, 3, 2, 1, 0, 15, 14, 13, 12, 11, 10, 9, 8. SCSI IDs are set using jumpers or DIP (dual inline package) switches on the SCSI device.

IRQ Device Description

Check the IRQ address for non-plug-and-play devices to ensure that there are no conflicts. The following table shows IRQ and I/O addresses for floppy drives, printer ports, and COM ports.

<i>Device</i>	<i>IRQ</i>	<i>I/O Address</i>
fd0 (floppy disk drive 1)	6	3f0–3f7
fd1 (floppy disk drive 2)	6	3f0–3f7
fd2 (floppy disk drive 3)	10	370–377
fd3 (floppy disk drive 4)	10	370–377
lp0 (LPT 1)	7	378–37f
lp1 (LPT2)	5	278–27f
ttyS0 (COM 1)	4	3f8
ttyS1 (COM 2)	3	2f8
ttyS2 (COM 3)	4	3e8
ttyS3 (COM 4)	3	2e8

Disk Space Tracking

The *df* and *du* commands facilitate disk space tracking. The disk free (*df*) command enables you to view the free disk space, filesystem, total size, disk space used, percentage value of space, and mount point. The disk usage (*du*) command displays how a disk is used, including the size of directory trees and files within. It also enables you to track space hogs, which are directories and files that consume large amounts of space on the hard disk.

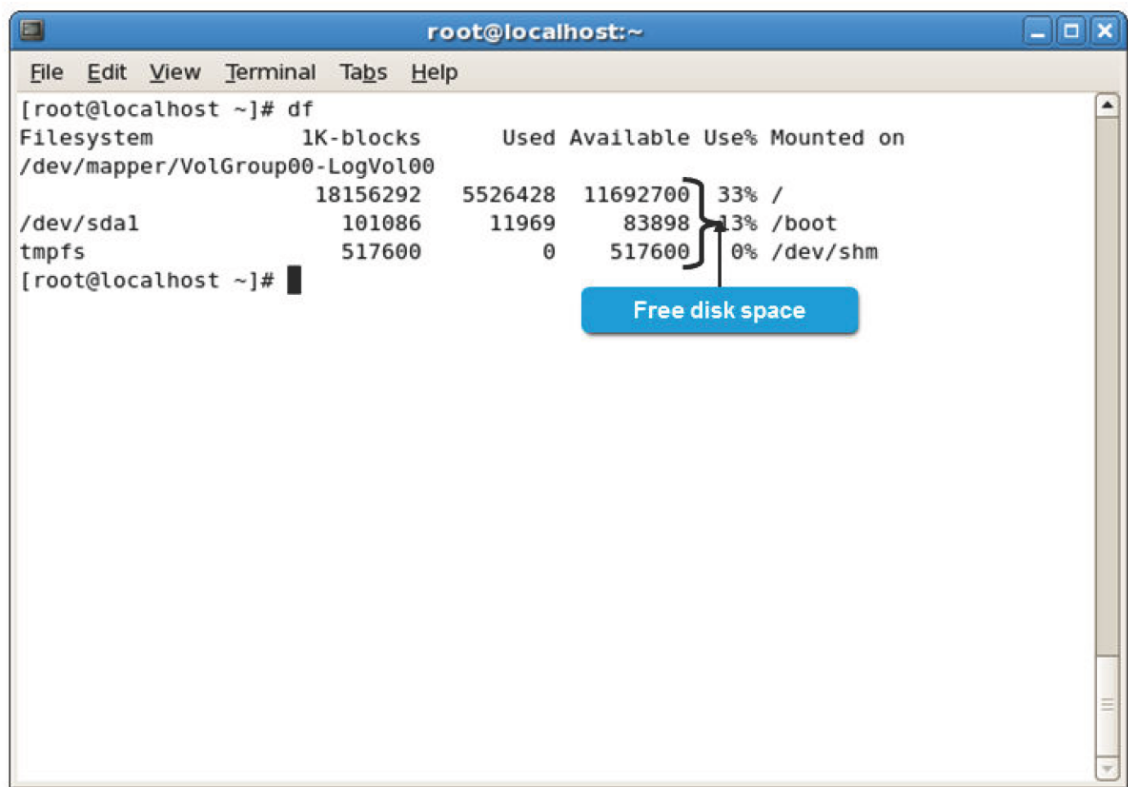


Figure 15-1: The *df* command displaying free disk space.

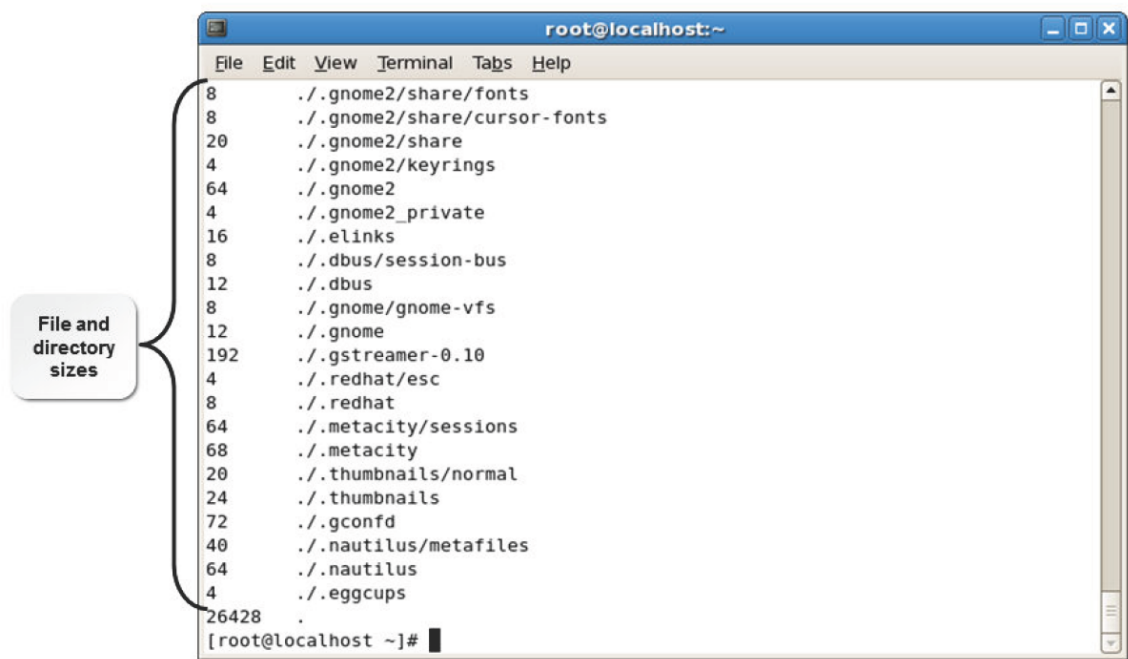


Figure 15-2: The *du* command displaying the size of directories and files on the disk.

The *du* command has various options. The following table lists some of the most commonly used options.

Option	Enables You To
-s	Display an entry for each specified file.
-h	Provide human-readable output using unit suffixes such as M (for MB) and G (for GB).

<i>Option</i>	<i>Enables You To</i>
-c	Display a grand total.

Syntax

The syntax of the du command is `du {options}`.

How to Identify Hardware Components and Resources

Follow these general procedures to identify hardware components and resources.

Identify Hardware Components and Resources

To identify hardware components and resources:

1. Log in as root.
2. Switch to the `/proc` directory.
3. Document the hardware components for future reference.
 - To display CPU information, use the `cat cpuinfo` command.
 - To display device information, use the `cat devices` command.
 - To display partition information, use the `cat partitions` command.
 - To display current hard disk size and usage, use the `df -h` command.
 - To display current IRQ resources, use the `cat interrupts` command.

TOPIC B Configure Removable Hardware

In the last topic, you examined hardware requirements for Linux systems. As a Linux administrator, you may need to add additional hardware devices or external peripherals to your system based on user requirements. In this topic, you will configure removable hardware.

Advancements in technology may require you to constantly upgrade the hardware configuration of your Linux system. Removable devices are a convenient and efficient way to upgrade your Linux system.

Removable Hardware

Linux systems allow you to utilize removable devices. Some of these devices are described in the following table.

<i>Device</i>	<i>Example</i>
Input	Keyboard, pointing device, joystick, scanner, webcam, digital tablet, and digital camera.
Network	Network Interface Card (NIC)
Output	Printer, sound card, and speakers.
Storage	CD, DVD, tape drive, flash drive, and hard drive.

USB

The *Universal Serial Bus (USB)* is a hardware interface standard designed to provide connections for numerous peripherals. *USB devices* are peripheral devices that communicate with a host computer. Some common USB devices include flash drives, memory card readers, and digital cameras.

USB Standards

USB 2.0 is the most commonly implemented standard. It can communicate at a rate of up to 480 Mb/s. The original USB 1.1 standard is still commonly found in devices and systems. It can communicate at a rate of up to 12 Mb/s. A USB 2.0 device connected to a USB 1.1 hub or port will communicate at only USB 1.1 speeds, even though it may be capable of faster speeds. Generally, the operating system will inform you of this when you connect the device.

USB 3.0, also called SuperSpeed USB, is a newer USB standard and features a maximum transfer rate of 5.0 Gbit/s. It is 10 times faster than the USB 2.0 standard, has enhanced power efficiency, and is backward compatible with USB-enabled devices currently in use.

USB 3.1, also called SuperSpeed+, is that latest USB standard and features a maximum transfer rate of 10.0 Gbit/s.

USB cables have a maximum distance before performance suffers. To work around this, one or more hubs can be used to create a chain to reach the necessary cable length. USB has a practical cable length of 3 meters, while a maximum of five hubs can be used to extend the cable length.

USB Plug and Play Capabilities

USB devices also incorporate plug-and-play technology that allows devices to self-configure as soon as a connection is made.

Kernel Support

While using USB devices, make sure that you are using a modern kernel that includes USB support.

Limited USB support was added in kernel 2.2.18, but it's best to run at least a 2.4.x kernel or newer for optimal USB support. Be diligent in keeping up to date with new versions, because they tend to change frequently.

USB in Linux

Linux treats USB drive devices like SCSI disks, registering the first device as `/dev/sda`, where "sda" means the first partition of the first device.

The Basic Architecture of the Layer Model of a USB Driver

The basic architecture of a USB driver consists of a host computer, an upper software layer, a host controller hardware layer, a physical bus, and one or more USB devices. The following table lists the components and their description.

<i>Component</i>	<i>Description</i>
<i>Host Computer</i>	Consists of two layers and controls data transfer to and from USB devices.
Upper Software Layer	Includes the USB device drivers.
<i>Host Controller Hardware Layer</i>	Converts data between the format used by the host computer and the physical format used by the USB. It is also known as the adapter layer.
<i>Physical Bus</i>	Consists of a set of USB cables that link the controller with the peripherals.

<i>Component</i>	<i>Description</i>
USB Devices	Peripheral devices that use the USB electrical and data format specifications to communicate with the host computer.

Host Controller Interface

Host Controller Interface (HCI) is an interface that instigates communication between an external device driver and the operating system of a PC. Enhanced Host Controller Interface (ECI), Open Host Controller Interface (OHCI), and Universal Host Controller Interface (UHCI) are different standards grouped under HCI, which are collectively referred to as xHCI modules, where x is a variant (E, O, or U) depending on the characteristics of the modules installed.

FireWire

FireWire is the IEEE 1394-standard, high-speed serial bus, much like USB, which can run up to 30 times faster than USB. FireWire's higher bandwidth makes it ideal for devices such as digital video cameras and high-speed hard disk drives.

FireWire Cables

In FireWire cables, electrical contacts are inside the structure of an IEEE 1394 cable connector.

This helps protect the user from electric shocks. These cables are easy and safe to use. Users can blindly insert them into the systems. Terminators are not required and manual IDs do not have to be set.

The 1394 Subsystem Core

The core of the 1394 subsystem is a module that manages high- and low-level drivers, handles transactions, and triggers events. Subsystem high-level drivers, or routines, register themselves with the IEEE 1394 module by calling the function `hpsb_register_highlevel`.

The Loopback Device

The loopback device in a Linux system allows you to access a file or a set of files as a block device.

It allows you to mount filesystem images, such as ISO, CD images, and floppy disk images, on the hard disk. You can create a floppy disk or CD, complete with the filesystem and files, on your Linux filesystem without actually copying or burning it onto the media. A copy of the floppy or CD can be taken later. Loopback devices are also used for filesystem encryption.

How to Configure Removable Hardware

Follow these general procedures to configure removable hardware.

Configure USB Hardware

To configure USB hardware:

1. Verify that the kernel version of your system is 2.4 or above.
2. Verify that the `/sys/bus/usb/devices` directory has been created.



Note: The older `/proc/bus/usb` directory is deprecated, and now the `/sys/bus/usb/` devices directory contains subdirectories with information about the USB devices connected to the system.

3. Ensure that the device is supported by Linux, or install required kernel modules for the device.
4. Plug in the device.

Review the USB Support

To review the USB support:

- If necessary, verify that the USB support has been activated.
 1. Log in as root.
 2. Connect the USB devices.
 3. To view the connected USB devices, enter `lsusb`.

TOPIC C Configure Disk Quotas

In the last topic, you identified and configured removable hardware. One such common removable hardware device is an external storage device, which can be used to access stored files, back up built-in storage devices, or store regularly used files. External storage devices are helpful when the hard disk space is not sufficient to store what a user requires. One way to manage this is to ensure that all users have access to sufficient disk space for storing individual files that they need to access regularly, but not so much space for files that they will never use again. In this topic, you will configure disk quotas.

One of the tasks that a system administrator must undertake is limiting disk space usage. Users may need to store files and data in a common location. By configuring disk quotas, you can ensure that all users have adequate storage space in that common location.

Disk Quotas

A *disk quota* is the disk space that is allotted to a user for file storage on a computer. Disk quotas need to be configured for each user. Every filesystem for which a disk quota has been implemented will have a default grace period of seven days. This means that when a user has reached the soft limit, the grace limit feature gets activated. The soft limit is the quota value beyond which disk space usage is allowed only during the grace period. Once the grace period expires, the soft limit will be enforced as the hard limit, a maximum number will be set on disk usage and users cannot exceed this limit.

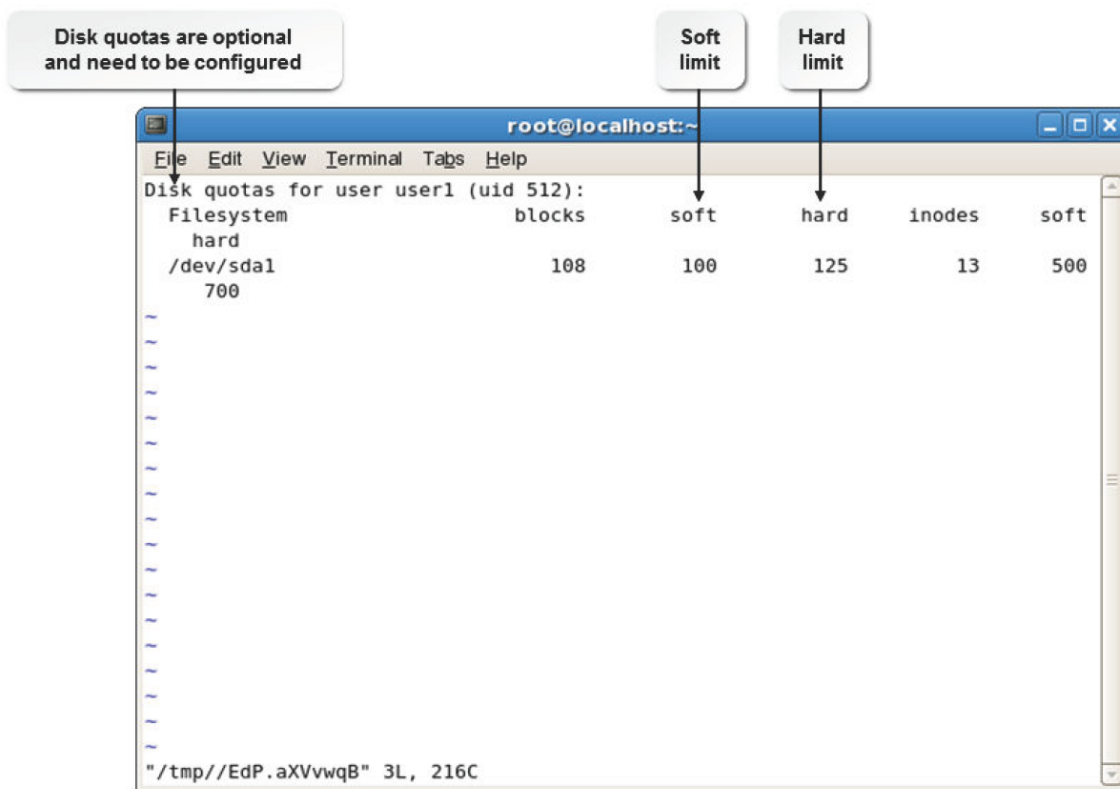


Figure 15-3: The disk quota for a user.

Quota Management Commands

Quota management is the effective allotment and monitoring of quotas for all users. Linux has various commands that help ease the job of quota management for the system administrator.



Note: Quotas should be assigned in such a way that users are able to maximize the utilization of disk resources without data overflow.

Command	Used To
edquota -u {user name}	Edit quotas for a specific user.
edquota -g {group name}	Edit quotas for a specific group.
setquota -u {user name}	Set quotas for a specific user.
setquota -g {group name}	Set quotas for a specific group.

Quota Reports

Quota reports are created by the system to view the usage of disk space by each user. These reports enable the system administrator to check which user is taking up maximum disk space. There are two types of quota reports: user quota reports and group quota reports.

A quota report contains the following details:

- The name of the user.
- The total number of blocks (in kilobytes) that have been utilized by the user on a partition.
- The *soft limit*, which is the maximum amount of disk usage that a quota user has on a partition.
- The *hard limit*, which is the absolute limit on disk usage that a quota user has on a partition.
- The *grace period*, which is the time limit before the soft limit is enforced for a filesystem with quota

enabled.

- The total number of inodes that have been used on a partition by a user.
- The soft limits on inodes.
- The hard limits on inodes.

The quotacheck Command

The quotacheck command examines filesystems for which you enabled quotas, builds a table of current disk usage, and updates the aquota.user file. The quotacheck command has various options.

<i>Option</i>	<i>Used To</i>
-g	Check group quotas.
-u	Check user quotas.
-a	Check all mounted non-NFS filesystems specified in the /etc/mtab file.
-b	Back up the quota file before writing new data.
-c	Perform a new scan and save it to the disk.

Quota Report Generation Commands

Several commands are available for the generation of effective quota reports.

<i>Command</i>	<i>Enables You To</i>
repquota -a	Display the reports for all filesystems indicated as read- write with quotas in the /etc/mtab file.
repquota -u {user name}	Display the quota report for a particular user.
quota -uv {user name}	Display the quota report for a particular user with verbose output.
warnquota -u	Check if users are not exceeding the allotted quota limit.
warnquota -g	Check if groups are not exceeding the allotted quota limit.

How to Configure Disk Quotas

Follow these general procedures to configure disk quotas.

Set Disk Quota for a Filesystem

To set disk quota for a filesystem:

1. Log in to the CLI as root.
2. Specify user quota for the partition you want to allocate to users.
 - a. Enter `vi /etc/fstab`.
 - b. In the fourth field of the partition entry, change the default values.
 - To define user quota for the specified partition, change defaults to defaults,usrquota.
 - To define group quota for the specified partition, change defaults to defaults,grpquota.
 - To define user and group quota for the specified partition, change defaults to defaults,usrquota,grpquota.

- c. Save and close the file.
3. To remount the partition, enter `mount -o remount {mount point}`.
4. To scan for the disk usage and create a quota file, enter `quotacheck -c {mount point of the partition}`.

Manage Quota Service on a System

To manage quota service on a system:

1. Log in to the CLI as root.
2. Manage the quota service on the system.
 - To turn on the quota, enter `quotaon [options] {mount point}`.
 - To turn off the quota, enter `quotaoff [options] {mount point}`.

Set Quota for Users

To set quotas for users:

1. Log in to the CLI as root.
2. Set quotas for users.
 - To set user quota, use the `edquota` command.
 - a. Enter `edquota [options] {user or group name}`.
 - b. Specify the soft and hard limits for blocks and inodes.
 - c. Save and close the file.
 - To set quotas for users using the `setquota` command, enter `setquota [options] {user or group name} {soft block limit} {hard block limit} {soft inode} {hard inode} [options] |dev|{device name} {partition number}`.

View Quota Reports

To view the quota report:

1. Log in to the CLI as root.
2. View the quota report.
 - To display the quota report for the user or group, enter `quota [options] [user or group name]`.
 - To display the quota report for the specified mount point, enter `repquota[options] {mount point}`.

1. Which of the hardware management tools do you expect you may use in your organization?
2. Do you expect that you will implement disk quotas in your environment? Why or why not?

Summary

In this lesson, you managed hardware associated with Linux systems. You will now be able to safely and efficiently replace, upgrade, manage, and troubleshoot any piece of hardware in a Linux system.

16 Troubleshooting Linux Systems

Lesson Time: 1 hour, 15 minutes

Lesson Introduction

While working with a Red Hat® Enterprise Linux® operating system, users may experience unexpected technical issues. To provide uninterrupted services to the users, you need to be able to solve the problems that arise while functioning. In this lesson, you will troubleshoot Linux-related issues.

As an administrator managing multiple systems on a network, you would have installed various services and packages required by users. However, when several users start using the systems, there may be instances when the applications and services do not function as desired. As the administrator, you will be expected to determine and resolve the problems.

Lesson Objectives

In this lesson, you will troubleshoot Linux system issues. You will:

- Use the Linux rescue environment for troubleshooting the Linux system issues.
- Troubleshoot hardware issues.
- Troubleshoot network connection and security issues.

TOPIC A Troubleshoot System-Based Issues

Previously, you managed hardware devices that help make up an entire Linux system. While Linux is inherently a stable system, it does need troubleshooting and servicing from time to time. While working with Linux, you may experience issues that may prevent you from using the system or its services. In this topic, you will troubleshoot system-based issues to help recover the Linux system.

As an administrator managing multiple systems on a network, you will eventually experience a wide variety of issues with the Linux operating system. Without proper identification and analysis, finding a solution will not only be time-consuming, but also cumbersome. Therefore, you must familiarize yourself with the procedures required to identify these issues and solve them efficiently.

Troubleshooting Strategies

Troubleshooting is the recognition, diagnosis, and resolution of problems. Troubleshooting begins with the identification of a problem, and it does not end until services have been restored and the problem no longer adversely affects users. Troubleshooting can take many forms, but all approaches have the same goal: to solve a problem efficiently with a minimal interruption of service. A troubleshooting strategy is a plan of action for identifying the causes and resolving the effects of a system-related issue. Various guidelines have to be considered while troubleshooting.

Guideline	Description
Analyze the problem.	Before attempting to troubleshoot an issue, try to identify the problem through its symptoms, such as error messages, and other available information such as log files and configuration files. Also, check if the relevant services are working properly.
Back up data.	Before experimenting with issues in configuration files, log files, or any other important data, it is recommended to make a backup to avoid loss of information and further complication of the issues.

Guideline	Description
Eliminate possible causes.	Observe whether the issue is related with the hardware, an application, a process, or any other service. Try to choose one or more symptoms and drill down to the root cause. Eliminating the root cause will rectify all the related issues.
Adopt fundamental problem- solving approaches.	After identifying the underlying causes, try out the fundamental methods of resolving the issue before proceeding to complicated problem-solving procedures.

A Basic Troubleshooting Model

A **troubleshooting model** is a standardized step-by-step approach to the troubleshooting process.

The model serves as a framework for correcting a problem on a network without introducing further problems or making unnecessary modifications to the network. Models can vary in the sequence, number, and name of the steps involved, but all models have the same goal: to move in a methodical and repeatable manner through the troubleshooting process.

Some companies developed troubleshooting processes that are systematic and logical. Following these guidelines will help you find and correct problems on your network quickly and efficiently.

One troubleshooting model divides the troubleshooting process into the following steps.

1. Identify the problem. This stage includes:
 - Gathering information.
 - Duplicating the problem, if possible.
 - Questioning users to gain experiential information.
 - Identifying the symptoms.
 - Determining if anything has changed.
 - Approaching multiple problems individually.
2. Establish a theory of probable cause. This stage includes:
 - Questioning the obvious.
 - Considering multiple approaches, such as examining the OSI (Open System Interconnect) model from top to bottom and bottom to top and dividing and conquering.
3. Test the theory to determine the cause.
 - a. When the theory is confirmed, determine the next steps to resolve the problem.
 - b. If the theory is not confirmed, establish a new theory or escalate the issue.
4. Establish a plan of action to resolve the problem, while identifying the potential effects of your plan.
5. Implement the solution, or escalate the issue.
6. Verify full system functionality and, if applicable, implement preventative measures.
7. Document your findings, actions, and the outcomes.

Troubleshooting can be a difficult process. It is not likely that anyone can develop a complete and accurate approach to troubleshooting, because troubleshooting is often done through intuitive guesses based on experience.

The Linux Rescue Environment

The [Linux rescue environment](#) is a stand-alone Linux program for troubleshooting a corrupt Linux installation. It serves as an external environment through which errors in the Linux system can be fixed without the help of the existing installation files. The rescue environment mounts the standard Linux system directories in the `/mnt/sysimage` directory. These directories are mounted either in read-write mode or read-only mode, depending on the kinds of issues.

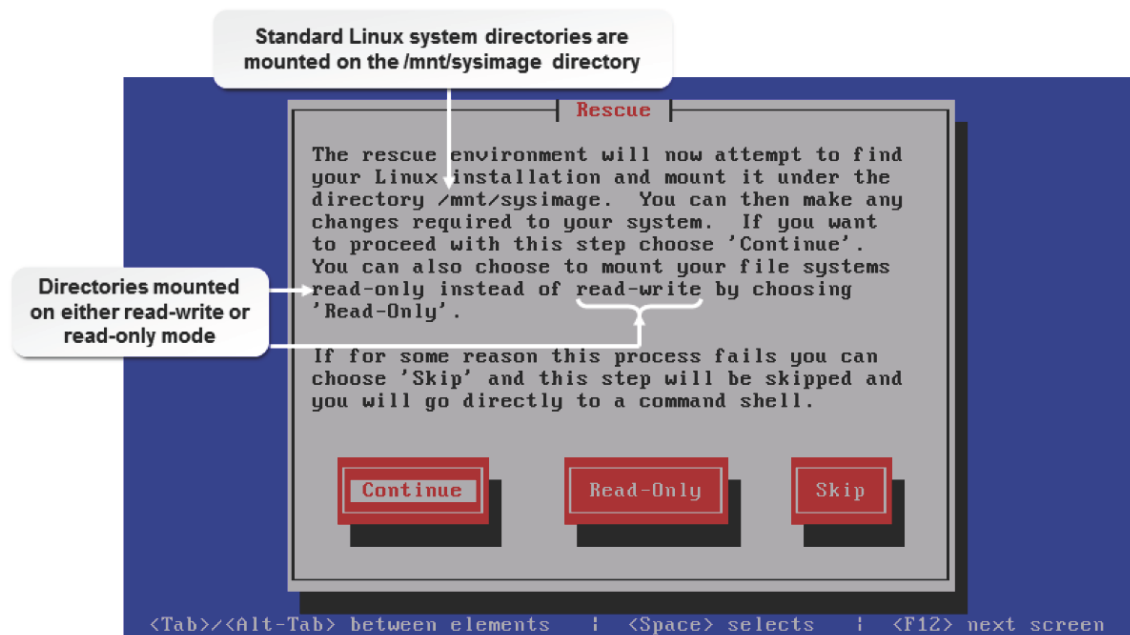



Figure 16-1: The rescue environment for troubleshooting Linux issues.



Note: In some cases, when system directories cannot be mounted on the `/mnt/sysimage` directory, the prompt will be available for troubleshooting.

chroot Mode

The chroot mode shifts the root (`/`) directory to a different location for recovery. It is also known as jail mode because it can be used in production scenarios to ensure a user will not be able to access any other file or directory except this directory and its subdirectories.

Troubleshooting the Boot Process

The following table can help you troubleshoot the boot process.

Cause	Solution
If the boot loader screen does not appear, then GRUB (GRand Unified Bootloader) may not be properly configured.	Reconfigure the <code>/boot/grub2/grub.cfg</code> file and/or reinstall GRUB in rescue mode.
If the <code>grub></code> prompt appears, then GRUB may be corrupted.	Install GRUB again in rescue mode.
If the kernel does not load, then the kernel image may be corrupted.	Install a new kernel in rescue mode.
If the kernel does not load, then the parameter passed during the system startup may be wrong.	Specify the correct parameter by editing GRUB on the boot loader screen.

Cause	Solution
<p>If there is a kernel panic, then:</p> <ol style="list-style-type: none"> 1. The boot loader may have been misconfigured. 2. The /etc/inittab file is misconfigured, or Systemd configuration is incorrect or incomplete. 3. The root filesystem is misconfigured. 	<p>Use the applicable solution:</p> <ol style="list-style-type: none"> 1. In rescue mode, configure the boot loader configuration. 2. In rescue mode, define parameters in the /etc/inittab or Systemd config files correctly. 3. In rescue mode, run a filesystem check using the fsck command on the filesystem.
If the kernel loads, but /etc/rc.d (or systemd settings) causes an issue, then the /etc/fstab file may have an error.	In rescue mode, fix the /etc/fstab file.
If the kernel loads, but /etc/rc.d (or systemd settings) causes an issue, then the fsck utility may have failed.	In rescue mode, run the fsck command manually.
If the services do not start correctly, then they may not have been configured properly.	Configure the services properly.

Rescue Environment Utilities

A set of utilities is available in the rescue environment to troubleshoot different issues.

Category	Utility
Disk maintenance utilities	<ul style="list-style-type: none"> • LVM (Logical Volume Manager) utilities such as lvcreate, lvresize, and lvremove. • Software RAID (Redundant Array of Independent Disks) utility such as mdadm. • Disk partitioning and swap utilities such as fdisk, sfdisk, gdisk, mount, umount, and mkswap. • Filesystem utilities such as mkfs, tune2fs, fsck, e2fsck, and XFS utilities.
Networking utilities	<ul style="list-style-type: none"> • Network debugging utilities such as ip, ifconfig, route, dig, netstat, traceroute, host, and hostname. • Network connectivity utilities such as ssh, ftp, and scp.
Other utilities	<ul style="list-style-type: none"> • Shell commands such as chroot and bash. • Process management tools such as ps and kill. • Editors such as vi and nano. • File management commands such as cd, ls, cp, rm, and mv. • Kernel management utility such as sysctl. • Package management tools such as rpm and yum. • Archiving and compression utilities such as tar and gzip.

Environment Configuration Problems

Configuration problems could prevent a user from logging in to a system and accessing the services provided by the server. Other problems could also be caused due to system variables or due to user and group accounts. The symptoms, causes, and solutions for common configuration problems are provided in the following table.

<i>Symptom</i>	<i>Cause and Solution</i>
The user is unable to create a user or a group account.	Cause: The user does not have admin privileges, or the system is unable to allocate memory to the user account due to insufficient memory. Solution: Check whether the required privileges are granted to the user. Also, check for free space in the memory that can be allocated to the user account.
The user is unable to log in.	Cause: The settings in the user account or the group account could be wrong. Solution: Check the user or group account settings, including the permission to log in to the system, the shell, and the path of the home directory.
The user is unable to access files and directories.	Cause: The required permission is not granted to the user. Solution: Check the user or group quota and the privileges granted to the user.
The user is unable to execute basic commands or applications.	Cause: The environmental variable is not properly set. Solution: Check the environmental variables and the library files of the application.
The scheduled jobs are not executed.	Cause: The crond daemon has not started or stopped due to the invalid configuration. Solution: Check whether the crond daemon is running. Otherwise, check whether the configuration set in the crontab file is correct.
The user is unable to switch between the runlevels.	Cause: The PATH variable is not set properly or permission is not granted to the user to switch between runlevels. Solution: Check whether the user is granted the necessary privileges required to change the runlevel or if the path of the sbin directory is set in the PATH variable.

Core System Variables

Core system variables affect the behavior of applications and commands. Some of the system variables and their functions are given in the following table.

<i>Use This Variable</i>	<i>If You Need To Specify</i>
HOSTNAME={hostname}	The hostname of the system.
SHELL={shell path}	The shell path for the system.
MAIL={mail path}	The path where mail will be stored.
HOME={home directory}	The home directory of the user.
PATH={user path}	The path in which the user needs to operate.
HISTSIZE={number}	The number of entries to be stored in the history.
USER={user name}	The name of the user.

Single-User Mode

Single-user mode in Linux can be initialized by changing the runlevel to 1. It is used when the system does not allow you to log in after booting. The networking feature is disabled in single-user mode, which makes it an ideal mode to troubleshoot network problems. Single-user mode can be used for filesystem checks, because most of the partitions are not mounted in runlevel 1. This mode can even be used to recover the root password.

```
sh-3.2# passwd
Changing password for user root.
New UNIX password:
Retype new UNIX password:
type=1188 audit(1248939967.762:16): user pid=3699 uid=0 auid=4294967295 msg='PAM
: chauthtok acct="root" : exe="/usr/bin/passwd" (hostname=?, addr=?, terminal=co
nsole res=success)'
type=1188 audit(1248939967.763:17): user pid=3699 uid=0 auid=4294967295 msg='op=
change password id=0 exe="/usr/bin/passwd" (hostname=?, addr=?, terminal=console
res=success)'
passwd: all authentication tokens updated successfully.
sh-3.2# _
```

Figure 16-2: Changing the root user password in single-user mode.

Boot Disks

A **boot disk** contains operating system files, such as *init*, *klogd*, and *syslogd*, required to start a system.

It can be a hard disk, floppy disk, CD-ROM, DVD-ROM, or USB (Universal Serial Bus) drive. The boot disk contains configuration files, startup files, and programs. The boot disk is used to boot a system following a hard disk crash. Some distributions use the first CD in the installation set as the boot disk. Other distributions allow you to create a floppy disk that can be used to boot the system.

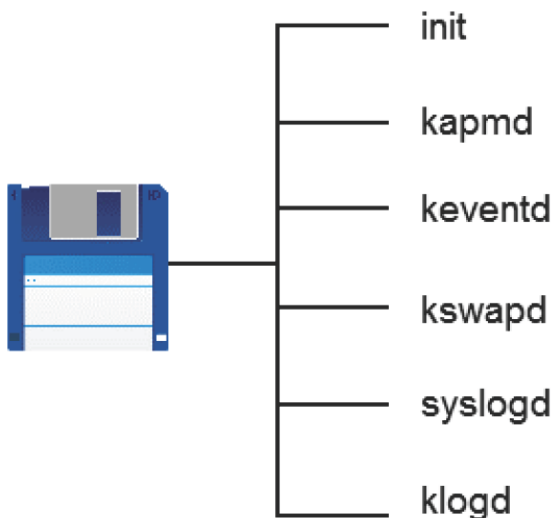


Figure 16-3: Contents of a boot disk.

Ramdisks

A **ramdisk** is a portion of memory that is allocated and used as a partition. The memory allocated as ramdisk is treated as a hard drive. Frequently accessed files can be placed in the ramdisk, which in turn will increase the performance of the system.

The ramdisk word Keyword

The ramdisk word keyword is a keyword that specifies the location of the root filesystem. The ramdisk word can be set and accessed using the *rdev* command.

The boot.iso File

The **boot.iso** file is an ISO-9660 image that is used to create bootable CD- or DVD-ROMs. This image file can be burned on to a CD-ROM, which can then be used for installing Linux, just like the original installation media itself. The boot speed of CD-ROMs is also an added advantage.

The diskboot.img File

The **diskboot.img** file is a VFAT (Virtual File Allocation Table) filesystem image that is used to create bootable USB pen drives. Once the image is written onto the USB, it can then be used as a media for Linux installation. However, using a USB to boot a system depends on the BIOS (Basic Input/Output System) settings. The *diskboot.img* image file should be written onto the USB using the *dd* command.

Root Disks

A **root disk** contains directories, such as *etc*, *bin*, *home*, and so on, which contain files required to run a Linux system. It need not contain a kernel or a boot loader. The root disk can run a system without depending on any

other disk.

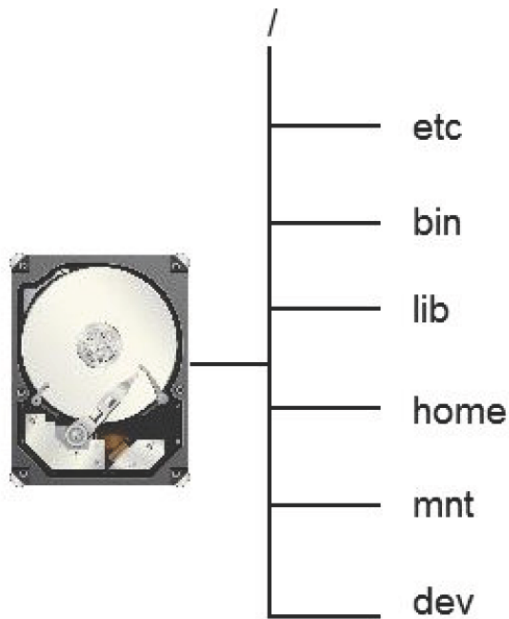


Figure 16-4: Components of the root disk.

Zero-Filled Files

There are times when you might need to create a filesystem that does not contain any data or partition table. One of these times might be when you need to build a compressed root filesystem.

To do this, you need to create a zero-filled file, partition, or ramdisk. The `dd` command can be used to create a zero-filled file or partition, which overwrites the disk with zeroes, effectively wiping out all data. This command erases data and partition tables, if any. By creating the zero-filled file or partition, you will be able to compress a filesystem to the maximum.

Kernel Panic

If a user is unable to boot a system, it may be due to disk errors caused by hardware devices. When the "Kernel Panic" message is displayed, the filesystem is corrupted or inaccessible. To resolve this issue, log in to rescue mode and perform an integrity check on the filesystem.

How to Troubleshoot Linux-Based Issues

Follow these general procedures to troubleshoot Linux-based issues.

Analyze a Problem by Gathering Data

To analyze a problem by gathering data:

1. Log in to the CLI as **root**.
2. Gather data about the issue using appropriate commands and files.
 - To analyze the history of commands run by the user, enter `history`.
 - To find the specified keyword in the log file while troubleshooting, enter `grep {keyword} {log file name}`.
 - To view if there are any changes in the file compared with the last backed up state, enter `diff {current file} {backed up file}`.
 - To find all the files that were modified within a specified timing, enter `find {location of the directory to`

search} -cmin -{time in minutes}.

- To collect more information when the specified command is run, enter *strace {command}*.
- To list all open files of all active processes in a system, enter *lsof*.
- To view the log file as and when it is updated, enter *tail -f {log file name}*.
- Configure system logs to log all debug messages.
 - a. To open the system log configuration file, enter *vi /etc/rsyslog.conf*.
 - b. To set the type and level of severity to be logged in the specified file, type *{facility} {level of severity} {file where the log messages will get stored}*.
 - c. Save and exit.
 - d. To restart the system log service and apply the changes, enter *service rsyslog restart*.

Use the Linux Rescue Environment

To use the Linux rescue environment:

1. To boot from the recovery disc, ensure that your system is set to boot from your DVD drive, modifying the BIOS boot order settings if necessary.
2. Insert the CentOS Installation DVD into the DVD drive and boot the system.
3. To view the **Troubleshooting** menu, at the boot menu, press the **Down Arrow** once to select Troubleshooting and press **Enter**.
4. To enter rescue mode, on the **Troubleshooting** menu, press the down arrow once to select Rescue a CentOS system and press **Enter**.
5. To enter rescue mode, on the **Rescue** menu, press the **Tab** key once to select Continue and press **Enter**.
6. A message is displayed, stating that the root partition will be mounted in the **/mnt/sysimage** directory. To mount the filesystem with read and write permissions, select **OK**.
7. A message is displayed, stating that your system has been mounted under the **/mnt/sysimage** directory. To continue with the boot process, select **OK**.
8. The root directory is now mounted on the ramdisk and a shell prompt is displayed. To change the root directory to the one mounted in the **/mnt/sysimage** directory, enter *chroot /mnt/sysimage*.
9. Troubleshoot to find the cause of system failure and make the necessary changes to recover the system.
10. To exit the chroot environment, enter *exit*.
11. Enter *sync* so that the changes you made are reflected in the filesystem on the hard disk.
12. To exit from rescue mode, enter *exit*. The system will now reboot.

TOPIC B Troubleshoot Hardware Issues

In the last topic, you corrected system-based issues in a Linux system. In addition to the system itself, hardware devices may get corrupted and may not work properly. In this topic, you will troubleshoot hardware issues.

Systems may be connected to external devices such as speakers or modems. Sometimes, these devices may not work properly. Finding the cause of the problem and fixing it will help you solve hardware issues and keep systems functioning smoothly.

Troubleshooting Tools

There are many troubleshooting tools that you can use, depending on the type of problem you are facing and the environment in which you are working. Some of these tools are described in the following table.

<i>Tool</i>	<i>Description</i>
dmesg	A system administration command that is used to examine and control the kernel initialization process. It is used to print messages about the status of various hardware devices on the system during kernel initialization. Status messages can also be accessed from the <i>/var/log/ dmesg</i> file.
/dev	A file that is used to create a boot or recoverable disk.
GNU Parted	A program that allows you to create, destroy, resize, move, and copy hard disk partitions.
HardDrake	A service that provides hardware detection in a graphical interface.
KNOPPIX	A bootable CD (or DVD) that contains GNU/Linux software, which includes automatic hardware detection and support. KNOPPIX can be used as a rescue system.
ifconfig	A command that is used to view the IP address and subnet mask and verify that they are allocated. It can also be used to debug or tune a system.
/proc and /sys	The <i>/proc</i> and <i>/sys</i> filesystems are pseudo-filefilesystems that are used as an interface to the kernel data structures. Each process contains a subdirectory in the <i>/proc</i> directory.

Starting and Stopping Processes to Locate and Correct Problems

Both services and processes can be stopped and restarted. This can sometimes be used to fix problems. You can use the `ps` command along with the `grep` command to locate processes that you need to check on. You can then kill the processes if necessary.

The `pgrep` command is used to look up or signal processes based on their names or other attributes. It looks through the running processes and lists PIDs that match the criteria you specify.

For instance, the `pgrep -u root sshd` command lists only processes called `sshd` and that are owned by the root user. The command `pgrep -u root,daemon` lists all processes owned by root or daemon.

The `kill` command can be used in conjunction with the `pgrep` command to stop processes.

Starting and stopping processes is just one more way to troubleshoot problems. When you see a certain symptom, such as a process taking too long, you should first check on the process using the `ps` or `pgrep` command; then if necessary, end the process using the `kill` or `kill` command. You should next examine the process (the script or other command sequences associated with that process) and check for any problems. After fixing the problems, you should try running the command or script again. Check on it periodically to see if it is working properly.

Hardware Problems

Hardware devices may experience failures anytime while the system is being used.

<i>Symptom</i>	<i>Cause and Solution</i>
The user is unable to hear from the speakers.	Cause: The speaker or the sound card is not functioning properly. Solution: Check the speaker and its corresponding driver. If you still have a problem, then you need to check the sound card.
A system connected to the UPS shuts down abruptly.	Cause: The UPS is malfunctioning, or there is a mismatch between the UPS settings and the configuration file. Solution: Check the serial ports, the cable, and the configuration file.

Symptom	Cause and Solution
The user is unable to move the pointer in GUI mode.	Cause: The mouse does not function properly due to the configuration settings or there could be a problem in the device. Solution: Unplug and reconnect the mouse, then restart the system.
The user is unable to access the CD/DVD drive.	Cause: The drive is not mounted or there is some problem with the driver. Solution: Check whether the read/write indicator is on. Otherwise, check the power cable connected to the drive.

Viewing Hardware Details

Some commands that are frequently used for viewing hardware details are listed in the table.

Command	Used To
/usr/bin/dmesg	View bootup messages.
/sbin/lspci	View information about Peripheral Component Interconnect (PCI) cards.
/sbin/lsusb	View information about USB devices.
/usr/bin/lscpu	View information about the installed CPU(s).
/sbin/lsmmod	View a list of loaded modules.
/bin/uname	View system information such as the kernel name, release and version numbers, hardware platform, and operating system.

Guidelines for Troubleshooting Hardware Issues

Follow these general guidelines to help you troubleshoot hardware issues.

Troubleshoot Sound Issues

To troubleshoot sound issues:

- Verify that the speaker is connected, switched on, and is functioning.
- If the speaker is functioning but the problem persists, verify that the sound card is detected while booting.
 1. Verify that the sound card is listed in the output of the lspci command.
 2. If the sound card/device is not detected, contact your hardware engineer to resolve the sound issue.
- If the sound card/device is detected and the problem still persists, verify that the sound card module is loaded.
 1. Verify that the sound card module details are listed in the output of the lsmod command and lsmod {module name} command.
 2. If the sound card module is not loaded, add an entry for the sound card in the **/etc/modprobe.conf** file. To add an entry in the file, you need to know the slot number and the name of the module used for the sound card and specify it in the format alias sound-slot-{slot number} {module name}.



Note: The **/lib/modules/[kernel version]/kernel/sound** directory contains modules for the sound card.

3. To load the module automatically, reboot the system.

Note: You can also load the module using the modprobe or insmod command. If you want to use the modprobe command,



run the `depmod` command to build or update a module database.

Troubleshoot Issues Related to UPS Devices

To troubleshoot issues related to UPS devices:

- Verify that the UPS device is connected properly to the server.
- Verify that the serial port is configured correctly.
 1. Verify that the settings listed in the output of the `setserial -a /dev/ttyS0` command matches your device specifications.
 2. If necessary, change the serial port settings, using the commands, `setserial /dev/ttyS{port number} {spd_normal | spd_hi | spd_vhi}` and `setserial /dev/ttyS{port number} baud_base {baud rate}`.
- If the UPS device is still not working properly, then the issue is hardware related. Contact your hardware engineer to resolve the issue.

Troubleshoot Mouse Issues

To troubleshoot mouse issues:

- Verify that the mouse is connected properly to the system.
- Reboot the system.
- If the mouse is still not working, then the issue is hardware related. Contact your hardware engineer to resolve the issue.

Troubleshoot DVD Disk Problems

To troubleshoot DVD disk issues:

- Verify that the power connector to the drive is connected and working.
- If the connection is not powered on, then there is a problem with the power connector.
 1. Verify that the drive access light indicator is glowing.
 2. If it is not glowing, the power connector needs to be checked and replaced.
- If the power connector is working and the access issue persists, then there is a problem with the DVD drive or the DVD.
 1. With your hardware engineer's help, verify that the DVD drive is functioning properly.
 2. If the DVD drive is functional, verify that your DVD is functioning properly.

Troubleshoot Printing Problems

To troubleshoot printing problems:

- Verify that the printer cables are connected properly and the power source is switched on.
- Verify that the paper trays are stocked.
- To verify that the printer daemon is running, enter `systemctl status cups.service` and, if the daemon service is not running, enter `systemctl restart cups.service`.
- Check the status of the print job in the queue.
 - In the CLI or in the GUI terminal window, enter `lpq -P {print queue name}`.
- To restart the CUPS service, enter `service cups restart`.
- To verify that the print job is getting executed, enter `lpr {file name}`.

TOPIC C Troubleshoot Network Connection and Security Issues

In the previous topics, you identified and solved system- and hardware-based issues in Linux. In a networking environment, Linux systems will be prone to connection- and security-related issues.

You need to continually identify and prepare for vulnerabilities. In this topic, you will troubleshoot network connection and security issues.

Security encompasses a number of different aspects; from passwords and permissions to data encryption, firewalls, and even physical security. Despite all this protection, if you are not aware of the symptoms that lead to security breaches, or if you are not familiar with steps required for repairing corrupted files, your network will remain open and vulnerable to potential attacks.

Network Issues

If users are unable to connect to a network, they will not be able to log in to their systems or access the services or shared resources. Network problems can be categorized as hardware-related issues and service-related issues. Hardware-related network issues can be solved by checking the network devices, including the network cable and the network card. Service-related network issues can be fixed by checking the network settings of a system or the server.

Network Troubleshooting Utilities

The `traceroute`, `ping`, and `arp` utilities are very useful in troubleshooting issues related to remote network services.

Utility	Used To
traceroute	Track the route data that it takes to get to its destination. Utilizing the Time to Live (TTL) field of the IP protocol, <code>traceroute</code> attempts to obtain an Internet Control Message Protocol (ICMP) <i>Time_Exceeded</i> response from each gateway encountered on the path between the sender and the final destination. User Datagram Protocol (UDP) probe packets are sent with a short TTL. The <code>traceroute</code> utility then listens for an ICMP <i>Time_Exceeded</i> reply from a gateway. This continues until you can get an ICMP <i>Port_Unreachable</i> response, which means that you either got to the host or reached the default maximum number of hops (30). The address of each system that responds (each gateway you pass through) is printed to your screen; if no response is received within five seconds, an asterisk (<code>*</code>) is printed for that probe.
ping	Verify that a system can be reached on a network. It checks the hostname, the IP address, and whether the remote system can be reached. <code>ping</code> uses the ICMP <i>Echo_Request</i> datagram to check connections among hosts, by sending echo packets and then listening for reply packets.

<i>Utility</i>	<i>Used To</i>
arp	Display information, such as the hardware address, the hostname, and the network interfaces, about the Address Resolution Protocol (ARP) cache.

ARP

Address Resolution Protocol (ARP) is a network protocol that is used by IP to map network addresses to MAC addresses.

Symptoms of Network Security Problems

There are a variety of ways that security can be compromised on a system. It is recommended that you check the Linux log files before troubleshooting. Some symptoms that indicate potential security problems include:

- Disruption or Denial-of-Service (DoS).
- Unauthorized system use for processing data.
- Unexplained system hardware changes.
- Theft (data information and vandalism).
- Unusual software characteristics.
- Suspected virus outbreak.

Security Tips

Avoid using authentication methods based solely on IP addresses. Keep network packages up-to-date, and be aware of the new versions of programs such as Berkeley Internet Name Domain (BIND), Postfix, and Secure Shell (SSH). Disable unnecessary network services.

System Security Monitoring Tools

Various tools can be used to effectively monitor a system for any security issue and identify symptoms.

<i>Tool</i>	<i>Description</i>
System Log Files	There are three types of system log files that can help in monitoring system security: Log: This file contains information about connections established and files transferred. Stats: This file lists file transfer statistics. Debug: This file contains debugging information and login and password information for remote system connections.
Central Network Log Server	The reports generated from the server contain useful information on server logs and online alerts, which can be analyzed for identifying security breaches or threats.
chkconfig and systemctl	These commands can be used to check configuration files and update and query runlevel information for system services.

Troubleshoot Security Issues

To troubleshoot security issues:

1. Check the **/var/log/messages** file for warnings or errors.
2. Check the **/var/log/secure** file for warnings or errors.

Network Security Vulnerabilities

Although Linux is considered a secure operating system, a network of systems can still have unauthorized users gaining access. Once an attacker gains access to a system, almost any security system can be compromised.

Vulnerabilities include:

- Proliferation of worms and viruses via email messages.
- Malicious execution of programs by a user with root privileges.
- Potential hole in the Linux kernel.
- Passwords that can be easily deciphered.
- Services running on the system such as File Transfer Protocol (FTP), Server Message Block (SMB), Sendmail, and Simple Network Management Protocol (SNMP).
- Domain spoofing.
- DNS servers running vulnerable versions of BIND.
- Remote Procedure Calls (RPCs).

IP Spoofing

IP spoofing is a technique for changing, or spoofing, your IP address in order to fool the target system into believing that your IP identity is actually another system with the spoofed address.

Software Vulnerabilities

Software vulnerabilities account for many successful attacks because attackers are opportunistic.

They exploit well-known flaws using the most effective and widely available attack tools. They also count on organizations that do not fix the problems and scan the Internet for vulnerable systems.

BIND Attack

In a BIND attack, an intruder can erase your system logs and install tools to gain administrative access. In addition, once the attacker has gained access, he or she uses the attacked system to scan for and attack other network systems running vulnerable versions of BIND. In effect, the intruder uses the compromised system to attack hundreds of remote systems, resulting in additional successful compromises.

Sendmail Flaws

Over the years, flaws have been found in Sendmail. In one of the most common intrusions, the attacker sends a crafted mail message to a machine running Sendmail. Sendmail, in turn, interprets the message as instructions requiring it to send the password file to the attacker's machine.

SNMP Flaws

SNMP uses an unencrypted community string as an authentication mechanism, and the default community string used by many SNMP devices is public. Sniffed SNMP traffic can reveal information about the structure of your network, as well as the systems and devices attached to it.

Honeypot Systems

A system designed to attract attackers is known as a **honeypot**. If an attacker manages to get past your packet filter and starts scanning for options, the honeypot should be the system configured to look like it is vulnerable to known attacks. A honeypot system should not be too easy to spot because a savvy intruder will be tempted to look further on the network.

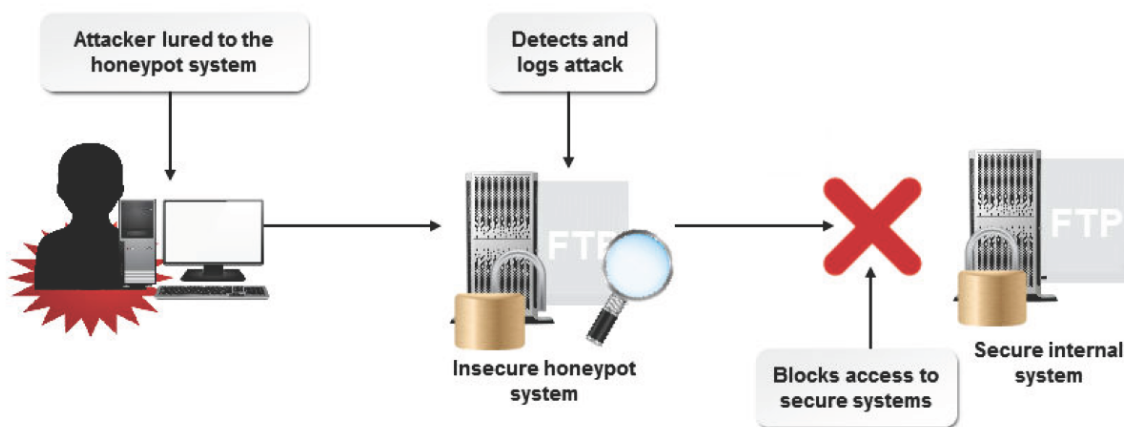


Figure 16-5: The functioning of a honeypot.

Goals of a Honeypot

There are several goals for a honeypot:

- To provide a lure so that attackers stay away from other equipment. You want the attackers to see a vulnerable system that they know they can exploit and gain access to. This system needs to be as such that the attacker focuses his or her energy on exploiting the system, as opposed to the server with important data that is sitting right next to it.
- To know that the honeypot system will be attacked, so that you can take extra measures when logging in to it. These logs should be moved off the system frequently—perhaps hourly or daily if your network is a high-profile target.
- To increase the ability to detect and respond to incidents. The theory is that if you are aware of what the attacker is doing to your honeypot, you can be better prepared to defend or, if possible, prevent the attack on your production systems.

Legal Issues Regarding Honeypots

Be aware that there may be legal issues surrounding the use of the honeypot technology. The intentional setup of a honeypot may be considered entrapment, and therefore the same rules apply as in the real world.

Another issue is that of privacy. If an attacker were to set up an IRC server on the honeypot, it will be possible to log all conversations on that server. There is currently no defined law explicitly regarding this subject. However, it should be noted that an attorney could make privacy a viable defense argument.

Guidelines for Troubleshooting Network Issues

When troubleshooting a network problem, well-established guidelines help you to narrow down the cause of the problem and map steps toward its resolution.

General Troubleshooting Process

Consider using the following process to troubleshoot a network problem:

1. Define the problem and gather the facts.

2. If possible, re-create the problem.
3. Consider all possibilities.
4. Create and implement an action plan.
5. Observe and document results.
6. Provide feedback.

Troubleshoot Network Issues

To troubleshoot network issues:

- Verify that the network cable is plugged in properly.
- To find out more information about the error, view the **/var/log/messages** file.
- Verify that the network service is started.
 1. To view the status of the network service, enter `service network status`.
 2. If the service is stopped, to start it, enter `service network start`.
- Verify that the IP address and subnet mask are allocated by viewing the output of the `ifconfig` or `ip` commands.
 1. To view the IP address and subnet mask, enter `ifconfig` or `ip addr`.
 2. If no entries for the IP address and subnet mask are displayed, determine if the IP addresses are allocated manually or through a DHCP server.
 - a. If IP addresses are allocated through a DHCP server, change the **BOOTPROTO** parameter to **dhcp** in the **/etc/sysconfig/network-scripts** file.
 - b. If IP addresses are allocated manually, verify that the **IPADDR** and **NETMASK** parameters are set in the **/etc/sysconfig/network-scripts** file.
 - c. Restart the network service.
 3. To verify that you are able to connect to the network, ping the network gateway using the command, `/bin/ping {IP address}`.
- Verify that the default gateway and routing table are configured properly.
- Verify that the name-to-IP address resolution on your network is working properly.
 - If you implemented DNS on your network, verify that the DNS entries are correct.
 1. Using the `host`, `dig`, or `nslookup` commands, verify that the name-to-IP address mapping is correct in the DNS configuration files.
 2. `dig {host name or FQDN}`.
 3. `host {host name or FQDN}`.
 4. `nslookup {host name or FQDN}`.
 - If you have not implemented DNS on your network, verify that the **/etc/hosts** file has correct name-to-IP address mapping information.
- Verify that IP forwarding is enabled.

1. Verify that the **proc/sys/net/ipv4/ip_forward** file has the value **1**.
2. If the file contains 0, change the value to **1**.
 - a. In the **/usr/lib/sysctl.d/00-system.conf** file (or the **/etc/sysctl.conf** file on older systems), modify the value of the **net.ipv4.ip_forward** parameter to **1**.
 - b. Run the **sysctl** command to apply the changes in the **sysctl -p** file.
- Verify that the ports of the service you are trying to access are open at the destination host.
 1. Use Telnet to access the service through a specific port, **telnet {host name} {port number}**.
 2. In the **/etc/hosts.allow** and **/etc/hosts.deny** files and iptables, verify that you are allowed to access the ports.
 3. If the port is not open, start the service by using the **systemctl start {service name}** command or by adding an entry for the startup script in a **/etc/systemd/system/ SCRIPTNAME.service.d/*.conf** file.
- Verify that the hostname is set.
 1. Display the hostname by using the **hostnamectl status** command.
 2. If the hostname is not set, to add an entry for the host, enter **hostnamectl set-hostname your-new-hostname** file.

ACTIVITY 16-1

Troubleshooting Linux Systems Review

Scenario

Answer the following review questions.

1. How does troubleshooting in Linux differ from the troubleshooting approach you've taken with other systems?
2. Provide an example of a recent problem you encountered in your environment and how you were able to resolve it.

Summary

In this lesson, you acquainted yourself with the various troubleshooting strategies in Linux. This will enable you to effectively tackle most of the issues that may arise while working with Linux-based systems.

17 Installing Linux

Lesson Time: 2 hours, 15 minutes

Lesson Introduction

You have knowledge about all elements and services in the Red Hat® Enterprise Linux® operating system. Getting acquainted with the services and working of the Linux operating system will enable you to recognize your requirements while installing Linux. In this lesson, you will install the Linux operating system.

As a Linux professional, you have to ensure that your computer's settings and hardware configuration are sufficient for hosting Linux. Also, you need to determine the features that have to be installed to suit your requirements.

Lesson Objectives

In this lesson, you will install the Linux operating system. You will:

- Prepare for a Linux installation.
- Identify the phases of the Linux boot sequence.
- Configure the GRUB 2 boot loader.
- Install the Linux operating system.

TOPIC A Prepare for Installation

In the last lesson, you did some troubleshooting of devices to ensure the smooth operation of the entire Linux system. One easy way to optimize the performance of the Linux system is to only select peripheral and hardware devices that are fully supported by Linux. In this topic, you will perform the preparation tasks necessary for a successful installation of Linux.

Preparing for a new Linux installation is much like setting out to cook a fine meal. You need to gather all the necessary ingredients to accomplish the task. Thorough preparation will help prevent failure when attempting to boot your system for the first time.

Hardware Compatibility

The first thing you should do before purchasing any hardware component is to check whether it is on the Hardware Compatibility List (HCL) for Linux. Before you install Linux, you should gather information about your system. Much of this information is available in your system documentation.

Some of the questions that you should address before purchasing a hardware component are listed in the following table.

<i>Component</i>	<i>Questions To Address</i>
Hard drive	<ul style="list-style-type: none">• How many devices are installed?• Is it Integrated Drive Electronics (IDE) or Small Computer Systems Interface (SCSI)?• How large is the drive?• How many cylinders are contained on the drive?

Component	Questions To Address
Hard disk controller	<ul style="list-style-type: none"> • Is it IDE or SCSI? • Who is the manufacturer?
Memory	How much RAM is installed?
CD/DVD drive	Is the interface type IDE, SAS, SATA, SCSI, or other? (If it is not a supported interface type, you may need to record the make and model.)
SCSI adapter	<ul style="list-style-type: none"> • Who is the manufacturer? • What is the model?
Network card	<ul style="list-style-type: none"> • Who is the manufacturer? • What is the model?
Mouse	What type is it? (If serial, record the port to which it is connected.)
Monitor	<ul style="list-style-type: none"> • Who is the manufacturer? • What is the model? • What are the horizontal and vertical refresh rate ranges of the monitor?
Display adapter	<ul style="list-style-type: none"> • What chipset is used in the display adapter? • How much video RAM does the display adapter use?

Linux Distributor Testing

Distributors, such as Red Hat®, Inc., test whether Linux operates correctly with specific hardware.

After testing, they produce HCLs for each supported version. The lists include information about the hardware, but because the computer system configurations vary between what you may have and what they tested, the variations in manufacturing specifications aren't guaranteed to be compatible with any hardware. An HCL is a list of hardware that has been tested under specific conditions (usually under many conditions) and should be used as a guide, not as an absolute testament that the hardware will work perfectly straight out of the box (because your system is probably configured in a slightly different way than the test systems).

Hardware Compatibility Websites

The following Linux hardware compatibility sites will assist you in determining if your hardware will work with Linux:

- Red Hat hardware compatibility list: <https://hardware.redhat.com/>
- CentOS hardware compatibility list: <http://wiki.centos.org/AdditionalResources/HardwareList>
- SUSE® hardware compatibility list: <http://en.opensuse.org/Hardware>
- Fedora® hardware compatibility list: <http://fedoraproject.org/wiki/HCL>
- Generic Linux hardware compatibility list: www.linuxquestions.org/hcl/
- Ubuntu® hardware compatibility list: www.ubuntu.com/certification/desktop/

- Linux Mint hardware compatibility list: <http://community.linuxmint.com/hardware>

CPU Compatibility

It is also important to determine if your Central Processing Unit (CPU) is compatible with Linux.

Certain distributions of Linux are tailored to different CPU types.

Installation on New Systems

On newer systems, the Linux installation program often automatically identifies your hardware. It is still a good idea to gather information ahead of time, just in case the program does not figure out what you have.

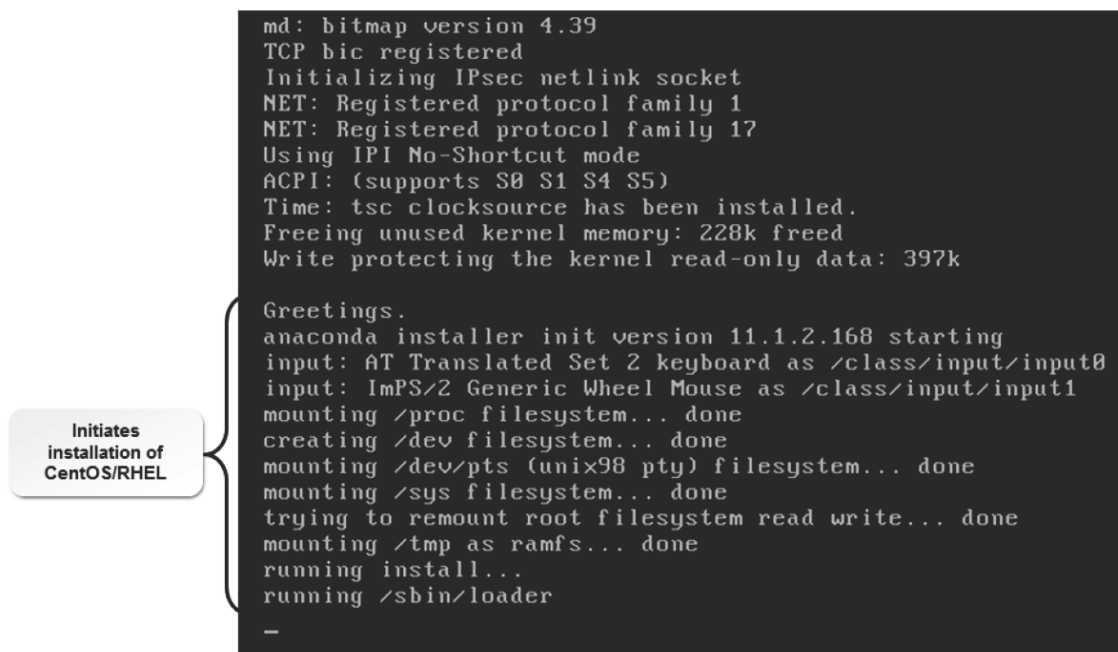
Gather Installation Information

Depending on your installed equipment, you may have to look for information in a variety of resources.

- One way to gather information is to view CMOS (Complementary Metal Oxide Semiconductor) or diagnostic information available at system boot time. Cold-boot your system and display setup or diagnostic information. This is displayed in different ways on different systems. For example, on most Compaq systems, when the block cursor moves to the upper-right corner of the screen, press **F10**. On some systems, when the "View Setup" message (or a similar message) is displayed, you press **Delete** (or the key indicated in the message).
- Access to manuals that came with the equipment can often be one of the best ways to find out about components. Manufacturers' websites also often contain valuable information about equipment.
- If you don't have the documentation and you don't have an application or a utility to provide this information, you will need to open the hood and look inside. Many cards, boards, and components have manufacturer and model information printed on them. However, they probably will not have a guide to interpreting the jumpers' settings.
- If your workstation is to be connected to a network, you will also need information about the network address to be assigned to your computer. Some systems will have a permanent network address and others will obtain a network address each time they access the network. Contact the system or network administrator to find out how to set up your system.

The Anaconda Installer

The [Anaconda installer](#) is a program for installing Linux through the text or graphical mode. It provides step-by-step instructions to guide you through the installation process. It also enables you to partition and organize hard disks and manage RAID and LVMs. The installer provides various options to choose from and allows you to add different packages based on your operating requirements.



```
md: bitmap version 4.39
TCP bic registered
Initializing IPsec netlink socket
NET: Registered protocol family 1
NET: Registered protocol family 17
Using IPI No-Shortcut mode
ACPI: (supports S0 S1 S4 S5)
Time: tsc clocksource has been installed.
Freeing unused kernel memory: 228k freed
Write protecting the kernel read-only data: 397k

Greetings.
anaconda installer init version 11.1.2.168 starting
input: AT Translated Set 2 keyboard as /class/input/input0
input: ImPS/2 Generic Wheel Mouse as /class/input/input1
mounting /proc filesystem... done
creating /dev filesystem... done
mounting /dev/pts (unix98 pty) filesystem... done
mounting /sys filesystem... done
trying to remount root filesystem read write... done
mounting /tmp as ramfs... done
running install...
running /sbin/loader
—
```

Initiates installation of CentOS/RHEL

Figure 17-1: The Anaconda installer in CentOS.

LVM

Logical Volume Manager (LVM) is a software tool that is used to manage the disk storage on a computer system.

LVM-based installations give the system administrator greater control over disk management. By creating logical volumes, an administrator can group disks or partitions together into manageable chunks that are more enterprise-capable than simple partitions. Two of the major features of LVM are the ability to create logical volume snapshots, and the resizing of logical volumes by absorbing or ejecting physical volumes.

- **pvcreate:** The Physical Volume Create utility prepares physical volumes to be used in logical volumes.
- **vgcreate:** The Volume Group Create tool creates and names volume groups.
- **lvcreate:** The Logical Volume Create tool creates and names logical volumes.

RAID

Redundant Array of Independent Disks (RAID) is a method that is used to store the same data in different locations on multiple hard disks of a server or a standalone disk storage system.

Partitioning Utilities

The Linux operating system is usually installed on a partition of the hard disk. A **partition utility** is a program that is used to manage partitions on the hard disk. It enables you to create a new partition, modify the attributes of an existing partition, assign a filesystem to a partition, and delete a partition. A partition utility also enables you to specify the size of a partition and indicate whether the partition is a primary or logical partition. The most frequently used partition utility is fdisk.

Disk druid is a program used for partitioning disk drives during the installation process.

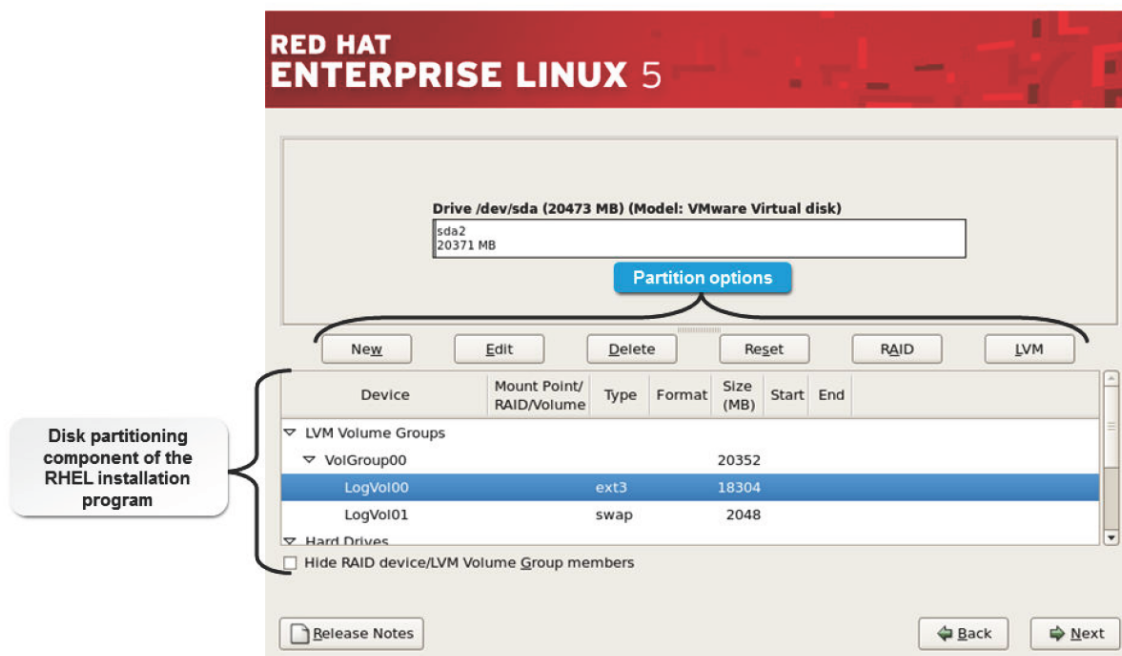


Figure 17-2: The disk partitioning component that appears during Red Hat Enterprise Linux installation.

Partition Device Name

The partition device name is often **/dev/hda1** or **/dev/hda2** for IDE disks and **/dev/sda1** or **/dev/sda2** for SCSI disks.

Repartitioning Strategies

If you have neither enough free space nor a partition to spare, then you can either add another drive to install Linux or repartition your existing drive. Most DOS-based systems have a single partition that includes the entire drive. This leaves you with no space to install Linux. You can either delete the existing partition and make it smaller to accommodate Linux, or use a partitioning utility to resize it.

Unless you use a special utility, you will have to delete and then re-create partitions of appropriate sizes. In recent years, several utilities have been developed that can move partition boundaries without destroying your data.

Destructive Repartitioning

Destructive repartitioning is a traditional method. You use the `fdisk` utility to delete the existing partition, and then re-create partitions. This means that you lose all the data saved on the partition when it is deleted. Therefore, before deleting the partition, if you want to retain any information, you will need to back it up, delete and re-create the partitions, and then restore the information.

Nondestructive Repartitioning

Several third-party utilities allow you to move partition boundaries without destroying the data currently stored in the partition. This is referred to as nondestructive repartitioning. You still have to be extremely careful and follow the utility's documentation directions exactly to avoid inadvertently destroying your data. Partition Magic and Partition-It are third-party utilities that can be purchased for this purpose.

Partitionless Installation

The partitionless installation option is available for users who want to try out Linux without installing it completely. All it requires is a formatted DOS FAT partition with enough free space for a workstation installation. However, it doesn't have all the options available that a full workstation installation would have, and you need to use a boot disk every time you want to run Linux. It is popularly known as Live CD.

The FIPS Program

The **First nondestructive Interactive Partition Splitting (FIPS)** program is a free utility that comes with some Linux distributions. It can be used to resize File Allocation Table (FAT) partitions.

When running FIPS, two partitions are created: the one you resized and the one FIPS creates.

FIPS Limitations

FIPS only works on DOS partitions with disk sector sizes of 512 bytes. Twelve-bit FATs will not be split by FIPS. Therefore, a partition will not be reduced below 4,085 clusters because this will require writing the 16-bit FAT as a 12-bit FAT. It also does not currently work with extended DOS partitions. If you already have four partitions, you cannot use FIPS to further split the partitions; FIPS requires a free partition entry with which it can work. Because of the wide variety of hardware and software configurations under which it must run, FIPS may not work properly on all systems.

BIOS

The **Basic Input/Output System (BIOS)** is a low-level firmware that acts as the interface between the hardware and the operating system on a computer. The BIOS settings can be modified according to the needs of a user. BIOS plays an important role in starting the boot process and determining the boot device settings. When a computer is powered on, BIOS is loaded into the memory, initiating the **Power-On Self Test (POST)**.



Note: The BIOS size varies among vendors and has a maximum size of 8 MB.

ROM BIOS

There are several BIOSes in your computer. When people say BIOS, they are generally referring to the main system BIOS. However, there are also BIOSes to control peripherals. Typically, the video card has its own BIOS, which contains hardware-driving instructions for displaying video information. SCSI host adapters, hard drives, and other peripherals can also contain their own BIOS instructions.

Plug and Play

When the main BIOS looks for the video card and other peripheral BIOSes, it will look for and configure plug-and-play devices if your BIOS supports the plug-and-play standard. When a plug-and-play device is found, BIOS displays a message on the screen prompting user input and action.

POST

POST takes an inventory of your system's hardware components in a specific order. The following table lists the components in the order in which they are tested.

Component	Test Description
Processor	If this test fails, the system halts without displaying any message.
ROM BIOS	A series of checksums are computed; if they don't match, the system halts.
DMA controller	If this test fails, the system halts.
Interrupt controller	If this test fails, you will hear a long beep followed by a short beep; the system then halts.
Timing chip	If this test fails, the system halts.
Video card	If this test fails, you will hear a long beep followed by two short beeps; if the test succeeds, ROM BIOS gets copied into RAM memory.

Component	Test Description
Expansion boards	Boards are initialized and, if necessary, the expansion board's ROM gets copied into the upper memory.
RAM memory	Counts and tests RAM by writing a bit to each memory bit.
Keyboard	Presence of keyboard and any stuck keys.
Other resources	Parallel and serial ports are queried; the system looks for an operating system to load.

Booting Devices

The common booting devices include the following:

- External Storage Device
- Serial Advanced Technology Attachment (SATA) or Serial Attached SCSI (SAS) Hard Drive
- CD-ROM or DVD-ROM
- Other Boot Device

CMOS

The **Complementary Metal Oxide Semiconductor (CMOS)** is a memory area with battery backup that is used to store system configuration settings. Prior to the use of CMOS, settings were configured with jumpers and switches. CMOS was introduced with the AT system boards. It allows more configuration options when compared to switches and jumpers. Some of the things you can configure through CMOS are:

- Password: You can specify whether a password is required following POST.
- Drive order: The order in which POST checks drives for the operating system.
- Memory: Some systems require you to specify in CMOS how much RAM is installed on the system.
- Drive type: Specifies the type of hard drive attached to the system.
- Display: Specifies the monitor type.

BIOS Variations

While setting the boot sequence, the BIOS settings and options will vary according to the BIOS version, type, and the system. For example, in some BIOSes, it will be the Boot Sequence, whereas in others it may be the Boot Order or Startup Sequence. Similarly, most options—right from accessing the BIOS screen to the various configuration settings—will vary. You can either read through the vendor manual or follow the instructions that appear on the BIOS screen to perform the desired task.

Common IRQ, DMA, and I/O Settings

In the case of an Interrupt Request (IRQ)/Input/Output (I/O) conflict with devices such as modems and sound cards, you can change the setting for these devices. After selecting the desired peripheral in the **Advanced Menu** screen of your BIOS, press **Enter** to view the options and select the IRQ/I/O options desired for the peripheral.

On modern hardware and Linux distributions, we no longer have to worry about IRQ, I/O, and Direct Memory Access (DMA) settings as these are handled automatically by computer architecture advancements towards the plug-and-play concept.

Guidelines to Prepare for Installation

Follow these general guidelines for the installation preparation process.

Prepare for Installation

To prepare for installation:

1. Gather basic system information about your computer, including information about:
 - Hardware
 - Software
 - Network environment
2. Verify that all hardware is compatible with Linux by checking it against the HCL.
3. Verify that the computer meets the minimum system requirements for the distribution you want to install.
4. Verify that the software you want to use after installing Linux will work.
5. Determine the purpose of the system.
6. Verify that your system hardware can handle the space and workload required for the purposes you need it for.
7. Plan the hard disk partitioning layout and the corresponding filesystems, including the size of the swap drive, depending on the physical RAM.
8. If necessary, check the installation media using the **Test the Media** option available on the Linux Installation CD.

How to Prepare for Installation

Follow these general procedures to prepare for installation.

Configure BIOS to Perform Preinstallation Check on an x86 Hardware

As part of the preinstallation tasks, you need to configure the BIOS settings to ensure that the system is suitable for hosting Linux OS. To configure the system to host Linux:

- Check the boot sequence.
 1. Access BIOS according to the manufacturer's instruction.
 2. Access the **Boot** menu.
 3. Order the devices in the desired boot sequence using the (+) or (-) keys to move the sequence up or down. The system will boot from the device that is at the top of the boot sequence. Ensure that the system is configured to boot from the DVD-ROM drive for installation.
 4. Save the setting and exit the **BIOS** screen.

Enable or Disable Integrated Peripherals

To enable or disable integrated peripherals such as serial ports, parallel ports, modems, and sound cards:

1. Access BIOS according to the manufacturer's instruction.
2. Access the **Advanced** menu.
3. Select the **I/O Device Configuration** option and press **Enter**.
4. Select the desired peripheral you want to enable or disable and press **Enter**.
 - To enable the peripheral, select the desired Interrupt/IRQ values and press **Enter**.
 - To disable the peripheral in case of an IRQ/IO conflict, select the **Disabled** option.
5. Press **Esc** to return to the **Advanced** menu.
6. Save the setting and exit **BIOS**.

TOPIC B The Linux Boot Process

In the last topic, you identified Linux compatible hardware and types of installation techniques. To understand how Linux is loaded on to your system, you need to learn about the boot process. In this topic, you will identify phases of the Linux boot sequence and its components.

The boot process is the most important process in system startup, and it is essential for proper loading of the operating system and all its applications. While installing Linux on multiple computers, it is important that you have sound knowledge of the boot process because it will help you identify and troubleshoot any issues related to system startup or the operating system.

Boot Loaders

A **boot loader** is a program that loads the kernel from the hard drive, or boot disk, and then starts the operating system. It is also referred to as the boot manager. Although boot loaders can load more than one operating system into the computer's memory, the user needs to select the desired operating system. Boot loaders interact with BIOS and utilize subroutines to load the operating system. In addition, they protect the boot process with a password.

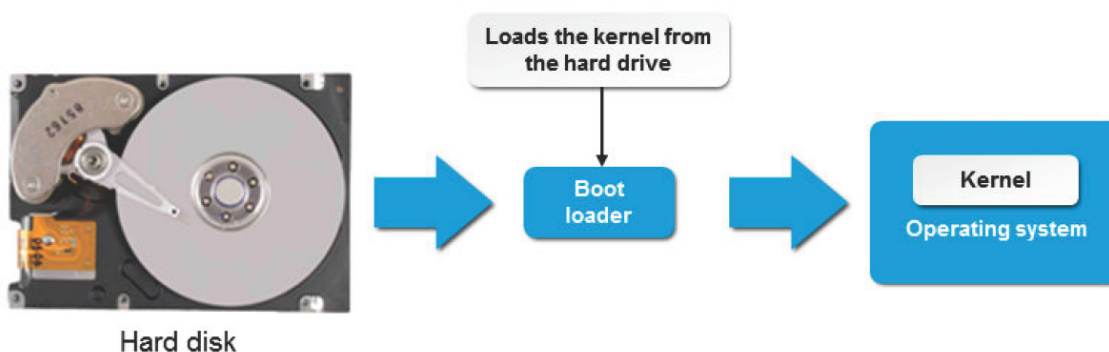


Figure 17-3: The function of a boot loader.

Boot Loader Components

The boot loader uses three main components that work together to systematically load the operating system in stages.

Component	Description
Boot sector program	It is the first component of the boot loader. It is loaded by BIOS on startup and has a fixed size of 512 bytes. Its main function is to load the second stage boot loader; however, it can also load another sector or a kernel.
Second stage boot loader	It loads the operating system and contains a kernel loader.

Component	Description
Boot loader installer	It controls the installation of disk sectors and can be run only when booting from a disk. It coordinates the activities of the boot sector and the boot loader.

Types of Boot Loaders

There are different types of boot loaders in Linux. Some of the boot loaders are described in the following table.

Boot Loader	Description
GRUB and GRUB 2	A popular Linux boot loader that allows you to place specific instructions in the Master Boot Record (MBR).
ELILO	A boot loader for Unified Extensible Firmware Interface (UEFI) machines. It supports flexible local booting from a FAT-32 filesystem and a wide variety of boot options via network booting over DHCP/TFTP.
System Commander	A third-party boot loader that can also be used as a boot manager. This utility enables you to control the environment you boot into. It has a full- featured boot manager and partitioning software.
SysLinux	An MS-DOS program that is sometimes used to simplify a first-time installation of Linux and for creating a rescue boot disc.

GRUB Loading Order

GRUB loads in the following order: the primary boot loader is loaded first, followed by the secondary boot loader and, finally, the operating system. After receiving the correct instructions for the operating system to start, GRUB locates the boot file and hands off control of the machine to that operating system.

UEFI

Earlier known as Extensible Firmware Interface (EFI), UEFI specifies an interface that operates between the operating system and the platform firmware. It is an alternative to BIOS for initiating a system, but does not completely replace BIOS.

GRUB

GRUB 2 is the newest version of the GRand Unified Bootloader (GRUB). The original version of GRUB is now referred to as GRUB Legacy and is no longer actively developed. Both versions are in use in Linux distributions. GRUB is the program that loads operating system kernels.

What's New in GRUB 2?

GRUB 2 is the new GRUB version being adopted by Linux distributions. However, GRUB 2 is more than simply a newer version of GRUB; it is a complete redesign and rewrite of the GRUB system. GRUB 2 offers administrators more control over the boot process, boot devices, and boot behavior. There are changes to configuration files, file names, and file locations in GRUB 2. Some of the major changes are outlined in the following table.

GRUB Legacy	GRUB 2
<code>/boot/grub/menu.lst</code> or <code>/boot/grub/ grub.conf</code>	<code>/boot/grub2/grub.cfg</code>
First partition number is 0	First partition number is 1
Manual update of <i>menu.lst</i> to include Linux kernels	Menu list of available Linux kernels auto- generated by running <code>update-grub</code>
No Loadable Modules	Loadable Modules
Menu title	Menu menuentry
Directly editable menu.lst file	Indirectly editable grub.cfg . Edit the grub.cfg file by editing entries in the <code>/etc/default/grub</code> file and files in the <code>/etc/grub.d</code> directory.

Superblocks

A **superblock**, often called *sb*, is a data structure that is stored on a disk and contains control information for a filesystem. Linux partitions are discussed in terms of blocks. A superblock comprises the first 512 bytes of a partition. It contains information about the block size used by the filesystem, the location of the root directory, and the time it was last checked.



Note: Each partition on a disk is identified by a number. The number 1 is assigned to the first partition, number 2 to the second partition, and so on.

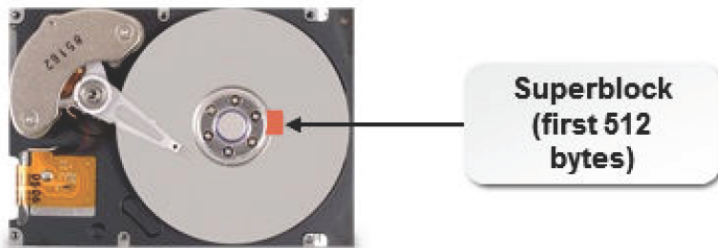


Figure 17-4: A superblock on a hard disk.

Sectors

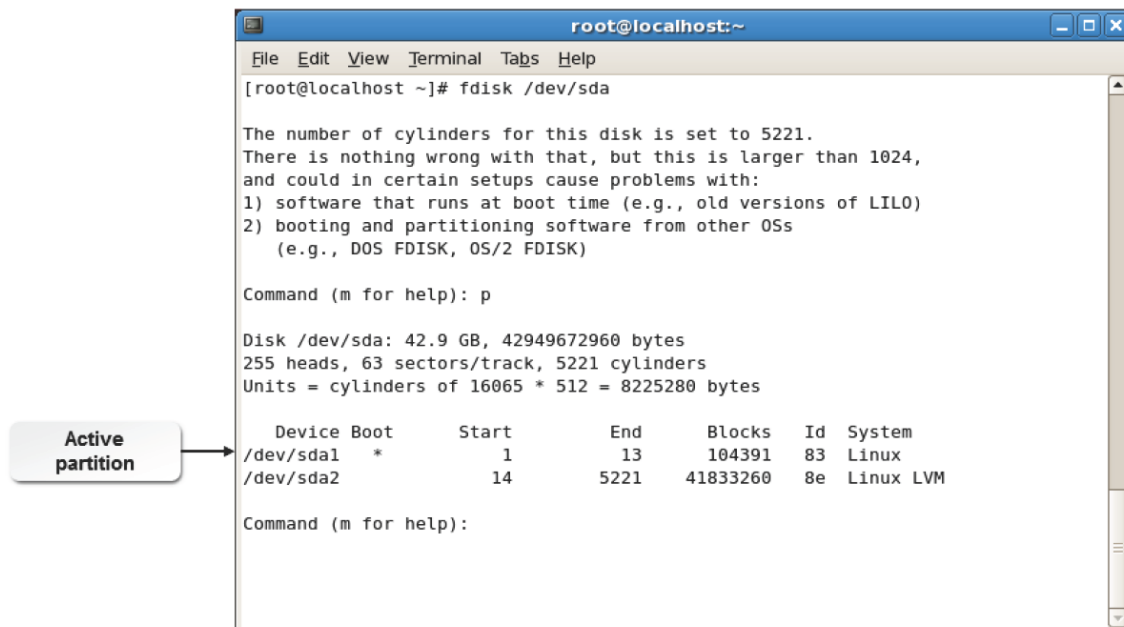
A **sector** is the smallest unit of storage read from or written onto a disk. A sector stores 512 bytes of data by default. A collection of sectors is called a track. The number of sectors in a track may vary, and so does their capacity to hold data. The size of a sector can be altered when formatting the hard disk.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# fdisk /dev/sda  
  
The number of cylinders for this disk is set to 5221.  
There is nothing wrong with that, but this is larger than 1024,  
and could in certain setups cause problems with:  
1) software that runs at boot time (e.g., old versions of LILO)  
2) booting and partitioning software from other OSs  
   (e.g., DOS FDISK, OS/2 FDISK)  
  
Command (m for help): p  
Disk /dev/sda: 42.9 GB, 42949672960 bytes  
255 heads, 63 sectors/track, 5221 cylinders  
Units = cylinders of 16065 * 512 = 8225280 bytes  
  
   Device Boot      Start         End      Blocks   Id  System  
/dev/sda1  *           1           13        104391   83   Linux  
/dev/sda2             14         5221       41833260   8e  Linux LVM  
  
Command (m for help):
```

Figure 17-5: A hard disk showing sectors and tracks.

MBR

Master Boot Record (MBR) is the first physical sector on a hard drive. It contains the code used for loading the operating system or boot loader into memory. It also contains the partition table of the hard drive. MBR helps determine the partition that is currently active.



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# fdisk /dev/sda  
  
The number of cylinders for this disk is set to 5221.  
There is nothing wrong with that, but this is larger than 1024,  
and could in certain setups cause problems with:  
1) software that runs at boot time (e.g., old versions of LILO)  
2) booting and partitioning software from other OSs  
   (e.g., DOS FDISK, OS/2 FDISK)  
  
Command (m for help): p  
  
Disk /dev/sda: 42.9 GB, 42949672960 bytes  
255 heads, 63 sectors/track, 5221 cylinders  
Units = cylinders of 16065 * 512 = 8225280 bytes  
  
   Device Boot      Start         End      Blocks   Id  System  
/dev/sda1  *         1           13       104391   83  Linux  
/dev/sda2             14        5221     41833260   8e  Linux LVM  
  
Command (m for help):
```

Figure 17-6: A hard disk showing MBR and the partition table loaded by it.

The Boot Process

The **boot process** is repeated each time your computer is started by loading the operating system from the hard drive. It involves a series of sequential steps that can be divided into BIOS initialization, boot loader, kernel and init initialization, and boot scripts.

The boot process consists of the following steps:

1. The processor checks for the BIOS program and executes it.
2. BIOS checks for peripherals, such as floppy disk drives, CD-ROMs, and the hard disk, for bootable media. It locates a valid device to boot the system.
3. BIOS loads the primary boot loader from the MBR into memory. The boot loader is a program that contains instructions required to boot a machine. It also loads the partition table along with it.
4. The user is prompted with a graphical screen that displays the different operating systems available in the system to boot from. The user should select an operating system and press **Enter** to boot the system. If the user does not respond, then the default operating system will be booted.
5. The boot loader determines the kernel and locates the corresponding kernel binary. It then uploads the respective initrd image into memory and transfers control of the boot process to the kernel.
6. The kernel configures the available hardware, including processors, I/O subsystems, and storage devices. It decompresses the initrd image and mounts it to load the necessary drivers. If the system implemented any virtual devices, such as LVM or software RAID, then they are initialized. The components configured by the kernel will be displayed one by one on the screen.
7. The kernel mounts the root partition and releases unused memory. To set up the user environment, the init program is executed.
8. The init program searches for the inittab file, which contains details of the runlevel that has to be started. It sets the environment path, checks the filesystem, initializes the serial ports, and runs background processes for the runlevel.
9. If graphical mode is selected, then xdm or kdm is started and the login window is displayed on the screen.
10. The user enters the user name and password to log in to the system.

11. The system authenticates the user. If the user is valid, then the profile, the .login, the .bash_login, and the .bash_profile files are executed. The shell is started and the system is ready for the user to work on.



Note: xdm refers to the X Window Desktop Manager. Users who utilize GNOME or KDE, use either gdm or kdm, respectively. In CentOS 7, the GNOME Display Manager gdm is the default desktop manager.

TOPIC C Configure GRUB

In the last topic, you discussed the boot sequence. To manage the boot process, you must understand how to use and configure the components involved in it. By configuring GRUB 2, you can modify the system to run it according to your requirements. In this topic, you will configure the GRUB 2 boot loader and understand its functions.

As a Linux administrator, you may be assigned the task of running multiple operating systems on the same system. In such a case, you must know how to add new kernels and boot the correct operating system. To accomplish this task, you should know about GRUB and how to configure it.

GRUB 2

GRand Unified Bootloader 2 (GRUB 2) is a program that is used to install a boot loader in MBR.

GRUB allows you to place specific instructions in MBR to load a GRUB menu or environment command. This enables you to start the operating system of your choice, pass instructions to the kernel when booting, or check for system parameters before booting.



Note: Now that GRUB 2 usage is widespread, earlier versions of GRUB are often referred to as GRUB Legacy.

Allows you to select
the OS and kernel

```
CentOS Linux, with Linux 3.10.0-123.el7.x86_64
CentOS Linux, with Linux 0-rescue-375aa607829142d98b4abd41ab7d8a0a
```

```
Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
```

Figure 17-7: The GRUB 2 boot loader menu screen with its various functions.

GRUB 2 Configuration

GRUB 2 configuration is managed via files in three separate locations, as outlined in the following table.

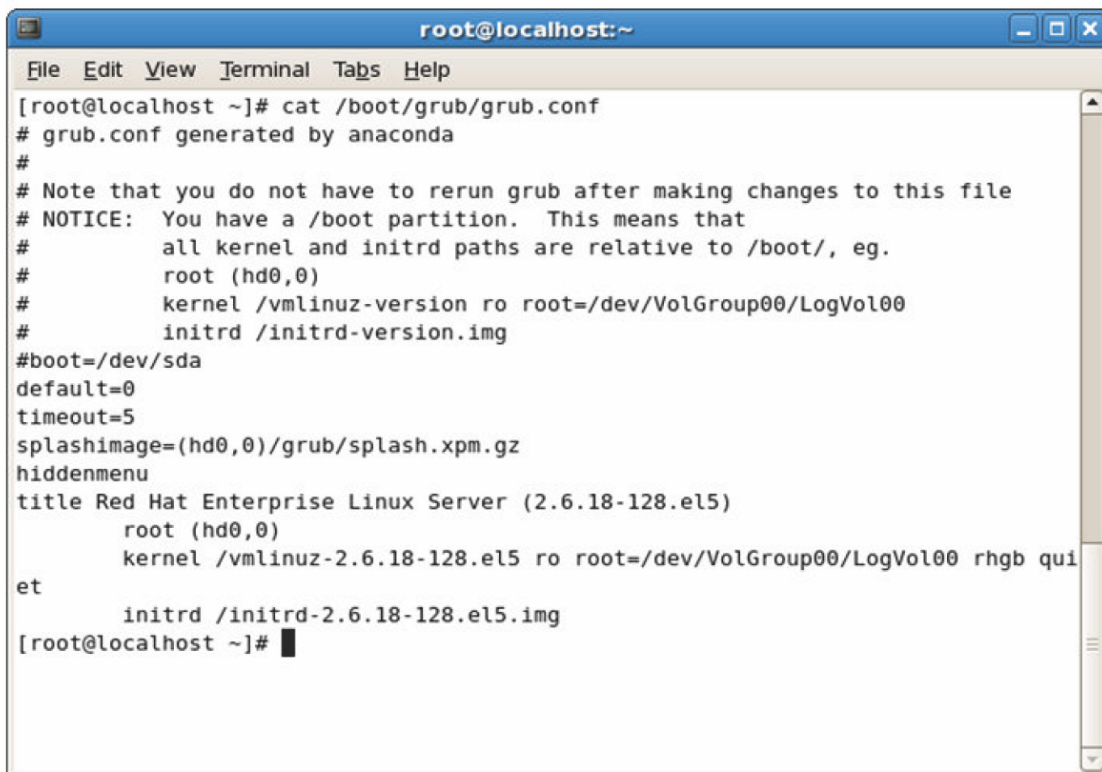
File Location	Purpose
---------------	---------

<i>File Location</i>	<i>Purpose</i>
boot/grub2/grub.cfg	This is the main configuration file for GRUB 2, and is generated from the files in this table. It is not advisable to edit this file by hand, as edits to the other files will overwrite changes made directly to this configuration file. This file replaces the former grub.conf and menu.lst files in earlier versions of GRUB. In some Linux distributions, this file is stored as /boot/grub/grub.cfg .
/etc/grub.d/	GRUB scripts placed in this directory form the key elements of the final grub.cfg file.
/etc/default/grub	This configuration file sets the default GRUB menu settings, which are incorporated into the final grub.cfg file.

The GRUB Legacy grub.conf and menu.lst Files

The **grub.conf** file found in the **/boot/grub** directory is the configuration file for the GRUB boot manager. It contains various configuration options for configuring and troubleshooting the boot manager.

<i>Option</i>	<i>Enables You To</i>
default={number}	Specify the default booting kernel number if multiple kernel images are found.
timeout={number}	Specify the time limit for the login screen to be displayed.
splashimage=(hdx,y)/grub/ {image location}	Specify the location of the login screen image.
title {user desired name}	Specify a title to differentiate between the kernel images in the login screen.
root (hdx,y)	Specify the location of MBR.
kernel {location} [option]	Specify the location of the kernel.
initrd {kernel image}	Specify the location of the kernel image.



```

root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# cat /boot/grub/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#         all kernel and initrd paths are relative to /boot/, eg.
#         root (hd0,0)
#         kernel /vmlinuz-version ro root=/dev/VolGroup00/LogVol00
#         initrd /initrd-version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.18-128.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-128.el5 ro root=/dev/VolGroup00/LogVol00 rhgb qui
et
    initrd /initrd-2.6.18-128.el5.img
[root@localhost ~]#

```

Figure 17-8: The grub.conf file that is used to configure the GRUB boot manager.

The GRUB Menu Configuration File

The **menu.lst** file is a GRUB configuration file that lists all the kernels available on the system along with their partition numbers, boot information, and details of which kernel is booted. The **menu.lst** file is stored in the

/boot/grub/ directory.

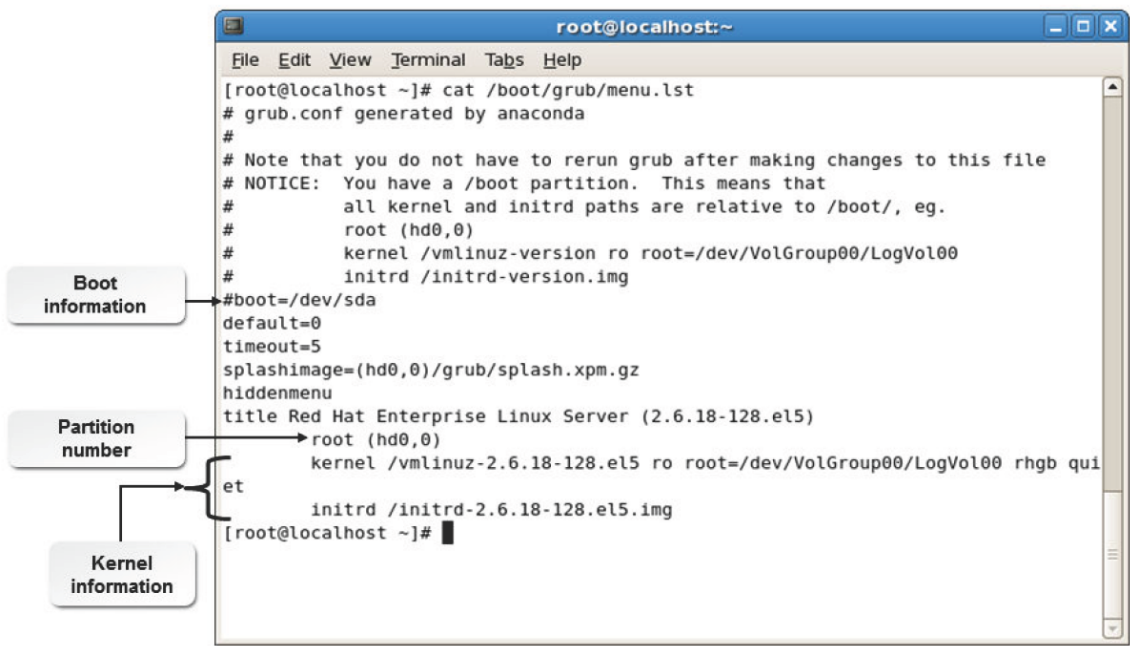


Figure 17-9: The menu.lst file displaying the system's kernel details.

GRUB Legacy Commands

GRUB commands enable a user to configure and modify the GRUB settings in each runlevel, based on user requirements. GRUB commands are categorized as general and Command Line Interface (CLI) commands.

General Command	Enables You To
bootp	Use the BOOTP protocol to initialize a network device.
device	Create a disk image and specify a file as a BIOS drive. This command is also used to troubleshoot GRUB in case of drive errors.
dhcp	Use the DHCP protocol to initialize a network device.
password	Set a password for the menu files. The locked files will not have the edit property set to them.
ifconfig	Configure a network manually. The gateway, IP address, subnet mask, and server address can be configured using this command.
terminal	Specify the terminal settings. Serial ports can be used only if this command is specified.
serial	Configure settings for various serial devices and serial ports.

CLI Command	Enables You To
boot	Load the operating system into the computer from the CLI.
cat {file name}	Display the content of a file.
find	Search for a file.
setup	Install and configure various services such as authentication, firewalls, and system services.
install	Install GRUB and other utilities.
kernel	Load a kernel boot image.
lmodule	Load a kernel module.
halt	Shutdown your system.
reboot	Reboot your system.
exit	Exit from the GRUB shell.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
  
GNU GRUB version 0.97 (640K lower / 3072K upper memory)  
  
[ Minimal BASH-like line editing is supported. For the first word, TAB  
lists possible command completions. Anywhere else TAB lists the possible  
completions of a device/filename.]  
  
grub>  
Possible commands are: blocklist boot cat chainloader clear cmp color configfi  
le debug device displayapm displaymem dump embed find fstest geometry halt help  
hide impsprobe initrd install ioprobe kernel lock makeactive map md5crypt modu  
le modulenounzip pager partnew parttype password pause quit read reboot root ro  
otnoverify savedefault serial setkey setup terminal terminfo testload testvbe u  
nhide uppermem vbeprobe  
  
grub> █
```

Figure 17-10: The commands that can be executed at the GRUB prompt.


GRUB Legacy Menu-Specific Commands

Menu-specific commands are used to configure GRUB from the configuration file. They can be enabled in the global section of the **grub.conf** configuration file.

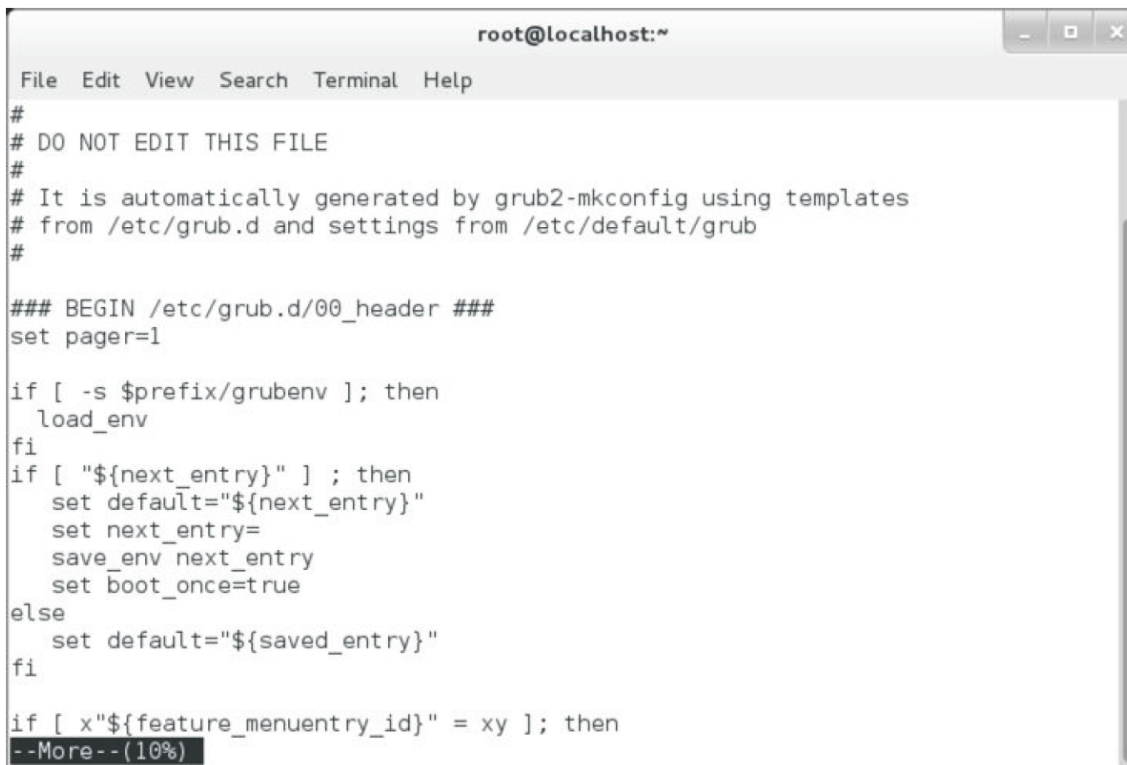
Command	Enables You To
default	Set the default entry for the entry number NUM, which is used by GRUB in case of boot entry errors.
fallback	Set the fallback entry, allowing GRUB to override any errors in the boot entry.
hiddenmenu	Hide the menu control from a user at the control terminal. This does not affect the boot entry.
timeout	Set the timeout value before booting into the default boot entry. The hiddenmenu command can be disabled here by pressing Esc before timeout elapses.
title	Start a new boot entry, which is displayed on the menu interface.

GRUB 2 Configuration

The **grub.cfg** file found in the **/boot/grub2** directory is the main configuration file for the GRUB 2 boot manager. Unlike the original GRUB configuration file, **grub.cfg** takes the form of a script and is generated from configuration information stored in the **/etc/default/grub** and **/etc/grub.d/** directory.



Note: On some Linux distributions, such as Ubuntu, the GRUB 2 **grub.cfg** configuration file is stored in the **/boot/grub** directory.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal displays the beginning of a GRUB configuration file. It starts with several comment lines explaining that the file is generated by grub2-mkconfig. Then, it sets 'set pager=1'. A conditional block follows: 'if [-s \$prefix/grubenv]; then load_env fi'. Inside this block, another conditional 'if ["\${next_entry}"] ; then' sets 'default=\${next_entry}', 'next_entry=', 'save_env next_entry', and 'boot_once=true'. An 'else' branch sets 'default=\${saved_entry}'. After the 'fi' for the first conditional, there is another 'if [x"\${feature_menuentry_id}" = xy]; then' line. The terminal ends with '--More--(10%)' on a highlighted line.

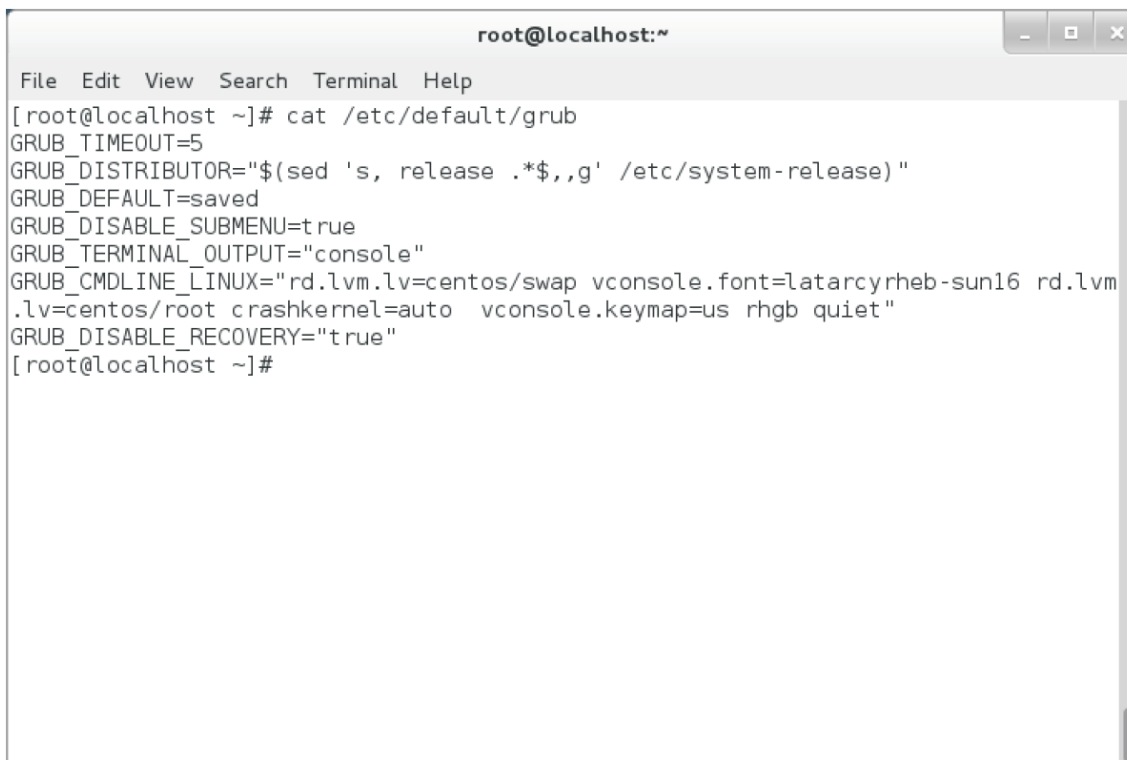
```
root@localhost:~  
File Edit View Search Terminal Help  
#  
# DO NOT EDIT THIS FILE  
#  
# It is automatically generated by grub2-mkconfig using templates  
# from /etc/grub.d and settings from /etc/default/grub  
#  
### BEGIN /etc/grub.d/00_header ###  
set pager=1  
  
if [ -s $prefix/grubenv ]; then  
    load_env  
fi  
if [ "${next_entry}" ] ; then  
    set default="${next_entry}"  
    set next_entry=  
    save_env next_entry  
    set boot_once=true  
else  
    set default="${saved_entry}"  
fi  
  
if [ x"${feature_menuentry_id}" = xy ]; then  
--More--(10%)
```

Figure 17-11: A portion of the generated grub.cfg configuration file.

The GRUB 2 Menu Configuration File

The [/etc/default/grub](#) file is a GRUB configuration file that lists all the kernels available on the system along with their partition numbers, boot information, and details of which kernel is booted.

The **menu.lst** file is stored in the **/boot/grub/** directory.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command '[root@localhost ~]# cat /etc/default/grub' and its output. The output lists various GRUB configuration variables: GRUB_TIMEOUT=5, GRUB_DISTRIBUTOR=\$(sed 's, release .*\$,,g' /etc/system-release), GRUB_DEFAULT=saved, GRUB_DISABLE_SUBMENU=true, GRUB_TERMINAL_OUTPUT="console", GRUB_CMDLINE_LINUX="rd.lvm.lv=centos/swap vconsole.font=latarcyrheb-sun16 rd.lvm.lv=centos/root crashkernel=auto vconsole.keymap=us rhgb quiet", and GRUB_DISABLE_RECOVERY="true". The terminal ends with '[root@localhost ~]#'.

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# cat /etc/default/grub  
GRUB_TIMEOUT=5  
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"  
GRUB_DEFAULT=saved  
GRUB_DISABLE_SUBMENU=true  
GRUB_TERMINAL_OUTPUT="console"  
GRUB_CMDLINE_LINUX="rd.lvm.lv=centos/swap vconsole.font=latarcyrheb-sun16 rd.lvm.lv=centos/root crashkernel=auto vconsole.keymap=us rhgb quiet"  
GRUB_DISABLE_RECOVERY="true"  
[root@localhost ~]#
```

Figure 17-12: The /etc/default/grub configuration file.

The /etc/grub.d/ Configuration Files

The [/etc/grub.d/](#) file is a GRUB 2 directory that contains configuration files that are used to configure the specific operations and menu options for the GRUB 2 boot manager.



Figure 17-13: The list of default [/etc/grub.d/](#) configuration files.

The grub2-mkconfig Command

The grub2-mkconfig command generates a new GRUB 2 configuration, and may be used to update the [/boot/grub2/grub.cfg](#) configuration file. The grub2-mkconfig utility combines the configuration file templates in the [/etc/grub.d/](#) directory with the settings in [/etc/default/](#)

grub to generate the [/boot/grub2/grub.cfg](#) ([/boot/grub/grub.cfg](#)) GRUB 2 configuration file.

Note: On CentOS, RHEL, and Fedora Linux distributions, the **grub2-mkconfig** command is named **grub2-mkconfig** to be consistent with all other GRUB 2 commands on these distributions. On other distributions, this command is often simply **grub-mkconfig**.

Syntax

The syntax of the grub2-mkconfig utility is `grub2-mkconfig -o {/boot/grub2/grub.cfg}`.

How to Configure GRUB

Follow these general procedures to configure GRUB.

Configure the GRUB Boot Loader

To configure the GRUB boot loader:

1. Log in to the CLI as **root**.
2. To open the **[/etc/default/grub](#)** file, enter `vi /etc/default/grub`.
3. Make the necessary changes to specify alternative timeouts, recovery settings, and default options.

4. Save and close the file.

Protect GRUB 2 Boot Configuration with a Password

To protect GRUB 2 with a password:

1. Log in to the CLI as **root**.
2. To create an PBKDF2 encrypted password, enter `grub2-mkpasswd-pbkdf2`.
3. To navigate to the **/etc/grub.d/** directory, enter `cd /etc/grub.d`.
4. To open the **40_custom** file, enter `vi 40_custom`.
5. To navigate to the end of the file, press **G**.
6. To switch to insert mode, press **I**.
7. To specify the username, on a new line type `set superusers="{username}"`.
8. To specify the password, on a new line below the **set superusers** entry, type `password_pbkdf2 {username} {PBKDF2-encrypted password}`.
9. Save and close the file.

Install GRUB 2 as the Boot Loader

To install GRUB 2 as the boot loader:

1. Log in to the CLI as **root**.
2. To generate a new version of the **grub.cfg** file, enter `grub2-mkconfig -o /boot/grub2/grub.cfg`.
3. Verify that no errors are displayed and the "done" message is displayed. This indicates that GRUB 2 has been installed successfully.
4. To identify the boot device of the system, enter `cat /boot/grub2/device.map`.
5. To install the GRUB 2 boot loader with the new configuration, enter `grub2-install {boot device}`.
6. To check if the system boots with the specified boot loader, enter `reboot`.

Boot from Menu Editing Mode

To boot from menu editing mode:

1. Start the system.
2. On the GRUB 2 boot splash screen, press **Esc**.
3. To enter menu editing mode, select an entry and press **E**.
 - To edit a line, select the line and press **E**.
 - To return to the menu, press **Esc**.

- To open the GRUB shell, select a line and press **C**.

4. To boot the system, select the desired menu option and press **Enter**.

TOPIC D Install the Operating System

You may have installed various operating systems previously. However, installation techniques do not apply universally to all operating systems. With a clear understanding of the boot process, you will be prepared to begin the installation of Linux. In this topic, you will install Linux on a computer after ensuring that the computer is suitable to host it.

Installation is perhaps the most important aspect in the Linux operating system. It involves many major tasks such as creating and configuring partitions and devices. Also, Linux can be installed in different ways. As a Linux administrator, you may be required to install and reinstall Linux on a number of systems. Knowing how to administer Linux installation will enable you to utilize the potential of the features packed into Linux to the optimum.

Linux Installation Methods

Linux can be installed on servers and workstations using different methods.

<i>Installation Method</i>	<i>Description</i>
Local CD/DVD-ROM installation	Linux can be installed from a set of installation CD/DVD-ROMs. It requires the system's BIOS settings to support booting from CD/ DVD-ROMs. This is like a local installation and is the easiest way to install Linux.
Local hard drive installation	Linux installation can be done by staging the installation files on the local hard drive.
USB drive installation	Linux can also be installed through USB drives if CD/DVD-ROMs or other modes of installation are not supported by the system. To enable booting from USB drives, the diskboot.img file has to be copied from the images folder of the installation CD/DVD-ROM to the USB drive. This mode of installation also requires BIOS to support booting from USB drives.
Network-based installation	Linux installation can be done on networked computers by staging all the installation files on a separate server and installing it on clients' systems. The network installation server shares the installation directory with the clients via NFS, FTP, or HTTP. This method is often faster than CD/DVD-ROM-based installations. The network installation server is necessary for all network-based Linux installations.

Documentation

Sufficient and proper documentation of setups, configurations, topologies, and histories can prove valuable while troubleshooting. This includes documenting the hardware and software components installed, why and when they were installed, by whom, and other important and specific information.

These log files contain information captured during the installation. You may want to include hard copies of these files in your installation documentation.

- `/var/log/messages`
- `/var/log/dmesg`
- `/var/log/yum.log`
- `/root/anaconda-ks.cfg`
- `/root/initial-setup-ks.cfg`

How to Install the CentOS Linux Operating System

Follow these general procedures to install the CentOS Linux operating system.

Install Linux from a CD/DVD

To install Linux from a CD-ROM or DVD:

1. Insert the CD/DVD into the drive and boot the system.
2. At the boot menu, select to **Install CentOS 7**.
3. If necessary, test the installation media and start the installation.
4. Choose the desired language, and select **Continue**.
5. Select from the **Installation Summary** options to confirm the **Date & Time**, **Installation Source**, **Software Selection**, **Installation Destination** (and partitioning scheme), and **Network configuration**.
6. Once details have been configured, select **Begin Installation** to begin the Linux installation.
7. While installation occurs, you may configure the root password and an initial system user on the **User Settings** screen.
8. On the installation complete page, select **Reboot** to finish the installation and reboot the system.

Be sure to remove your installation media from the CD/DVD drive.

Install Linux via a Network

To install Linux via a network, use a CD-ROM based on the CentOS 7 NetInstall image:

1. At the boot prompt, choose the mode of installation.
2. Select the appropriate network installation medium and press **Enter**.
3. Choose the desired language.
4. Choose the desired keyboard type.
5. Choose the type of installation method.
6. Configure the network settings.
7. Specify the remote install server information, such as http://mirror.centos.org/centos/7/os/x86_64/.
8. If necessary, check the boot media and start the installation.
9. Select **Next** to begin the installation.
10. A warning message will be displayed if you are using a new hard disk; select **Yes** to continue.
11. From the partition layout drop-down list, select the desired partition layout and select **Next**.
12. If a warning message is displayed, select **Yes** to continue.
13. If necessary, review the partition table and select **Next**.
14. On the boot loader installation page, choose the type of boot loader and its location and then select **Next**.
15. On the network configuration page, configure the network and select **Next**.
16. Select the desired time zone for the machine and select **Next**.

17. Enter the root password and select **Next**.
18. Accept the default package list or select **Customize now** and select **Next**.
19. Select the necessary package and select **Next**.
20. On the installation complete page, select **Reboot** to finish the installation and reboot the system.

Access the Network Installation Server

To access the network installation server:

1. Log in as **root**.
2. Insert the installation CD into the CD-ROM drive and mount it using the command `mount /dev/cdrom /{mount point}`.
3. To copy the installation image into the destination folder, enter `cp -R /{mount point}/* / {destination directory}/`.
4. Replace the CD with the next installation CD and perform steps 2 and 3.
5. Configure the server to be used during the remote installation.
 - a. Configure the NFS server for network installation.
 - i. To open the **exports** file, enter `vi /etc/exports`.
 - ii. To specify the destination directory for obtaining installation files, type `/ {destination directory} [options]`.
 - iii. Save and close the file.
 - iv. To start the NFS server, enter `service nfs start`.
 - v. To export the directory, enter `exportfs -r`.
 - b. Configure the FTP server for network installation.
 - i. Ensure that `/ {destination directory}` is `/var/ftp/pub` for FTP-based installation.
 - ii. To specify the destination directory for obtaining installation files, type `/ {destination directory} [options]`.
 - iii. Save and close the file.
 - iv. To start the FTP server, enter `service vsftpd start`.
 - c. Configure the HTTP server for network installation.
 - i. Ensure that `/ {destination directory}` is `/var/www/html` for HTTP-based installation.
 - ii. To specify the destination directory for obtaining installation files, type `/ {destination directory} [options]`.
 - iii. Save and close the file.
 - iv. To start the **httpd** server, enter `service httpd start`.

Create a Boot Media

To create a boot media:

1. Log in as **root**.
2. Create a boot media.
 - a. Create a boot CD.
 - i. To copy the boot image to the specified location, at the command prompt, enter `cp /media/images/boot.iso {destination directory}`.
 - ii. To navigate to that location, enter `cd {destination directory}`.
 - iii. To create the boot CD, enter `cdrecord -v boot.iso`.
 - b. Create a boot USB drive.
 - i. To redirect the **bootdisk.img** content to where the USB device node is located, at the command prompt, enter `cat /media/images/bootdisk.img /dev/{device name} {device number}`.

ACTIVITY 17-1

Installing Linux Review

Scenario

Answer the following review questions.

1. How does the boot process affect the applications installed on a system?
Why?
2. Which is the best runlevel for you to boot a system in your organization?
Why?

Summary

In this lesson, you installed the Linux operating system. You also performed pre-installation and post-installation tasks and documented your actions. These steps ensured your successful installation of the Linux operating system.

18 Configuring the GUI

Lesson Time: 1 hour, 15 minutes

Lesson Introduction

In the previous lessons, you used the Command Line Interface (CLI) to perform tasks in Red Hat® Enterprise Linux®. However, for those who are not comfortable with the CLI, Linux also provides a more user-friendly Graphical User Interface (GUI) for system management and maintenance tasks. In this lesson, you will configure the GUI.

Linux provides the flexibility of switching back and forth between the CLI and the GUI.

While the command line allows you to perform an action with speed, the GUI is more user-friendly and allows you to find options and functions easily when you cannot remember the corresponding commands.

Lesson Objectives

In this lesson, you will configure the GUI. You will:

- Implement X.
- Customize the display manager.
- Enable accessibility settings.

TOPIC A Implement X

In the last lesson, you performed post-installation tasks. Sometimes, when you are guiding users through a process, they may not understand the commands you tell them to type in the CLI. In such situations, you can choose the GUI because it is more user-friendly. Combining the GUI and the CLI in Linux provides users with greater control and a greater number of options. In this topic, you will implement X to work with the GUI.

Because Linux provides both the CLI and GUI, users can choose to work in either one of them or both. Some users may not like the blank screen of the command line. They may prefer to work with more user-friendly icons and windows. Also, they may not always remember the commands to carry out a task. In such cases, they can use the GUI to accomplish the task.

X Windows

X Windows, also known as **X** or **X11**, is a client/server, multiuser system that resides on top of the operating system. The X Window system configuration option provides you with a choice of using a graphical or text-based interface. The primary configuration file defines hardware devices and other critical components of the X server environment. Configuration options include monitor, video card, keyboard, pointing device, and many more.

X.Org

X.Org is a free version of the X Window GUI system for some Linux distributions. It provides an interface between input hardware, such as the mouse and the keyboard, and the desktop environment. It is platform independent and extensible because it can be modified by changing or adding new features.

Configuring X

Newer versions of X.Org configure themselves automatically each time they are started. The Xorg command may be used to generate a manual configuration file to set the system resolution, color depth, and other advanced display settings.

The xorg.conf File

The **xorg.conf** file is a configuration file for X.Org. This file is used for configuring different X Window parameters and its default location is **/etc/X11/xorg.conf**.

X Servers

An **X server** is a program that implements the GUI by providing the X Window system. It runs on a local machine and manages the keyboard, mouse, and display device. It converts the X Windows protocol commands to machine language commands. It also converts the GUI commands to X Windows protocol commands for clients. An X server can draw pictures and display text on screen.

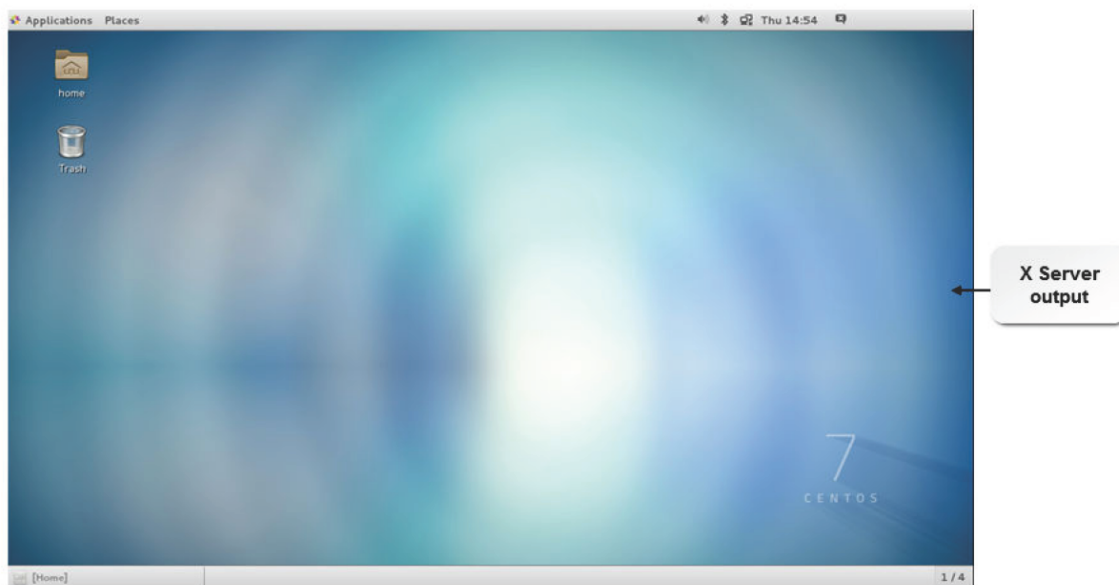


Figure 18-1: The GUI desktop using Linux CentOS 7.

The X Protocol

The **X protocol** is the standard protocol used by clients and servers in the X Window system. Using this protocol, requests for window operations can be exchanged.

The xdpinfo Utility

The xdpinfo utility displays information about the X server. The details displayed include values of parameters related to communication between clients and server, available screen types, and available visual types. By default, the utility displays statistics related to the X server. The statistics cover details such as the name of display, version number, vendor details, and pixmap formats.

The xwininfo Utility

The xwininfo utility displays information about the currently selected window in the GUI. The window can be selected by using the mouse, specifying its id, or specifying its name. The details displayed include the window id; the position of the four window coordinates, width, height, color, and depth; and various states of the window such as Bit Gravity State, Window Gravity State, and Map State.

Refresh Rate

Refresh rate or vertical scan rate is the speed at which a screen is refreshed. Normally, color displays are refreshed 60 times per second.

Resolution and Color Depth

Resolution is the number of pixels that a computer monitor is capable of displaying. It is described in terms of width x height. The most common resolutions are 640 x 480, 800 x 600, and 1,024 x 768.

Color depth refers to the number of colors used to display an image. The values can range from 256 colors to millions of colors. The size of a file increases with the increase in the color depth value.

The xvidtune Command

The xvidtune command displays the **xvidtune** dialog box to configure the horizontal and vertical display settings.



Note: This command, when wrongly used, may cause permanent damage to the monitor or video card. Therefore, you must ensure that you do not change any setting without fully understanding the purpose of the setting.

X Clients

An **X client** is an application that is written with the aid of the Xlib library, which gives programs access to any X server. An X client sends requests to the X server for a certain action to take place, for example, to create a window. In response to the request, the X server sends the event that the X client is expecting. An X client also receives errors in requests from the server. There can be more than one X client sending requests to the X server.

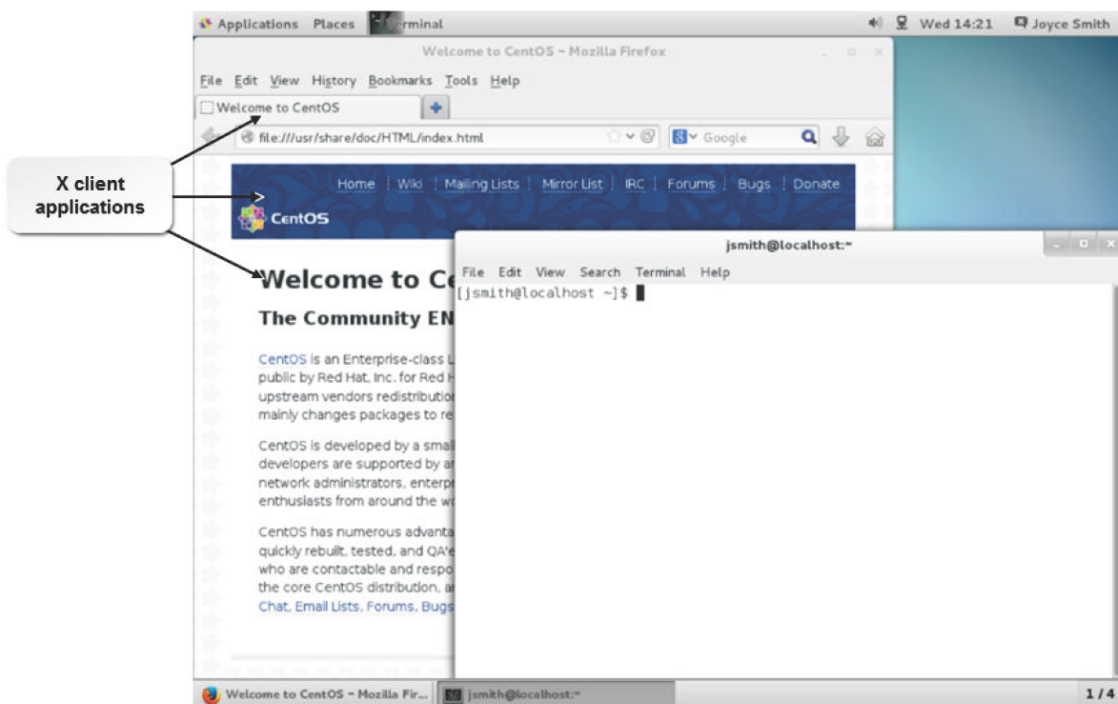


Figure 18-2: X client applications on Linux CentOS 7.

X Font Servers

An **X font server (Xfs)** is a service that provides fonts to the X server and X client applications that connect to the X server. The **font path**, which is a collection of paths in the filesystem where font files are stored, can also be edited using Xfs. Fonts may be stored on one machine, which acts as a networked font server. Multiple X servers can share these fonts over the network. Xfs supports the TrueType, Type1, and bitmap fonts. The X font server has been deprecated in more recent configurations in favor of client-side fonts, whereby the fonts are rendered by the X clients, not the X server.

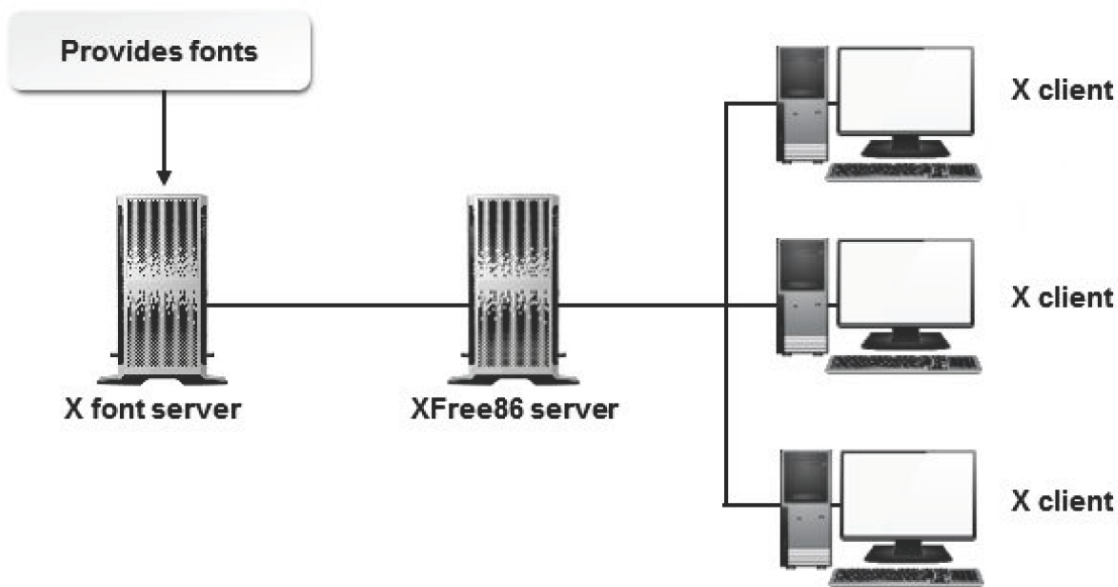


Figure 18-3: *The X font server providing fonts to clients.*

Virtual Desktops

There will be times when a user needs to have multiple windows open on their desktop. Typically, to help manage the need to go between many different programs running simultaneously, users would do one of three things:

- Leave all windows open, which may result in a cluttered desktop.
- Minimize those windows that are not needed and use the taskbar or press **Alt+Tab** to switch between them. This approach could still be a bit confusing.
- Use virtual desktops.

The default configuration provides two or four desktops depending on the distribution. For example, Ubuntu® provides two desktops by default and CentOS provides four. Users can switch between the virtual desktops by clicking one of the desktop buttons on the panel at the bottom of the desktop screen.

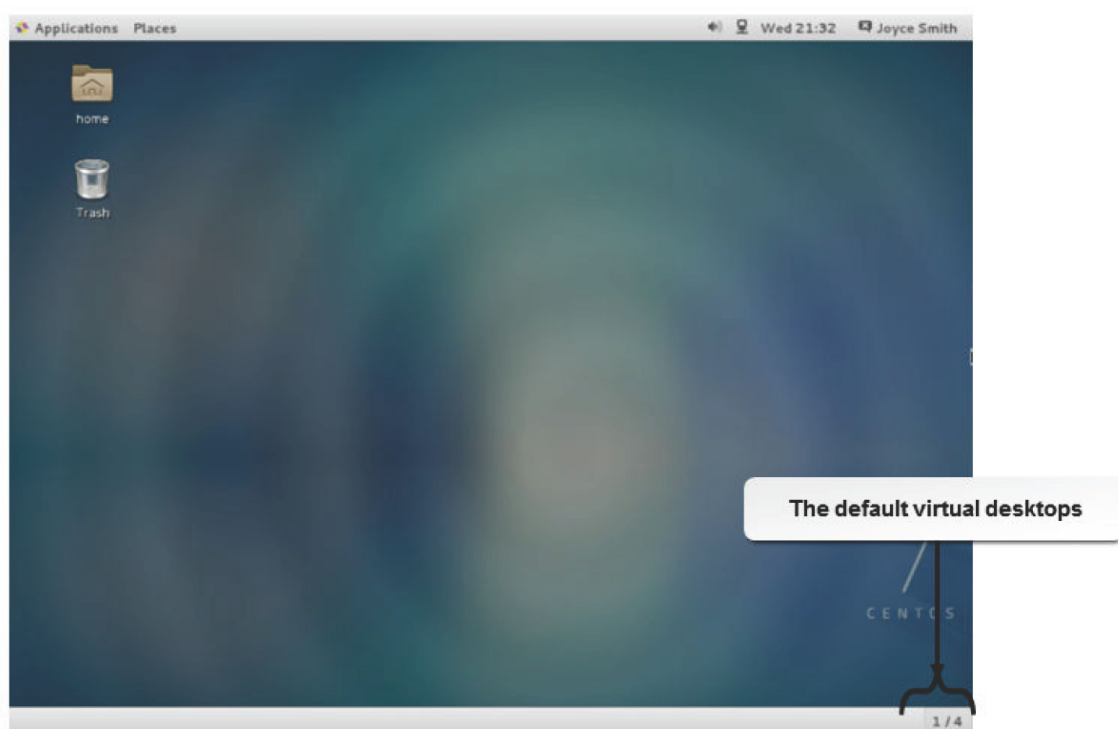


Figure 18-4: *Virtual desktop buttons on the panel.*

X.Org Runlevels

The X Window system boots in two main runlevels: runlevel 3 (multi-user.target) and runlevel 5 (graphical.target). When you start a machine, it boots in graphical mode, which is runlevel 5 (graphical.target). You can also boot the machine in CLI or text mode, which is runlevel 3 (multi-user.target). Runlevel 3 is full multiuser mode. The X server is started from runlevel 3 (multi-user.target) using the startx or xinit command.

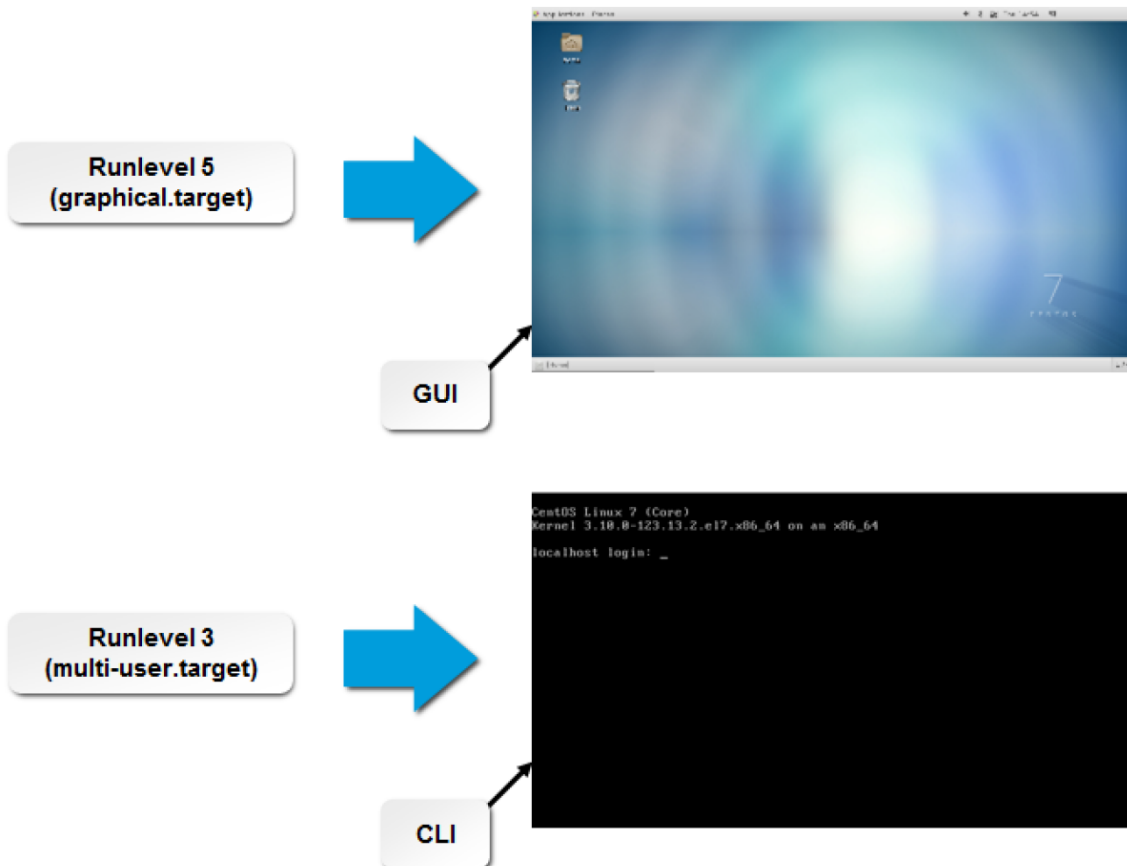


Figure 18-5: Booting in different runlevels results in different displays (runlevel 5, graphical.target).

Remote X Sessions

Remote X sessions are sessions where a user on a remote workstation is able to view the X Window of the host and run the host's applications. These sessions can run on local and TCP/IP networks. Remote X sessions can be either host-based or user-based. Host-based sessions are implemented by invoking the xhost command, which allows the user to add or remove hosts. User-based sessions are implemented by the xauth utility, which authorizes users who can access the remote X host using keys.

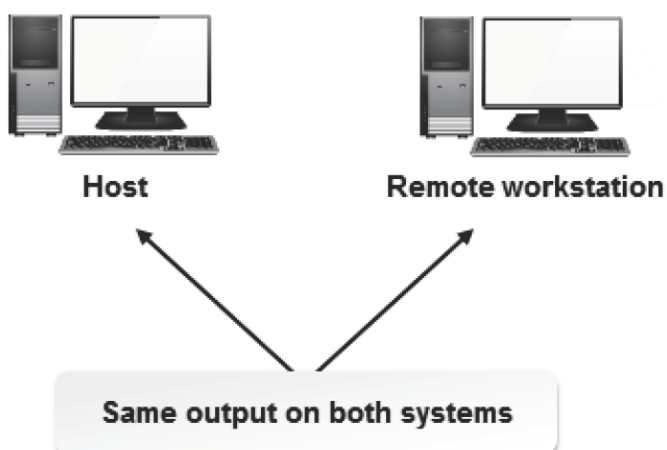


Figure 18-6: Remote X session over a network.

Commands Used in Remote X Sessions

The xhost and xauth commands are used to manage remote X sessions. The following options are provided for effective session management.

<i>Option</i>	<i>Enables You To</i>
xhost - help	Display a usage message.
xhost +{name}	Add a name to the list of hosts or users connecting to the X server.
xhost -{name}	Remove a name from the list of hosts or users connecting to the X server.
xauth -f {authfile}	Set the authority file to be used by xauth.
xauth -i	Let xauth bypass authority file locks.
xauth -v	Let xauth print status messages.

X-Stations

An **X-station** is a terminal or diskless workstation that is connected to a network and engineered to run the X Window system remotely. An X-system is not directly connected to a computer's CPU.

All X-station systems on a network are connected to a central workstation. The central workstation provides the terminals with the operating system, memory, programs, and CPU cycles.

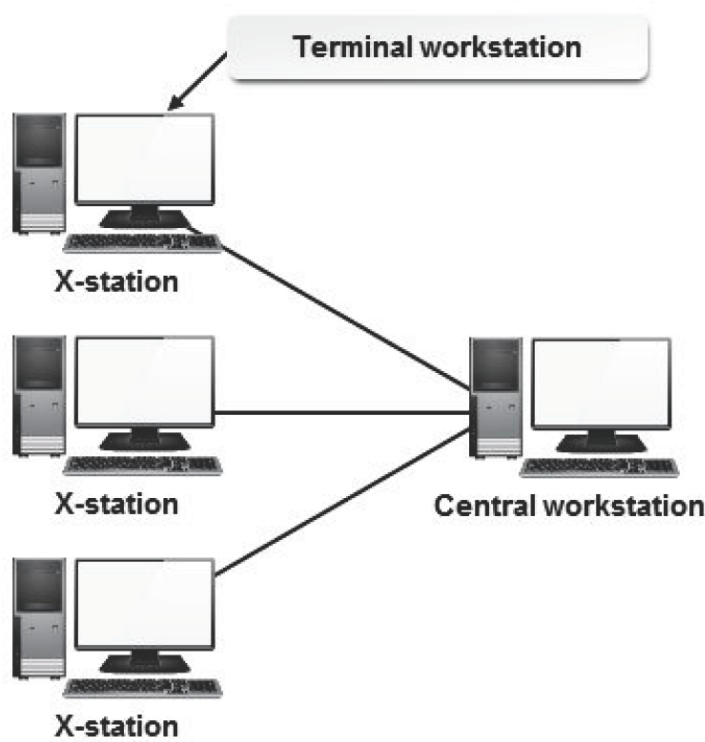


Figure 18-7: X-station systems connected to a central workstation on a network.

How to Implement X

Follow these general procedures to implement X.

Configure XOrg in Runlevel 3 (multi-user.target)

To configure XOrg in runlevel 3:

1. Log in to the CLI as **root**.
2. To boot in runlevel 3, enter `systemctl isolate multi-user.target`.
3. To start the X server from the command line, enter `startx` or `systemctl isolate graphical.target`.

Configure X to run at boot (graphical.target)

To configure the system to boot into the graphical target (runlevel 5):

1. Log in as **root**.
2. To boot into the graphical target (runlevel 5), enter `systemctl set-default graphical.target`.
3. Restart the computer.

Customize X for the Monitor Manually

To customize X for the monitor manually:

1. Log in to the CLI as **root**.
2. To generate a basic `xorg.conf` configuration file, enter `Xorg :1 -configure`.
3. To copy the generated `xorg.conf` configuration file to the system configuration folder, enter `cp /root/xorg.conf.new /etc/X11/xorg.conf`.
4. To navigate to the `/etc/X11` folder, enter `cd /etc/X11`.
5. To open the X configuration file, enter `vi xorg.conf`.
6. Below the **Section "Screen"** column, make necessary changes to the monitor settings.
7. Save and close the file.
8. Log out and log in for the changes to take effect.

Export X Sessions

To export X sessions:

1. Log in to the GUI.
2. To set the display variable, at the terminal, enter `export DISPLAY={client IP address}: 0.0`.



Note: DISPLAY is an environment variable that is used to specify where to export the X display.

3. To add the server to the list of hosts, enter `xhost +{server IP address}`.

Configure X Windows

To configure X Windows:

1. Log in to **GNOME**.
2. Select **Applications** → **System Tools** → **Settings** → **Displays**.

3. If necessary, adjust the **Display** settings.
 - Resolution
 - Rotation
4. To adjust the hardware settings, select the **Hardware** tab.
 - Monitor Type
 - Video Card
5. To adjust settings for multiple monitors, select the **Detect Displays** button and review the **Mirror displays** setting.

TOPIC B Customize the Display Manager

In the last topic, you implemented X to work with the Linux GUI. Now, you want to customize the GUI environment. In the GUI, the desktop is one of the first screens that a user interacts with.

Therefore, it is necessary that the desktop be appealing and easy to use. In this topic, you will customize the display manager to manage the desktop environment.

The desktop is an important part of any GUI. Users may want to customize their desktop environments according to their preferences. They can keep applications that they access frequently and shortcuts to different programs on the desktop. This will enable easy access to various applications and options.

Display Managers

A *display manager*, or *window manager*, is a program that controls the look and feel of a desktop environment. The display manager provides a graphical login screen and manages a collection of X servers. These servers may be on the local host or on remote systems. Display managers can be customized to run every time the system boots. The most popular desktop environments that are used by users are GNOME and KDE.

You can customize any of the applications present in the **Applications**, **Places**, or **System** folders for KDE and GNOME. After saving the settings, they will then be applied to the desktop environment. Most of the applications are common to both KDE and GNOME, while some are specific to the individual environment, such as **Control Center** in KDE.



Figure 18-8: Output of the display manager.

Display Managers for Linux

Common display managers for Linux are as follows:

- The **GNOME Display Manager (GDM)** is the default display manager for Red Hat Linux. GDM allows users to configure language settings, log in, shut down, or reboot the system.
- The **KDE Display Manager (KDM)** is the display manager for KDE, or K Desktop Environment. It allows users to log in, shut down, or reboot the system.
- The X Display Manager, or **xdm**, is a basic display manager that allows the user to log in, shut down, or reboot the system.

The GNOME Desktop Environment

The GNOME desktop environment (GDE) is the default desktop environment in most Linux distributions. The GNOME desktop initially displays three icons: **Computer**, **root's Home**, and **Trash**. There are two horizontal panels, one at the top and one at the bottom of the desktop. A user can customize these panels with shortcuts to applications that are frequently used. GDM is used to customize GDE.

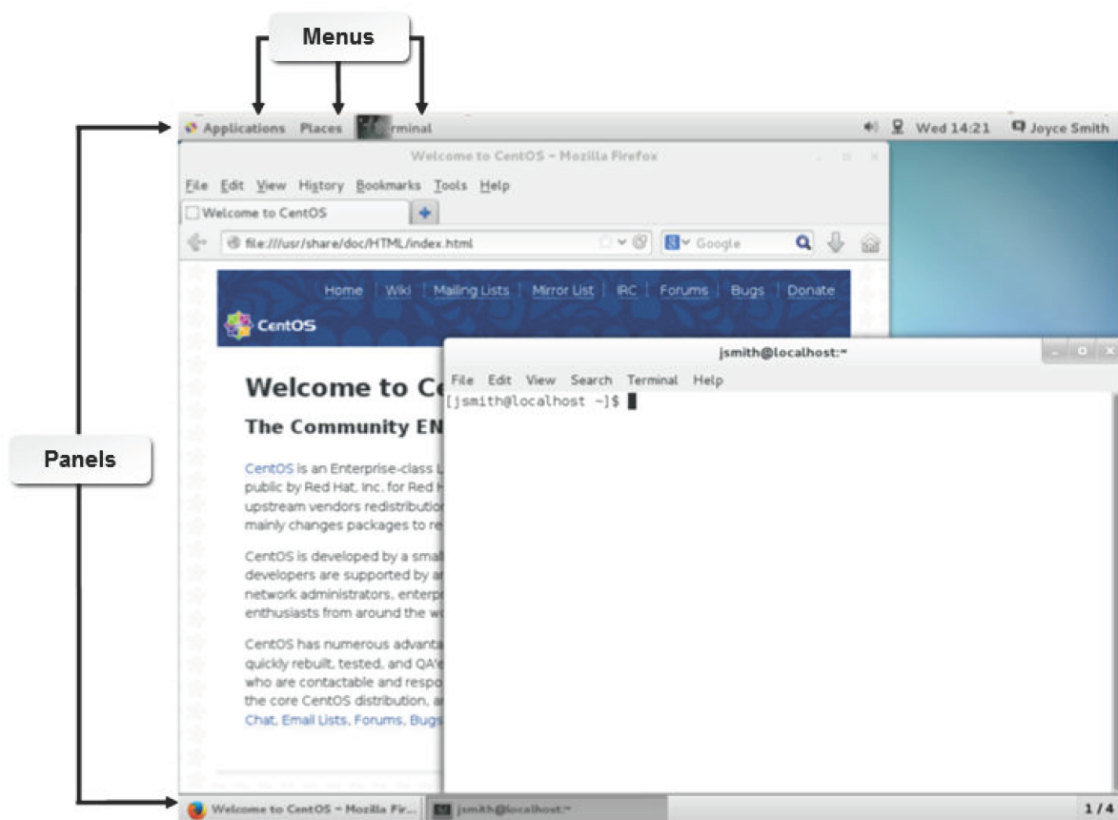


Figure 18-9: The GDM with its various components.

The KDE Desktop Environment

The KDE desktop environment is installed along with GDE in some distributions such as RHEL and CentOS. In KDE, there is only one horizontal panel at the bottom of the desktop. In CentOS, the main menu can be accessed by clicking the **CentOS** logo at the bottom-left corner of the KDE panel. KDE can be customized to suit users' needs.



Figure 18-10: The KDE display manager displaying only one panel.

Configure KDM

The **Default Desktop Settings** window, accessible by right-clicking the desktop, enables you to customize the appearance of KDM. The options that are available in the **Desktop Settings -Plasma Desktop Shell** window are provided in the following table.

<i>Configure - KDesktop Option</i>	<i>Used To</i>
View	Configure the background and other display settings.
Layout	Configure the number of virtual desktops. For example, you can specify a Newspaper Layout that puts widgets into columns in CentOS.
Wallpaper	Change the background settings such as wallpaper and background.
Mouse Actions	Change the effect that each button or scroll feature of your mouse has on the desktop environment.

KDE Panel Configuration Options

The **Add Panel** and **Add Widgets** options are used to access and configure different applications that are categorized under **Widget** and **Panel**. Some of the options on each menu are provided in the following table.

<i>Add Panel</i>	<i>Add Widget</i>
Default Panel	Activities
Empty Panel	Activity Bar
Grouping Panel	Analog Clock
	Application Launcher
	Battery Monitor
	Black Board
	Bookmarks
	Bouncy Ball
	Calendar
	Comic Strip

The LightDM Desktop Manager

The **LightDM** desktop manager is a lightweight alternative to the xdm and GDM display managers, and manages the displays on an X server system. LightDM was designed to be fast, flexible, and simple and is often paired with a similarly simple desktop environment on older machines. LightDM is the default window manager on Ubuntu, and is available for most other Linux distributions including RHEL/CentOS.

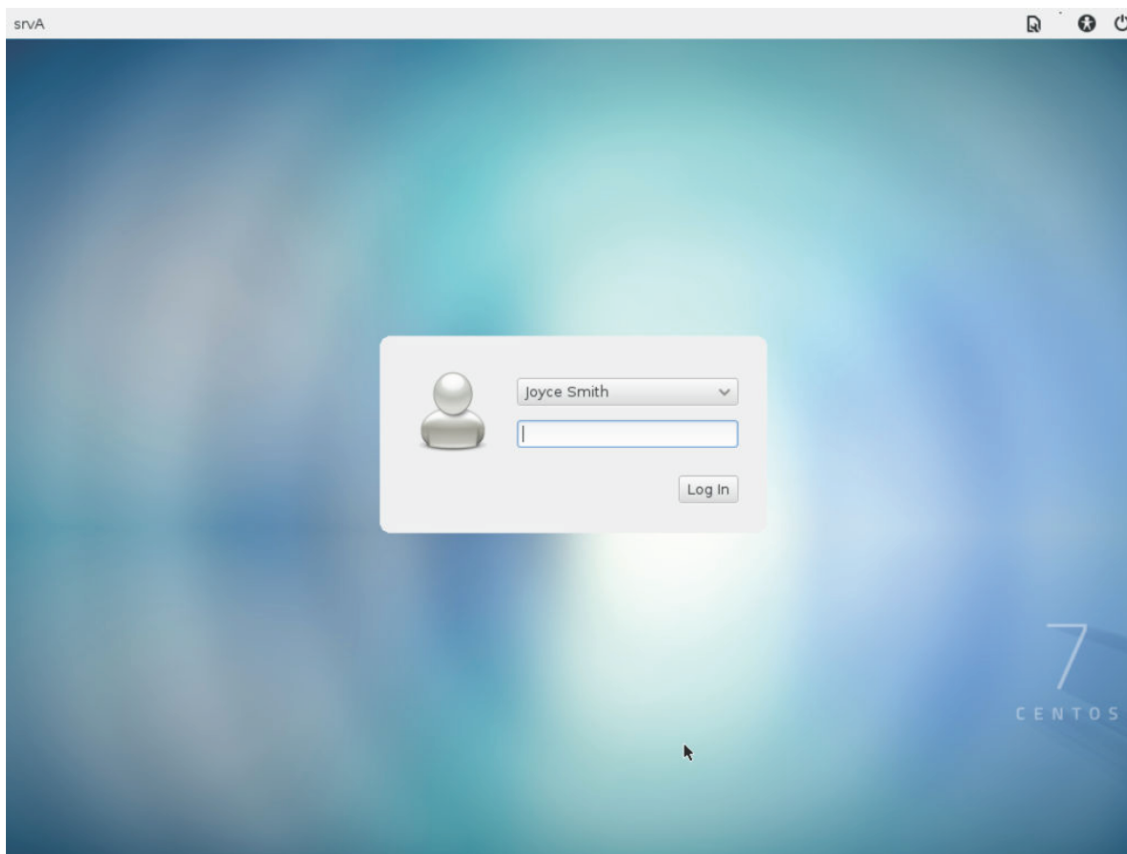


Figure 18-11: The LightDM display manager displaying the login screen.

How to Customize the Display Manager

Follow these general procedures to customize the display manager.

Switch Between Desktop Environments

To switch between desktop environments:

1. View the GNOME Login Screen.
2. Select a user and verify that the **Password** entry window is displayed.
3. To the left of the **Sign In** button, select the **Settings** icon.
4. Verify that a number of **Desktop Environment** options are displayed, including **KDE Plasma Workspace**, **GNOME**, and **GNOME Classic**.
5. Select **KDE Plasma Workspace**.
6. Enter the password for the user, and select **Sign In**.
7. Verify that the display environment is now KDE.

Configure KDE Desktop

To configure KDE Desktop:

1. View the GNOME Login Screen.
2. Select a user and verify that the **Password** entry window is displayed.
3. To the left of the **Sign In** button, select the **Settings** icon.

4. Verify that a number of **Desktop Environment** options are displayed, including **KDE Plasma Workspace**, **GNOME**, and **GNOME Classic**.
5. Select **KDE Plasma Workspace**.
6. Enter the password for the user, and select **Sign In**.
7. Verify that the display environment is now KDE.
8. On the KDM desktop, right-click and select **Default Desktop Settings**.
9. In the **Desktop Settings - Plasma Desktop Shell** dialog box, in the left pane, verify that **View** is selected.
10. In the right pane, in the **images** section, select an option from the thumbnails list.

Install the LightDM Display Manager

To install the LightDM Display Manager:

1. Log in as **root** in GUI.
2. If you have Internet access on your server:
 - a. To install the Extra Packages for Enterprise Linux (EPEL) Yum Repository, enter `yum install epel-release`.
 - b. To confirm installation, enter **Y** at the **Is this ok [y/d/N]** prompt.
 - c. To install LightDM, enter `yum install lightdm`.
 - d. To confirm installation, enter **Y** at the **Is this ok [y/d/N]** prompt.
 - e. To accept the EPEL Repository GPG key, enter **Y** at the **Is this ok [y/N]** prompt.
3. If you do not have Internet access on your server:
 - a. To install the local copy of LightDM via RPM files, enter `yum localinstall /opt/linuxplus/Data/Graphical_User_Interface/LightDM/lightdm*.rpm`.
4. To disable the GDM Display Manager, enter `systemctl disable gdm`.
5. To enable the LightDM Display Manager, enter `systemctl enable lightdm`.
6. To switch to runlevel 5 (graphical target), enter `systemctl isolate graphical.target`.
7. Verify that the login (greeter) window is now the LightDM window.
8. To verify the applied changes, reboot the server.

Change the Default Display Manager to LightDM

To change the default display manager to LightDM:

1. Log in as **root** in GUI.
2. To switch to runlevel 3 (multi-user target), enter `systemctl isolate multi-user.target`.
3. To disable the GDM Display Manager, enter `systemctl disable gdm`.

4. To enable the LightDM Display Manager, enter `systemctl enable lightdm`.
5. To switch to runlevel 5 (graphical target), enter `systemctl isolate graphical.target`.
6. Verify that the login (greeter) window is now the LightDM window.
7. To verify the applied changes, reboot the server.

Change the Background Wallpaper of LightDM Login Screen

To change the background wallpaper of LightDM login screen:

1. Log in as **root** in CLI mode.
2. Enter `cd /etc/lightdm`.
3. Enter `vi lightdm-gtk-greeter.conf`.



Note: *lightdm-gtk-greeter.conf* is a global configuration file that is used to customize the LightDM Display Manager Greeter (Login) display.

4. To customize the wallpaper, change `background=/usr/share/backgrounds/day.jpg` to `background=/usr/share/backgrounds/night.jpg`.



Note: You can also specify a hexadecimal color code (i.e., `#000000` for Black) to change the background to a solid color, or specify a path to an alternative image for display.

5. Save and close the file.
6. To start the LightDM Display Manager and verify the changes made, enter `lightdm`.
7. Verify that the new login screen background is the *night.jpg* image (a darker, blue design).

View the LightDM Logs

To view the LightDM logs:

1. Log in as **root** in GUI mode.
2. Enter `cd /var/log/lightdm`.
3. If necessary, to list all the logs in the *lightdm* folder, enter `ls`.
4. To view the 10 most recent entries from the LightDM log, enter `tail lightdm.log`.
5. Verify that no errors are visible.

TOPIC C Enable Accessibility Settings in Linux

In the previous topic, you customized the window environment. Sometimes, you may be required to modify the computer operating system environment to be usable to users who may have physical or visual disabilities. In this topic, you will enable accessibility settings in Linux.

As a Linux administrator, you need to ensure that all users can use their system with ease.


Sometimes, users may have a physical or visual disability, which may prevent them from using the default settings available on the system. In such cases, you may need to enable additional keyboard options, sound options, or display themes, so that the users with such disabilities can use their systems with relative ease.

Accessibility Options

Linux provides accessibility options for users to customize Linux and accomplish tasks despite physical and environmental challenges. You need to enable Assistive Technologies (ATs) that support three features, namely Screen reader, Magnifier, and On Screen Keyboard. Enabling these features before logging on to Linux helps users optimize the Linux environment to suit their needs.

The following table describes the three accessibility options.


<i>Option</i>	<i>Description</i>
Screen reader	Reads all the highlighted screen elements.
Magnifier	Magnifies the highlighted section on the screen.
On Screen Keyboard	Enables the On Screen Keyboard so that you can click to type text using the mouse, instead of the regular physical keyboard.



Note: Orca is a screen reader in Linux.

The GOK - main Window

After you enable the assistive technology option, when you re-login to the system, a new window named GOK-main window will be displayed. GOK stands for GNOME On Screen Keyboard, a virtual keyboard that can be operated using the mouse. This window contains options that enable users to navigate around the GUI elements with ease.



Note: In GNOME 3, the version of GNOME included with RHEL/CentOS 7, the Caribou engine has replaced GNOME 2's GOK for the display and management of an onscreen keyboard.

Gestures at gdm Login

Modern GNOME systems support multitouch gestures on touch-enabled touchscreens, enabling accessibility at login to log in to the system.

Keyboard Accessibility Options

Keyboard accessibility options (AccessX) enable additional features that allow users to handle the keys of the keyboard and the mouse in an easier manner.

<i>Keyboard Accessibility Option</i>	<i>Enables Features To</i>
Sticky keys	Allow users to press one key on the keyboard instead of a combination of several keys at once.
Repeat keys	Recognize the same keyboard key pressed multiple times by using the time specified for the delay and the number of characters per second specified as the speed.
Slow keys	Accept only the keystrokes of keys that are held for a specific duration.
Bounce keys	Avoid unintended successive strokes of a specific key on the keyboard.

Keyboard Accessibility Option	Enables Features To
Toggle keys	Enable system beep when toggle keys, such as Num Lock , Caps Lock , or Scroll Lock that illuminate LEDs on the keyboard, are pressed.
Mouse keys	Allow users to set maximum pointer speed, time to accelerate to maximum speed, and the delay between mouse keypress and pointer movement.

Accessibility Based Themes

Linux contains specific themes that are meant for improving the accessibility of users with visibility problems.

Theme	Displays
High Contrast	The screen using black on white text and black on white icons.
High Contrast Inverse	The screen using white on black text and white on black icons.
High Contrast Large Print	The screen similar to the High Contrast theme with larger text and icons.
High Contrast Large Print Inverse	The screen similar to the High Contrast Inverse theme with larger text and icons.

Orca

Orca is a built-in screen reader in GNOME. A screen reader is software that describes the screen layout and content for users that are visually impaired or learning disabled. Orca allows users to modify settings in the **Orca Preferences** dialog box according to their requirements. Each setting can be enabled by using the check boxes on the tabs of the **Orca Preferences** dialog box.

Tab	Allows You To
Speech	Select either Emacspeak Speech Services or GNOME Speech Services as the speech system. You can select the desired settings for Speech synthesizer , Voice settings , and Person . In addition, you can set the desired values for Rate , Pitch , and Volume . You can also set the Punctuation Level , Verbosity , and Table Row Speech options. Finally, you can enable speak indentation and justification.
Braille	Enable Braille support, Braille monitor, and Abbreviated role names to enable support for Braille display and specify the desired setting. You can select either Brief or Verbose as the Verbosity option.
Key Echo	Enable key echo for alphanumeric and punctuation keys, modifier keys, locking keys, function keys, and action keys. You can also enable key echo by word.
Magnifier	Select the desired cursor and crosshair settings. You can set the values for Zoomer Settings and Zoomer Position . In addition, you can choose the desired settings for Smoothing and Mouse tracking mode .

How to Configure Accessibility Settings

Follow these general procedures to configure accessibility settings.

Enable Assistive Technology

To enable assistive technology:

1. Log in as **root** in the GUI.
2. From the panel, select **Applications** → **System Tools** → **Settings** → **Universal Access**.
3. To enable assistive technology, in the **Universal Access** dialog box, switch the desired options on from among the **Seeing**, **Hearing**, **Typing**, and **Pointing and Clicking** tabs.
4. To display the **Orca** application, enter orca in a Terminal window.
5. To display the **Orca Preferences** dialog box, select **Preferences**.

6. In the dialog box, select the desired tab to enable the settings related to **General**, **Voice**, **Speech**, **Braille**, **Key Echo**, and **Key Bindings**.
7. To enable assistive technology during login, select **Close and Log Out**.

Use On Screen Keyboard Assistive Technology

To use On Screen Keyboard assistive technology:

1. Log in as **root** in the GUI.
2. From the panel, select **Applications** → **System Tools** → **Settings** → **Universal Access**.
3. In the **Universal Access** dialog box, in the **Typing** tab, switch the **On Screen Keyboard** option to **ON**.
4. To enable the On Screen Keyboard, close the dialog box.
5. From the panel, select **Applications** → **Accessories** → **gedit**.
6. To display the On Screen Keyboard, in the **Unsaved Document 1- gedit** window, select the text area.
7. To input data to the system, select the desired buttons on the On Screen Keyboard.

Enable Keyboard Accessibility Features

To enable keyboard accessibility features:

1. Log in as **root** in the GUI.
2. From the panel, select **Applications** → **System Tools** → **Settings** → **Universal Access**.
3. On the **Typing** tab, if necessary, ensure the **Stick Keys**, **Slow Keys**, and **Bounce Keys** settings are set to **ON**.
4. Select the **Pointing and Clicking** tab. Set the **Mouse Keys** setting to **ON**. If necessary, select **Mouse Settings** and specify the desired setting.
5. To save, close the dialog box.

Enable Accessibility at GDM Login

To enable accessibility at GDM login.

1. Log in as **root** in the GUI.
2. From the panel, select **Applications** → **System Tools** → **Settings** → **Universal Access**.
3. In the **Universal Access** dialog box, in the **Typing** tab, switch the **On Screen Keyboard** option to **ON** to enable the On Screen Keyboard.
4. Log out of the system.
5. If necessary, select the **Universal Access** icon in the top menu bar to enable additional accessibility settings.

Configuring the GUI Review

Scenario

Answer the following review questions.

1. Do you think using the Linux GUI in conjunction with the CLI will yield better results? Why?
2. In what way do you think customizing window managers is useful?

Summary

In this lesson, you configured the GUI. Working with the GUI of Linux can be useful when recalling commands becomes difficult. The GUI is user-friendly and easy to understand. As a Linux administrator, it will help you direct users to configure their systems.

A Taking the Exams

When you think you have learned and practiced the material sufficiently, you can book a time to take the test.

Preparing for the Exam

We've tried to balance this course to reflect the percentages in the exam so that you have learned the appropriate level of detail about each topic to comfortably answer the exam questions. Read the following notes to find out what you need to do to register for the exam and get some tips on what to expect during the exam and how to prepare for it.

Questions in the exam are weighted by domain area as follows:

CompTIA Linux+® Powered by LPI LX0-103 Certification Domain Areas	Weighting
1.0 System Architecture	14%
2.0 Linux Installation and Package Management	18%
3.0 GNU and Unix Commands	43%
4.0 Devices, Linux Filesystems, and Filesystem Hierarchy Standard	25%

CompTIA Linux+® Powered by LPI LX0-104 Certification Domain Areas	Weighting
1.0 Shells, Scripting and Data Management	17%
2.0 User Interfaces and Desktops	8%
3.0 Administrative Tasks	20%
4.0 Essential System Services	17%
5.0 Networking Fundamentals	23%
6.0 Security	15%

Registering for and Taking the Exam

CompTIA Certification exams are delivered exclusively by Pearson VUE.

- Log on to **Pearson VUE** and register your details to create an account.
- To book a test, log in using your account credentials then click the link to schedule an appointment.
- The testing program is CompTIA and the exam code is **LX0-103** or **LX0-104**.
- Use the search tool to locate the test center nearest you, then book an appointment.
- If you have purchased a voucher or been supplied with one already, enter the voucher number to pay for the exam. Otherwise, you can pay with a credit card.
- When you have confirmed payment, an email will be sent to the account used to register, confirming the appointment and directions to the venue. Print a copy and bring it with you when you go to take your test.

When You Arrive at the Exam

On the day of the exam, note the following:

- Arrive at the test center at least **15 minutes before the test** is scheduled.
- You must have **two forms of ID**; one with picture, one preferably with your private address, and both with signature. View CompTIA's candidate ID policy for more information on acceptable forms of ID.



Note: See the candidate ID policy at <https://certification.comptia.org/testing/test-policies/candidate-id-policy>.

- Books, calculators, laptops, cellphones, smartphones, tablets, or other reference materials are not allowed in the exam room.
- You will be given note taking materials, but you must not attempt to write down questions or remove anything from the exam room.
- It is CompTIA's policy to make reasonable accommodations for individuals with disabilities.
- The test center administrator will demonstrate how to use the computer-based test system and wish you good luck. Check that your name is displayed, read the introductory note, and then click the button to start the exam.

Taking the Exam

CompTIA has prepared a **Candidate Experience video**. Watch this to help to familiarize yourself with the exam format and types of questions.



Note: The Candidate Experience video is available at <https://www.youtube.com/embed/kyTdN2GZiZ8>.

- There are up to 90 multiple-choice questions and **performance-based items**, which must be answered in 165 minutes. The exam is pass/fail only with no scaled score.
- Read each question and its option answers carefully. Don't rush through the exam as you'll probably have more time at the end than you expect.
- At the other end of the scale, don't get "stuck" on a question and start to panic. You can mark questions for review and come back to them.
- As the exam tests your ability to recall facts and to apply them sensibly in a troubleshooting scenario, there will be questions where you cannot recall the correct answer from memory.

Adopt the following strategy for dealing with these questions:

- Narrow your choices down by eliminating obviously wrong answers.
- Don't guess too soon! You must select not only a correct answer, but the best answer. It is therefore important that you read all of the options and not stop when you find an option that is correct. It may be impractical compared to another answer.
- Utilize information and insights that you've acquired in working through the entire test to go back and answer earlier items that you weren't sure of.
- Think your answer is wrong - should you change it? Studies indicate that when students change their answers they usually change them to the wrong answer. If you were fairly certain you were correct the first time, leave the answer as it is.
- As well as multiple-choice questions, there will be a number of performance-based items.

Performance-based items require you to complete a task or solve a problem in simulated IT environments. Make sure you read the item scenario carefully and check your submission.

- The performance items are usually positioned at the start of the exam, but it is not required that you complete them first. You may consider completing the multiple-choice items first and returning to the performance items.
- Don't leave any questions unanswered! If you really don't know the answer, just guess.
- The exam may contain "unscored" questions, which may even be outside the exam objectives.

These questions do not count toward your score. Do not allow them to distract or worry you.

- The exam questions come from a regularly updated pool to deter cheating. Do not be surprised if the questions you get are quite different to someone else's experience.



Caution: Do not discuss the contents of the exam or attempt to reveal specific exam questions to anyone else. By taking the exam, you are bound by CompTIA's confidentiality agreement.

After the Exam

Note the following after taking the exam:

- A score report will be generated immediately, and a copy will be printed for you by the test administrator.
- The score report will show whether you have passed or failed and your score in each section.

Make sure you retain the report!

- If you passed your CompTIA exam, your score report will provide you with instructions on creating an account with the Certmetrics candidate database for viewing records, ordering duplicate certificates, or downloading certification logos in various file formats. You will also be sent an email containing this information. If you failed your CompTIA exam, you'll be provided with instructions for retaking the exam.
- Newly-certified individuals will receive a physical certificate by mail. If six weeks have passed after taking your exam and you haven't received a copy of your certificate, contact CompTIA support.

Retaking the Exam and Additional Study

If you fail the first attempt of your certification, you can retake it at your convenience. However, before your third attempt or any subsequent attempt to pass such examination, you are required to wait a certain amount of time since your last attempt. Review your score report to understand how long before you can attempt again. Note that you will have to pay the exam price each time you attempt.

B Mapping Course Content to the CompTIA® Linux +® Powered by LPI (Exams LX0-103 and LX0-104) Exam Objectives

Obtaining CompTIA Linux+ Powered by LPI certification requires candidates to pass exams LX0-103 and LX0-104. This table describes where the objectives for exams LX0-103 and LX0-104 are covered in this course.

101 System Architecture

Exam Objective (LX0-103)	Course Lesson and Topic Reference
101.1 Determine and Configure hardware settings	
<ul style="list-style-type: none">• Enable and disable integrated peripherals	Lesson 17, Topic A
<ul style="list-style-type: none">• Configure systems with or without external peripherals such as keyboards	Lesson 15, Topic B
<ul style="list-style-type: none">• Differentiate between the various types of mass storage devices	Lesson 3, Topic D Lesson 15, Topic A
<ul style="list-style-type: none">• Know the differences between coldplug and hotplug devices	Lesson 8, Topic D
<ul style="list-style-type: none">• Determine hardware resources for devices	Lesson 8, Topic E Lesson 15, Topic A
<ul style="list-style-type: none">• Tools and utilities to list various hardware information (e.g. lsusb, lspci, etc.)	Lesson 8, Topic D Lesson 16, Topic B

Exam Objective (LX0-103)	Course Lesson and Topic Reference
101.1 Determine and Configure hardware settings	
<ul style="list-style-type: none">• Tools and utilities to manipulate USB devices	Lesson 8, Topic D Lesson 16, Topic B
<ul style="list-style-type: none">• Conceptual understanding of sysfs, udev, dbus	Lesson 8, Topic D Lesson 8, Topic E
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none">• /sys	Lesson 8, Topic D
<ul style="list-style-type: none">• /proc	Lesson 8, Topic B
<ul style="list-style-type: none">• /dev	Lesson 3, Topic B
<ul style="list-style-type: none">• modprobe	Lesson 8, Topic B
<ul style="list-style-type: none">• lsmod	Lesson 8, Topic B
<ul style="list-style-type: none">• lspci	Lesson 8, Topic E
<ul style="list-style-type: none">• lsusb	Lesson 8, Topic E

Exam Objective (LX0-103)	Course Lesson and Topic Reference
101.2 Boot the System	
<ul style="list-style-type: none"> Provide common commands to the boot loader and options to the kernel at boot time 	Lesson 8, Topic B Lesson 17, Topic B
<ul style="list-style-type: none"> Demonstrate knowledge of the boot sequence from BIOS to boot completion 	Lesson 17, Topic B
<ul style="list-style-type: none"> Understanding of SysVinit and systemd 	Lesson 1, Topic D
<ul style="list-style-type: none"> Awareness of Upstart 	Lesson 1, Topic D
<ul style="list-style-type: none"> Check boot events in the log file 	Lesson 16, Topic A
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> dmesg 	Lesson 16, Topic B
<ul style="list-style-type: none"> BIOS 	Lesson 17, Topic A Lesson 17, Topic B
<ul style="list-style-type: none"> bootloader 	Lesson 17, Topic B
<ul style="list-style-type: none"> kernel 	Lesson 8, Topic A
<ul style="list-style-type: none"> initramfs 	Lesson 1, Topic D
<ul style="list-style-type: none"> init 	Lesson 1, Topic D
<ul style="list-style-type: none"> SysVinit 	Lesson 1, Topic D
<ul style="list-style-type: none"> systemd 	Lesson 11, Topic A

Exam Objective (LX0-103)	Course Lesson and Topic Reference
101.3 Change runlevels / boot targets and shutdown or reboot system	
<ul style="list-style-type: none"> Set the default runlevel or boot target 	Lesson 1, Topic D
<ul style="list-style-type: none"> Change between runlevels/boot targets including single user mode 	Lesson 1, Topic D
<ul style="list-style-type: none"> Shutdown and reboot from the command line 	Lesson 1, Topic D
<ul style="list-style-type: none"> Alert users before switching runlevels/boot targets or other major system events 	Lesson 1, Topic D
<ul style="list-style-type: none"> Properly terminate processes 	Lesson 10, Topic B
The following is a partial list of the used files, terms and utilities:	

Exam Objective (LX0-103)	Course Lesson and Topic Reference
101.3 Change runlevels / boot targets and shutdown or reboot system	
<ul style="list-style-type: none"> • /etc/inittab 	Lesson 1, Topic D
	Lesson 11, Topic A
<ul style="list-style-type: none"> • shutdown 	Lesson 1, Topic D
<ul style="list-style-type: none"> • init 	Lesson 1, Topic D
<ul style="list-style-type: none"> • /etc/init.d 	Lesson 11, Topic A
<ul style="list-style-type: none"> • telinit 	Lesson 1, Topic D
<ul style="list-style-type: none"> • systemd 	Lesson 11, Topic A
<ul style="list-style-type: none"> • systemctl 	Lesson 11, Topic A
<ul style="list-style-type: none"> • /etc/systemd/ 	Lesson 16, Topic C
<ul style="list-style-type: none"> • /usr/lib/systemd/ 	Lesson 13, Topic B
<ul style="list-style-type: none"> • wall 	Lesson 1, Topic B

102 Linux Installation and Package Management

Exam Objective (LX0-103)	Course Lesson and Topic Reference
102.1 Design hard disk layout	
<ul style="list-style-type: none"> • Allocate filesystems and swap space to 	Lesson 3, Topic A
separate partitions or disks	Lesson 3, Topic C
<ul style="list-style-type: none"> • Tailor the design to the intended use of the system 	Lesson 3, Topic C
<ul style="list-style-type: none"> • Ensure the /boot partition conforms to the hardware architecture requirements for booting 	Lesson 3, Topic A Lesson 3, Topic B Lesson 16, Topic A
	Lesson 17, Topic C
<ul style="list-style-type: none"> • Knowledge of basic features of LVM 	Lesson 17, Topic A
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> • / (root) filesystem 	Lesson 3, Topic B
<ul style="list-style-type: none"> • /var filesystem 	Lesson 3, Topic B
<ul style="list-style-type: none"> • /home filesystem 	Lesson 3, Topic B

Exam Objective (LX0-103)	Course Lesson and Topic Reference
102.1 Design hard disk layout	
<ul style="list-style-type: none"> • /boot filesystem 	Lesson 3, Topic B
<ul style="list-style-type: none"> • swap space 	Lesson 3, Topic C
<ul style="list-style-type: none"> • mount points 	Lesson 3, Topic C
<ul style="list-style-type: none"> • partitions 	Lesson 3, Topic A

Exam Objective (LX0-103)	Course Lesson and Topic Reference
102.2 Install a boot manager	
<ul style="list-style-type: none"> • Providing alternative boot locations and backup boot options 	Lesson 17, Topic C
<ul style="list-style-type: none"> • Install and configure a boot loader such as GRUB Legacy 	Lesson 17, Topic C
<ul style="list-style-type: none"> • Perform basic configuration changes for 	Lesson 17, Topic B
GRUB 2	Lesson 17, Topic C
<ul style="list-style-type: none"> • Interact with the boot loader 	Lesson 17, Topic C
The following is a partial list of the used files, terms, and utilities	
<ul style="list-style-type: none"> • menu.lst, grub.cfg, and grub.conf 	Lesson 17, Topic C
<ul style="list-style-type: none"> • grub-install 	Lesson 17, Topic C
<ul style="list-style-type: none"> • grub-mkconfig 	Lesson 17, Topic C
<ul style="list-style-type: none"> • MBR 	Lesson 17, Topic B

Exam Objective (LX0-103)	Course Lesson and Topic Reference
102.3 Manage shared libraries	
<ul style="list-style-type: none"> • Identify shared libraries 	Lesson 7, Topic F
<ul style="list-style-type: none"> • Identify the typical locations of system libraries 	Lesson 3, Topic B Lesson 7, Topic F
<ul style="list-style-type: none"> • Load shared libraries 	Lesson 7, Topic F
The following is a partial list of the used files, terms and utilities	
<ul style="list-style-type: none"> • ldd 	Lesson 7, Topic F
<ul style="list-style-type: none"> • ldconfig 	Lesson 7, Topic F

Exam Objective (LX0-103)	Course Lesson and Topic Reference
102.3 Manage shared libraries	
<ul style="list-style-type: none"> • <code>/etc/ld.so.conf</code> 	Lesson 7, Topic F
<ul style="list-style-type: none"> • <code>LD_LIBRARY_PATH</code> 	Lesson 7, Topic F

Exam Objective (LX0-103)	Course Lesson and Topic Reference
102.4 Use Debian package management	
<ul style="list-style-type: none"> • Install, upgrade and uninstall Debian binary packages 	Lesson 7, Topic F
<ul style="list-style-type: none"> • Find packages containing specific files or libraries which may or may not be installed 	Lesson 7, Topic F
<ul style="list-style-type: none"> • Obtain package information like version, content, dependencies, package integrity and installation status (whether or not the package is installed) 	Lesson 7, Topic F
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> • <code>/etc/apt/sources.list</code> 	Lesson 7, Topic F
<ul style="list-style-type: none"> • <code>dpkg</code> 	Lesson 7, Topic F
<ul style="list-style-type: none"> • <code>dpkg-reconfigure</code> 	Lesson 7, Topic F
<ul style="list-style-type: none"> • <code>apt-get</code> 	Lesson 7, Topic F
<ul style="list-style-type: none"> • <code>apt-cache</code> 	Lesson 7, Topic F
<ul style="list-style-type: none"> • <code>aptitude</code> 	Lesson 7, Topic F

Exam Objective (LX0-103)	Course Lesson and Topic Reference
102.5 Use RPM and YUM package management	
<ul style="list-style-type: none"> • Install, re-install, upgrade and remove packages using RPM and YUM 	Lesson 7, Topic A Lesson 7, Topic E
<ul style="list-style-type: none"> • Obtain information on RPM packages such as version, status, dependencies, integrity and signatures 	Lesson 7, Topic A
<ul style="list-style-type: none"> • Determine what files a package provides, as well as find which package a specific file comes from 	Lesson 7, Topic A
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> • <code>rpm</code> 	Lesson 7, Topic A

Exam Objective (LX0-103)	Course Lesson and Topic Reference
102.5 Use RPM and YUM package management	
<ul style="list-style-type: none"> rpm2cpio 	Lesson 7, Topic A
<ul style="list-style-type: none"> /etc/yum.conf 	Lesson 7, Topic D
<ul style="list-style-type: none"> /etc/yum.repos.d/ 	Lesson 7, Topic D
<ul style="list-style-type: none"> yum 	Lesson 7, Topic E
<ul style="list-style-type: none"> yumdownloader 	Lesson 7, Topic E

103 GNU and UNIX Commands

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.1 Work on the command line	
<ul style="list-style-type: none"> Use single shell commands and one line command sequences to perform basic tasks on the command line 	Lesson 1, Topic B
<ul style="list-style-type: none"> Use and modify the shell environment including defining, referencing and exporting environment variables 	Lesson 9, Topic C
<ul style="list-style-type: none"> Use and edit command history 	Lesson 1, Topic B Lesson 9, Topic A Lesson 9, Topic C
<ul style="list-style-type: none"> Invoke commands inside and outside the defined path 	Lesson 1, Topic B
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> bash 	Lesson 1, Topic B
<ul style="list-style-type: none"> echo 	Lesson 1, Topic B
<ul style="list-style-type: none"> env 	Lesson 2, Topic B
<ul style="list-style-type: none"> export 	Lesson 9, Topic C
<ul style="list-style-type: none"> pwd 	Lesson 3, Topic B
<ul style="list-style-type: none"> set 	Lesson 2, Topic B
<ul style="list-style-type: none"> unset 	Lesson 2, Topic B
<ul style="list-style-type: none"> man 	Lesson 1, Topic C
<ul style="list-style-type: none"> uname 	Lesson 8, Topic A Lesson 8, Topic E

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.1 Work on the command line	
<ul style="list-style-type: none"> history 	Lesson 9, Topic A
<ul style="list-style-type: none"> .bash_history 	Lesson 9, Topic A

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.2 Process text streams using filters	
<ul style="list-style-type: none"> Send text files and output streams through text utility filters to modify the output using standard UNIX commands found in the GNU textutils package 	Lesson 4, Topic D
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> cat 	Lesson 4, Topic D
<ul style="list-style-type: none"> cut 	Lesson 4, Topic D
<ul style="list-style-type: none"> expand 	Lesson 4, Topic D
<ul style="list-style-type: none"> fmt 	Lesson 4, Topic D
<ul style="list-style-type: none"> head 	Lesson 4, Topic D
<ul style="list-style-type: none"> od 	Lesson 4, Topic D
<ul style="list-style-type: none"> join 	Lesson 4, Topic D
<ul style="list-style-type: none"> nl 	Lesson 4, Topic D
<ul style="list-style-type: none"> paste 	Lesson 4, Topic D
<ul style="list-style-type: none"> pr 	Lesson 4, Topic D
<ul style="list-style-type: none"> sed 	Lesson 11, Topic B
<ul style="list-style-type: none"> sort 	Lesson 4, Topic D
<ul style="list-style-type: none"> split 	Lesson 4, Topic D
<ul style="list-style-type: none"> tail 	Lesson 4, Topic D
<ul style="list-style-type: none"> tr 	Lesson 4, Topic D
<ul style="list-style-type: none"> unexpand 	Lesson 4, Topic D

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.2 Process text streams using filters	
<ul style="list-style-type: none"> • <code>uniq</code> 	Lesson 4, Topic D
<ul style="list-style-type: none"> • <code>wc</code> 	Lesson 4, Topic D

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.3 Perform basic file management	
<ul style="list-style-type: none"> • Copy, move and remove files and directories individually 	Lesson 3, Topic B
<ul style="list-style-type: none"> • Copy multiple files and directories recursively 	Lesson 3, Topic B
<ul style="list-style-type: none"> • Remove files and directories recursively 	Lesson 3, Topic B
<ul style="list-style-type: none"> • Use simple and advanced wildcard specifications in commands 	Lesson 9, Topic A
<ul style="list-style-type: none"> • Using <code>find</code> to locate and act on files based on type, size, or time 	Lesson 4, Topic B Lesson 9, Topic A
<ul style="list-style-type: none"> • Usage of <code>tar</code>, <code>cpio</code>, and <code>dd</code> 	Lesson 4, Topic F
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> • <code>cp</code> 	Lesson 3, Topic B
<ul style="list-style-type: none"> • <code>find</code> 	Lesson 4, Topic B
<ul style="list-style-type: none"> • <code>mkdir</code> 	Lesson 3, Topic B

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.3 Perform basic file management	
<ul style="list-style-type: none"> • <code>mv</code> 	Lesson 3, Topic B
<ul style="list-style-type: none"> • <code>ls</code> 	Lesson 3, Topic B
<ul style="list-style-type: none"> • <code>rm</code> 	Lesson 3, Topic B
<ul style="list-style-type: none"> • <code>rmdir</code> 	Lesson 3, Topic B
<ul style="list-style-type: none"> • <code>touch</code> 	Lesson 3, Topic B
<ul style="list-style-type: none"> • <code>tar</code> 	Lesson 4, Topic F
<ul style="list-style-type: none"> • <code>cpio</code> 	Lesson 4, Topic F

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.3 Perform basic file management	
<ul style="list-style-type: none"> • dd 	Lesson 4, Topic F
<ul style="list-style-type: none"> • file 	Lesson 3, Topic B
<ul style="list-style-type: none"> • gzip 	Lesson 4, Topic F
<ul style="list-style-type: none"> • gunzip 	Lesson 4, Topic F
<ul style="list-style-type: none"> • bzip2 	Lesson 4, Topic F
<ul style="list-style-type: none"> • xz 	Lesson 4, Topic F
<ul style="list-style-type: none"> • file globbing 	Lesson 9, Topic A

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.4 Use streams, pipes and redirects	
<ul style="list-style-type: none"> • Redirecting standard input, standard output and standard error 	Lesson 9, Topic D
<ul style="list-style-type: none"> • Pipe the output of one command to the input of another command 	Lesson 9, Topic D
<ul style="list-style-type: none"> • Use the output of one command as arguments to another command 	Lesson 9, Topic D
<ul style="list-style-type: none"> • Send output to both stdout and a file 	Lesson 9, Topic D
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> • tee 	Lesson 9, Topic D
<ul style="list-style-type: none"> • xargs 	Lesson 9, Topic D

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.5 Create, monitor and kill processes	
<ul style="list-style-type: none"> • Run jobs in the foreground and background 	Lesson 10, Topic A
<ul style="list-style-type: none"> • Signal a program to continue running after logout 	Lesson 10, Topic C
<ul style="list-style-type: none"> • Monitor active processes 	Lesson 10, Topic B
<ul style="list-style-type: none"> • Select and sort processes for display 	Lesson 10, Topic B

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.5 Create, monitor and kill processes	
<ul style="list-style-type: none"> Send signals to processes 	Lesson 10, Topic B
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> & 	Lesson 10, Topic A
<ul style="list-style-type: none"> bg 	Lesson 10, Topic A
<ul style="list-style-type: none"> fg 	Lesson 10, Topic A
<ul style="list-style-type: none"> jobs 	Lesson 10, Topic A
<ul style="list-style-type: none"> kill 	Lesson 10, Topic B
<ul style="list-style-type: none"> nohup 	Lesson 10, Topic C
<ul style="list-style-type: none"> ps 	Lesson 10, Topic B
<ul style="list-style-type: none"> top 	Lesson 10, Topic B
<ul style="list-style-type: none"> free 	Lesson 8, Topic E
<ul style="list-style-type: none"> uptime 	Lesson 1, Topic B
<ul style="list-style-type: none"> pgrep 	Lesson 16, Topic B
<ul style="list-style-type: none"> pkill 	Lesson 16, Topic B
<ul style="list-style-type: none"> killall 	Lesson 10, Topic B
<ul style="list-style-type: none"> screen 	Lesson 10, Topic C

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.6 Modify process execution priorities	
<ul style="list-style-type: none"> Know the default priority of a job that is created 	Lesson 10, Topic B
<ul style="list-style-type: none"> Run a program with higher or lower priority than the default 	Lesson 10, Topic B
<ul style="list-style-type: none"> Change the priority of a running process 	Lesson 10, Topic B
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> nice 	Lesson 10, Topic B

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.6 Modify process execution priorities	
<ul style="list-style-type: none"> ps 	Lesson 10, Topic B
<ul style="list-style-type: none"> renice 	Lesson 10, Topic B
<ul style="list-style-type: none"> top 	Lesson 10, Topic B

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.7 Search text files using regular expressions	
<ul style="list-style-type: none"> Create simple regular expressions containing several notational elements 	Lesson 4, Topic C
<ul style="list-style-type: none"> Use regular expression tools to perform searches through a filesystem or file content 	Lesson 4, Topic C
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> grep 	Lesson 4, Topic B Lesson 4, Topic D Lesson 11, Topic B
<ul style="list-style-type: none"> egrep 	Lesson 11, Topic B
<ul style="list-style-type: none"> fgrep 	Lesson 11, Topic B
<ul style="list-style-type: none"> sed 	Lesson 11, Topic B
<ul style="list-style-type: none"> regex(7) 	Lesson 4, Topic C Lesson 11, Topic B

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.8 Perform basic file editing operations using vi	
<ul style="list-style-type: none"> Navigate a document using vi 	Lesson 4, Topic A
<ul style="list-style-type: none"> Use basic vi modes 	Lesson 4, Topic A
<ul style="list-style-type: none"> Insert, edit, delete, copy and find text 	Lesson 4, Topic A
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> vi 	Lesson 4, Topic A
<ul style="list-style-type: none"> /, ? 	Lesson 4, Topic A
<ul style="list-style-type: none"> h, j, k, l 	Lesson 4, Topic A

Exam Objective (LX0-103)	Course Lesson and Topic Reference
103.8 Perform basic file editing operations using vi	
<ul style="list-style-type: none"> i, o, a 	Lesson 4, Topic A
<ul style="list-style-type: none"> c, d, p, y, dd, yy 	Lesson 4, Topic A
<ul style="list-style-type: none"> ZZ, :w!, :q!, :e! 	Lesson 4, Topic A

104 Devices, Linux Filesystems, Filesystem Hierarchy Standard

Exam Objective (LX0-103)	Course Lesson and Topic Reference
104.1 Create partitions and filesystems	
<ul style="list-style-type: none"> Manage MBR partition tables 	Lesson 3, Topic A

Exam Objective (LX0-103)	Course Lesson and Topic Reference
104.1 Create partitions and filesystems	
<ul style="list-style-type: none"> Use various mkfs commands to set up partitions and create various filesystems such as: <ul style="list-style-type: none"> ext2/ext3/ext4 XFS VFAT 	Lesson 3, Topic A
<ul style="list-style-type: none"> Awareness of ReiserFS and Btrfs 	Lesson 3, Topic A
<ul style="list-style-type: none"> Basic knowledge of gdisk and parted with GPT 	Lesson 3, Topic A
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> fdisk 	Lesson 3, Topic A
<ul style="list-style-type: none"> gdisk 	Lesson 3, Topic A
<ul style="list-style-type: none"> parted 	Lesson 3, Topic A
<ul style="list-style-type: none"> mkfs 	Lesson 3, Topic A
<ul style="list-style-type: none"> mkswap 	Lesson 3, Topic C

Exam Objective (LX0-103)	Course Lesson and Topic Reference
104.2 Maintain the integrity of filesystems	

Exam Objective (LX0-103)	Course Lesson and Topic Reference
104.2 Maintain the integrity of filesystems	
<ul style="list-style-type: none"> Verify the integrity of filesystems 	Lesson 3, Topic D Lesson 15, Topic A
<ul style="list-style-type: none"> Monitor free space and inodes 	Lesson 15, Topic A Lesson 15, Topic C
<ul style="list-style-type: none"> Repair simple filesystem problems 	Lesson 3, Topic D
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> du 	Lesson 15, Topic A
<ul style="list-style-type: none"> df 	Lesson 15, Topic A
<ul style="list-style-type: none"> fsck 	Lesson 3, Topic D
<ul style="list-style-type: none"> e2fsck 	Lesson 3, Topic D
<ul style="list-style-type: none"> mke2fs 	Lesson 3, Topic A
<ul style="list-style-type: none"> debugfs 	Lesson 3, Topic D
<ul style="list-style-type: none"> dumpe2fs 	Lesson 3, Topic D
<ul style="list-style-type: none"> tune2fs 	Lesson 3, Topic D
<ul style="list-style-type: none"> xfs tools (such as xfs_metadump and xfs_info) 	Lesson 3, Topic D

Exam Objective (LX0-103)	Course Lesson and Topic Reference
104.3 Control mounting and unmounting of filesystems	
<ul style="list-style-type: none"> Manually mount and unmount filesystems 	Lesson 3, Topic C
<ul style="list-style-type: none"> Configure filesystem mounting on bootup 	Lesson 3, Topic A
<ul style="list-style-type: none"> Configure user mountable removable filesystems 	Lesson 3, Topic C
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> /etc/fstab 	Lesson 3, Topic A
<ul style="list-style-type: none"> /media 	Lesson 3, Topic B
<ul style="list-style-type: none"> mount 	Lesson 3, Topic C
<ul style="list-style-type: none"> umount 	Lesson 3, Topic C

Exam Objective (LX0-103)	Course Lesson and Topic Reference
104.4 Manage disk quotas	
<ul style="list-style-type: none"> Set up a disk quota for a filesystem 	Lesson 15, Topic C
<ul style="list-style-type: none"> Edit, check and generate user quota reports 	Lesson 15, Topic C
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> quota 	Lesson 15, Topic C
<ul style="list-style-type: none"> edquota 	Lesson 15, Topic C
<ul style="list-style-type: none"> repquota 	Lesson 15, Topic C
<ul style="list-style-type: none"> quotaon 	Lesson 15, Topic C

Exam Objective (LX0-103)	Course Lesson and Topic Reference
104.5 Manage file permissions and ownership	
<ul style="list-style-type: none"> Manage access permissions on regular and special files as well as directories 	Lesson 5, Topic A
<ul style="list-style-type: none"> Use access modes such as suid, sgid and the sticky bit to maintain security 	Lesson 5, Topic D
<ul style="list-style-type: none"> Know how to change the file creation mask 	Lesson 5, Topic B
<ul style="list-style-type: none"> Use the group field to grant file access to group members 	Lesson 5, Topic C
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> chmod 	Lesson 5, Topic A
<ul style="list-style-type: none"> umask 	Lesson 5, Topic B
<ul style="list-style-type: none"> chown 	Lesson 5, Topic C
<ul style="list-style-type: none"> chgrp 	Lesson 5, Topic C

Exam Objective (LX0-103)	Course Lesson and Topic Reference
104.6 Create and change hard and symbolic links	
<ul style="list-style-type: none"> Create links 	Lesson 4, Topic E
<ul style="list-style-type: none"> Identify hard and/or soft links 	Lesson 4, Topic E

Exam Objective (LX0-103)	Course Lesson and Topic Reference
104.6 Create and change hard and symbolic links	
<ul style="list-style-type: none"> Copying versus linking files 	Lesson 4, Topic E
<ul style="list-style-type: none"> Use links to support system administration tasks 	Lesson 4, Topic E
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> ln 	Lesson 4, Topic E
<ul style="list-style-type: none"> ls 	Lesson 1, Topic B

Exam Objective (LX0-103)	Course Lesson and Topic Reference
104.7 Find system files and place files in the correct location	
<ul style="list-style-type: none"> Understand the correct locations of files under the FHS 	Lesson 3, Topic B
<ul style="list-style-type: none"> Find files and commands on a Linux system 	Lesson 4, Topic B
<ul style="list-style-type: none"> Know the location and propose of important file and directories as defined in the FHS 	Lesson 3, Topic B
The following is a partial list of the used files, terms and utilities:	
<ul style="list-style-type: none"> find 	Lesson 4, Topic B
<ul style="list-style-type: none"> locate 	Lesson 4, Topic B
<ul style="list-style-type: none"> updatedb 	Lesson 4, Topic B
<ul style="list-style-type: none"> whereis 	Lesson 4, Topic B
<ul style="list-style-type: none"> which 	Lesson 1, Topic B
<ul style="list-style-type: none"> type 	Lesson 3, Topic A
<ul style="list-style-type: none"> /etc/updatedb.conf 	Lesson 4, Topic B

105 Shells, Scripting and Data Management

Exam Objective (LX0-104)	Course Lesson and Topic Reference
105.1 Customize and use the shell environment	
<ul style="list-style-type: none"> Set environment variables (e.g. PATH) at login or when spawning a new shell 	Lesson 9, Topic C

Exam Objective (LX0-104)	Course Lesson and Topic Reference
105.1 Customize and use the shell environment	
<ul style="list-style-type: none"> Write BASH functions for frequently used sequences of commands 	Lesson 9, Topic E
<ul style="list-style-type: none"> Maintain skeleton directories for new user accounts 	Lesson 2, Topic B
<ul style="list-style-type: none"> Set command search path with the proper directory 	Lesson 2, Topic B Lesson 9, Topic C
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> source 	Lesson 9, Topic C
<ul style="list-style-type: none"> /etc/bash.bashrc 	Lesson 9, Topic C
<ul style="list-style-type: none"> /etc/profile 	Lesson 9, Topic C
<ul style="list-style-type: none"> env 	Lesson 2, Topic B
<ul style="list-style-type: none"> export 	Lesson 2, Topic B
<ul style="list-style-type: none"> set 	Lesson 2, Topic B
<ul style="list-style-type: none"> unset 	Lesson 2, Topic B
<ul style="list-style-type: none"> ~/bash_profile 	Lesson 2, Topic B Lesson 9, Topic C
<ul style="list-style-type: none"> ~/bash_login 	Lesson 9, Topic D Lesson 17, Topic B
<ul style="list-style-type: none"> ~/profile 	Lesson 9, Topic D
<ul style="list-style-type: none"> ~/bashrc 	Lesson 9, Topic B Lesson 9, Topic C
<ul style="list-style-type: none"> ~/bash_logout 	Lesson 9, Topic D
<ul style="list-style-type: none"> function 	Lesson 9, Topic E
<ul style="list-style-type: none"> alias 	Lesson 9, Topic C
<ul style="list-style-type: none"> lists 	Lesson 9, Topic D

Exam Objective (LX0-104)	Course Lesson and Topic Reference
105.2 Customize or write simple scripts	
<ul style="list-style-type: none"> Use standard sh syntax (loops, tests) 	Lesson 1, Topic B Lesson 9, Topic B

Exam Objective (LX0-104)	Course Lesson and Topic Reference
105.2 Customize or write simple scripts	
<ul style="list-style-type: none"> Use command substitution 	Lesson 9, Topic D
<ul style="list-style-type: none"> Test return values for success or failure or other information provided by a command 	Lesson 9, Topic B
<ul style="list-style-type: none"> Perform conditional mailing to the superuser 	Lesson 9, Topic E
<ul style="list-style-type: none"> Correctly select the script interpreter through the shebang (!) line 	Lesson 9, Topic B
<ul style="list-style-type: none"> Manage the location, ownership, execution and suid-rights of scripts 	Lesson 9, Topic C

Exam Objective (LX0-104)	Course Lesson and Topic Reference
105.2 Customize or write simple scripts	
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> for 	Lesson 9, Topic E
<ul style="list-style-type: none"> while 	Lesson 9, Topic E
<ul style="list-style-type: none"> test 	Lesson 9, Topic B
<ul style="list-style-type: none"> if 	Lesson 9, Topic E
<ul style="list-style-type: none"> read 	Lesson 9, Topic D
<ul style="list-style-type: none"> seq 	Lesson 9, Topic D
<ul style="list-style-type: none"> exec 	Lesson 1, Topic B Lesson 9, Topic B

Exam Objective (LX0-104)	Course Lesson and Topic Reference
105.3 SQL data management	
<ul style="list-style-type: none"> Use of basic SQL commands 	Lesson 4, Topic G
<ul style="list-style-type: none"> Perform basic data manipulation 	Lesson 4, Topic G
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> insert 	Lesson 4, Topic G
<ul style="list-style-type: none"> update 	Lesson 4, Topic G

Exam Objective (LX0-104)	Course Lesson and Topic Reference
105.3 SQL data management	
<ul style="list-style-type: none"> select 	Lesson 4, Topic G
<ul style="list-style-type: none"> delete 	Lesson 4, Topic G
<ul style="list-style-type: none"> from 	Lesson 4, Topic G
<ul style="list-style-type: none"> where 	Lesson 4, Topic G
<ul style="list-style-type: none"> group by 	Lesson 4, Topic G
<ul style="list-style-type: none"> order by 	Lesson 4, Topic G
<ul style="list-style-type: none"> join 	Lesson 4, Topic G

106 User Interfaces and Desktops

Exam Objective (LX0-104)	Course Lesson and Topic Reference
106.1 Install and configure X11	
<ul style="list-style-type: none"> Verify that the video card and monitor are supported by an X server 	Lesson 17, Topic A Lesson 18, Topic A
<ul style="list-style-type: none"> Awareness of the X font server 	Lesson 18, Topic A
<ul style="list-style-type: none"> Basic understanding and knowledge of the X Window configuration file 	Lesson 18, Topic A

Exam Objective (LX0-104)	Course Lesson and Topic Reference
106.1 Install and configure X11	
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> /etc/X11/xorg.conf 	Lesson 16, Topic B Lesson 18, Topic A
<ul style="list-style-type: none"> xhost 	Lesson 18, Topic A
<ul style="list-style-type: none"> DISPLAY 	Lesson 18, Topic A
<ul style="list-style-type: none"> xwininfo 	Lesson 18, Topic A
<ul style="list-style-type: none"> xdpyinfo 	Lesson 18, Topic A
<ul style="list-style-type: none"> X 	Lesson 18, Topic A

Exam Objective (LX0-104)	Course Lesson and Topic Reference
106.2 Setup a display manager	
<ul style="list-style-type: none"> Basic configuration of LightDM 	Lesson 18, Topic B
<ul style="list-style-type: none"> Turn the display manager on or off 	Lesson 18, Topic A
<ul style="list-style-type: none"> Change the display manager greeting 	Lesson 18, Topic B
<ul style="list-style-type: none"> Awareness of XDM, KDM, and GDM 	Lesson 18, Topic B
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> lightdm 	Lesson 1, Topic D Lesson 18, Topic B
<ul style="list-style-type: none"> /etc/lightdm 	Lesson 18, Topic B

Exam Objective (LX0-104)	Course Lesson and Topic Reference
106.3 Accessibility	
<ul style="list-style-type: none"> Basic knowledge of keyboard accessibility settings (AccessX) 	Lesson 18, Topic D
<ul style="list-style-type: none"> Basic knowledge of visual settings and themes 	Lesson 18, Topic D
<ul style="list-style-type: none"> Basic knowledge of assistive technology (ATs) 	Lesson 18, Topic D
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> Sticky/Repeat Keys 	Lesson 18, Topic D
<ul style="list-style-type: none"> Slow/Bounce/Toggle Keys 	Lesson 18, Topic D
<ul style="list-style-type: none"> Mouse Keys 	Lesson 18, Topic D
<ul style="list-style-type: none"> High Contrast/Large Print Desktop Themes 	Lesson 18, Topic D
<ul style="list-style-type: none"> Screen Reader 	Lesson 18, Topic D

Exam Objective (LX0-104)	Course Lesson and Topic Reference
106.3 Accessibility	
	Lesson 18, Topic D
<ul style="list-style-type: none"> Braille Display 	
	Lesson 18, Topic D
<ul style="list-style-type: none"> Screen Magnifier 	

Exam Objective (LX0-104)	Course Lesson and Topic Reference
106.3 Accessibility	
<ul style="list-style-type: none"> On-Screen Keyboard 	Lesson 18, Topic D
<ul style="list-style-type: none"> Gestures (used at login, for example gdm) 	Lesson 18, Topic D
<ul style="list-style-type: none"> Orca 	Lesson 18, Topic D
<ul style="list-style-type: none"> GOK 	Lesson 18, Topic D
<ul style="list-style-type: none"> emacspeak 	Lesson 18, Topic D

107 Administrative Tasks

Exam Objective (LX0-104)	Course Lesson and Topic Reference
107.1 Manage user and group accounts and related system files	
<ul style="list-style-type: none"> Add, modify and remove users and groups 	Lesson 2, Topic A Lesson 2, Topic C
<ul style="list-style-type: none"> Manage user/group info in password/group databases 	Lesson 2, Topic C
<ul style="list-style-type: none"> Create and manage special purpose and limited accounts 	Lesson 2, Topic A Lesson 2, Topic C
	Lesson 14, Topic B
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> /etc/passwd 	Lesson 2, Topic A
<ul style="list-style-type: none"> /etc/shadow 	Lesson 2, Topic A
<ul style="list-style-type: none"> /etc/group 	Lesson 2, Topic A
<ul style="list-style-type: none"> /etc/skel 	Lesson 2, Topic B
<ul style="list-style-type: none"> chage 	Lesson 2, Topic C
<ul style="list-style-type: none"> getent 	Lesson 12, Topic C

Exam Objective (LX0-104)	Course Lesson and Topic Reference
107.1 Manage user and group accounts and related system files	
• groupadd	Lesson 2, Topic A
• groupdel	Lesson 2, Topic C
• groupmod	Lesson 2, Topic C
• passwd	Lesson 2, Topic A
• useradd	Lesson 2, Topic A
• userdel	Lesson 2, Topic C
• usermod	Lesson 2, Topic C

Exam Objective (LX0-104)	Course Lesson and Topic Reference
107.2 Automate system administration tasks by scheduling jobs	
• Manage cron and at jobs	Lesson 10, Topic D
• Configure user access to cron and at services	Lesson 10, Topic D
• Configure anacron	Lesson 10, Topic D
The following is a partial list of the used files, terms, and utilities:	
• /etc/cron.{d,daily,hourly,monthly,weekly}	Lesson 10, Topic D
• /etc/at.deny	Lesson 10, Topic D
• /etc/at.allow	Lesson 10, Topic D
• /etc/crontab	Lesson 10, Topic D
• /etc/cron.allow	Lesson 10, Topic D
• /etc/cron.deny	Lesson 10, Topic D
• /var/spool/cron/*	Lesson 10, Topic D
• crontab	Lesson 10, Topic D
• at	Lesson 10, Topic D
• atq	Lesson 10, Topic D

Exam Objective (LX0-104)	Course Lesson and Topic Reference
107.2 Automate system administration tasks by scheduling jobs	
<ul style="list-style-type: none"> atrm 	Lesson 10, Topic D
<ul style="list-style-type: none"> anacron 	Lesson 10, Topic D
<ul style="list-style-type: none"> /etc/anacrontab 	Lesson 10, Topic D

Exam Objective (LX0-104)	Course Lesson and Topic Reference
107.3 Localization and internationalization	
<ul style="list-style-type: none"> Configure locale settings and environment variables 	Lesson 10, Topic E
<ul style="list-style-type: none"> Configure timezone settings and environment variables 	Lesson 10, Topic E
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> /etc/timezone 	Lesson 10, Topic E
<ul style="list-style-type: none"> /etc/localtime 	Lesson 10, Topic E
<ul style="list-style-type: none"> /usr/share/zoneinfo 	Lesson 10, Topic E
<ul style="list-style-type: none"> Environment variables: <ul style="list-style-type: none"> LC_* LC_ALL LANG TZ 	Lesson 10, Topic E
<ul style="list-style-type: none"> /usr/bin/locale 	Lesson 10, Topic E
<ul style="list-style-type: none"> tzselect 	Lesson 10, Topic E
<ul style="list-style-type: none"> tzconfig 	Lesson 10, Topic E
<ul style="list-style-type: none"> date 	Lesson 10, Topic E
<ul style="list-style-type: none"> iconv 	Lesson 10, Topic E
<ul style="list-style-type: none"> UTF-8 	Lesson 10, Topic E
<ul style="list-style-type: none"> ISO-8859 	Lesson 10, Topic E

Exam Objective (LX0-104)	Course Lesson and Topic Reference
107.3 Localization and internationalization	
<ul style="list-style-type: none"> • ASCII 	Lesson 10, Topic E
<ul style="list-style-type: none"> • Unicode 	Lesson 10, Topic E

108 Essential System Services

Exam Objective (LX0-104)	Course Lesson and Topic Reference
108.1 Maintain system time	
<ul style="list-style-type: none"> • Set the system date and time 	Lesson 10, Topic E
<ul style="list-style-type: none"> • Set the hardware clock to the correct time in UTC 	Lesson 10, Topic E
<ul style="list-style-type: none"> • Configure the correct timezone 	Lesson 10, Topic E
<ul style="list-style-type: none"> • Basic NTP configuration 	Lesson 10, Topic E
<ul style="list-style-type: none"> • Knowledge of using the pool.ntp.org service 	Lesson 10, Topic E
<ul style="list-style-type: none"> • Awareness of the ntpq command 	Lesson 10, Topic E
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> • /usr/share/zoneinfo 	Lesson 10, Topic E
<ul style="list-style-type: none"> • /etc/timezone 	Lesson 10, Topic E
<ul style="list-style-type: none"> • /etc/localtime 	Lesson 10, Topic E
<ul style="list-style-type: none"> • /etc/ntp.conf 	Lesson 10, Topic E
<ul style="list-style-type: none"> • date 	Lesson 10, Topic E
<ul style="list-style-type: none"> • hwclock 	Lesson 10, Topic E
<ul style="list-style-type: none"> • ntpd 	Lesson 10, Topic E
<ul style="list-style-type: none"> • ntpdate 	Lesson 10, Topic E
<ul style="list-style-type: none"> • pool.ntp.org 	Lesson 10, Topic E

Exam Objective (LX0-104)	Course Lesson and Topic Reference
108.2 System logging	

Exam Objective (LX0-104)	Course Lesson and Topic Reference
108.2 System logging	
<ul style="list-style-type: none"> Configuration of the syslog daemon 	Lesson 11, Topic B
<ul style="list-style-type: none"> Understanding of standard facilities, priorities and actions 	Lesson 1, Topic D
<ul style="list-style-type: none"> Configuration of logrotate 	Lesson 11, Topic B
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> syslog.conf 	Lesson 11, Topic B
<ul style="list-style-type: none"> syslogd 	Lesson 11, Topic B
<ul style="list-style-type: none"> klogd 	Lesson 11, Topic B
<ul style="list-style-type: none"> /var/log/ 	Lesson 11, Topic B
<ul style="list-style-type: none"> logger 	Lesson 11, Topic B
<ul style="list-style-type: none"> logrotate 	Lesson 10, Topic D
<ul style="list-style-type: none"> /etc/logrotate.conf 	Lesson 10, Topic D
<ul style="list-style-type: none"> /etc/logrotate.d/ 	Lesson 10, Topic D
<ul style="list-style-type: none"> journalctl 	Lesson 11, Topic B
<ul style="list-style-type: none"> /etc/systemd/journald.conf 	Lesson 11, Topic B
<ul style="list-style-type: none"> /var/log/journal/ 	Lesson 11, Topic B

Exam Objective (LX0-104)	Course Lesson and Topic Reference
108.3 Mail Transfer Agent (MTA) basics	
<ul style="list-style-type: none"> Create e-mail aliases 	Lesson 13, Topic A
<ul style="list-style-type: none"> Configure e-mail forwarding 	Lesson 13, Topic A
<ul style="list-style-type: none"> Knowledge of commonly available MTA programs (postfix, sendmail, qmail, exim) (no configuration) 	Lesson 13, Topic A
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> ~/forward 	Lesson 13, Topic A
<ul style="list-style-type: none"> sendmail emulation layer commands 	Lesson 13, Topic A

Exam Objective (LX0-104)	Course Lesson and Topic Reference
108.3 Mail Transfer Agent (MTA) basics	
<ul style="list-style-type: none"> newaliases 	Lesson 13, Topic A
<ul style="list-style-type: none"> mail 	Lesson 13, Topic A
<ul style="list-style-type: none"> mailq 	Lesson 13, Topic A
<ul style="list-style-type: none"> postfix 	Lesson 13, Topic A
<ul style="list-style-type: none"> sendmail 	Lesson 13, Topic A
<ul style="list-style-type: none"> exim 	Lesson 13, Topic A

Exam Objective (LX0-104)	Course Lesson and Topic Reference
108.3 Mail Transfer Agent (MTA) basics	

<ul style="list-style-type: none"> qmail 	Lesson 13, Topic A
---	--------------------

Exam Objective (LX0-104)	Course Lesson and Topic Reference
108.4 Manage printers and printing	
<ul style="list-style-type: none"> Basic CUPS configuration (for local and remote printers) 	Lesson 6, Topic A
<ul style="list-style-type: none"> Manage user print queues 	Lesson 6, Topic C
<ul style="list-style-type: none"> Troubleshoot general printing problems 	Lesson 16, Topic B
<ul style="list-style-type: none"> Add and remove jobs from configured printer queues 	Lesson 6, Topic C
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> CUPS configuration files, tools and utilities 	Lesson 6, Topic A
<ul style="list-style-type: none"> /etc/cups 	Lesson 6, Topic A
<ul style="list-style-type: none"> lpd legacy interface (lpr, lprm, lpq) 	Lesson 6, Topic A

109 Networking Fundamentals

Exam Objective (LX0-104)	Course Lesson and Topic Reference
109.1 Fundamentals of internet protocols	

Exam Objective (LX0-104)	Course Lesson and Topic Reference
109.1 Fundamentals of internet protocols	
<ul style="list-style-type: none"> Demonstrate an understanding network masks and CIDR notation 	Lesson 12, Topic A
<ul style="list-style-type: none"> Knowledge of the differences between private and public "dotted quad" IP-Addresses 	Lesson 12, Topic A
<ul style="list-style-type: none"> Knowledge about common TCP and UDP ports (20, 21, 22, 23, 25, 53, 80, 110, 123, 139, 143, 161, 162, 389, 443, 465, 514, 636, 993, 995) 	Lesson 12, Topic A Lesson 13, Topic A
<ul style="list-style-type: none"> Knowledge about the differences and major features of UDP, TCP and ICMP 	Lesson 12, Topic A
<ul style="list-style-type: none"> Knowledge of the major differences between IPv4 and IPv6 	Lesson 12, Topic A
<ul style="list-style-type: none"> Knowledge of the basic features of IPv6 	Lesson 12, Topic A
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> /etc/services 	Lesson 12, Topic A
<ul style="list-style-type: none"> IPv4, IPv6 	Lesson 12, Topic A
<ul style="list-style-type: none"> Subnetting 	Lesson 12, Topic A
<ul style="list-style-type: none"> TCP, UDP, ICMP 	Lesson 12, Topic A

Exam Objective (LX0-104)	Course Lesson and Topic Reference
109.2 Basic network configuration	
<ul style="list-style-type: none"> Manually and automatically configure network interfaces 	Lesson 12, Topic A
<ul style="list-style-type: none"> Basic TCP/IP host configuration 	Lesson 12, Topic A
<ul style="list-style-type: none"> Setting a default route 	Lesson 12, Topic B
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> /etc/hostname 	Lesson 12, Topic A
<ul style="list-style-type: none"> /etc/hosts 	Lesson 12, Topic C
<ul style="list-style-type: none"> /etc/nsswitch.conf 	Lesson 12, Topic C
<ul style="list-style-type: none"> ifconfig 	Lesson 12, Topic A

Exam Objective (LX0-104)	Course Lesson and Topic Reference
109.2 Basic network configuration	
<ul style="list-style-type: none"> • ifup 	Lesson 12, Topic A
<ul style="list-style-type: none"> • ifdown 	Lesson 12, Topic A
<ul style="list-style-type: none"> • ip 	Lesson 12, Topic A
<ul style="list-style-type: none"> • route 	Lesson 12, Topic B
<ul style="list-style-type: none"> • ping 	Lesson 12, Topic A

Exam Objective (LX0-104)	Course Lesson and Topic Reference
109.3 Basic network troubleshooting	

<ul style="list-style-type: none"> • Manually and automatically configure network interfaces and routing tables to include adding, starting, stopping, restarting, deleting or reconfiguring network interfaces 	Lesson 12, Topic A
<ul style="list-style-type: none"> • Change, view or configure the routing table and correct an improperly set default route manually 	Lesson 12, Topic B
<ul style="list-style-type: none"> • Debug problems associated with the network configuration 	Lesson 16, Topic C
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> • ifconfig 	Lesson 12, Topic A
<ul style="list-style-type: none"> • ip 	Lesson 12, Topic A
<ul style="list-style-type: none"> • ifup 	Lesson 12, Topic A
<ul style="list-style-type: none"> • ifdown 	Lesson 12, Topic A

Exam Objective (LX0-104)	Course Lesson and Topic Reference
109.3 Basic network troubleshooting	
<ul style="list-style-type: none"> • route 	Lesson 12, Topic B
<ul style="list-style-type: none"> • host 	Lesson 12, Topic C
<ul style="list-style-type: none"> • hostname 	Lesson 1, Topic B Lesson 9, Topic C
<ul style="list-style-type: none"> • dig 	Lesson 12, Topic C

Exam Objective (LX0-104)	Course Lesson and Topic Reference
109.3 Basic network troubleshooting	
<ul style="list-style-type: none"> netstat 	Lesson 12, Topic B
<ul style="list-style-type: none"> ping 	Lesson 12, Topic A
<ul style="list-style-type: none"> ping6 	Lesson 12, Topic B
<ul style="list-style-type: none"> tracert 	Lesson 12, Topic B
<ul style="list-style-type: none"> tracert6 	Lesson 12, Topic B
<ul style="list-style-type: none"> tracert6 	Lesson 12, Topic B
<ul style="list-style-type: none"> tracert6 	Lesson 12, Topic B
<ul style="list-style-type: none"> tracert6 	Lesson 12, Topic B
<ul style="list-style-type: none"> netcat 	Lesson 12, Topic B

Exam Objective (LX0-104)	Course Lesson and Topic Reference
109.4 Configure client side DNS	
<ul style="list-style-type: none"> Query remote DNS servers 	Lesson 12, Topic C
<ul style="list-style-type: none"> Configure local name resolution and use remote DNS servers 	Lesson 12, Topic C
<ul style="list-style-type: none"> Modify the order in which name resolution is done 	Lesson 12, Topic C
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> /etc/hosts 	Lesson 12, Topic C
<ul style="list-style-type: none"> /etc/resolv.conf 	Lesson 12, Topic C
<ul style="list-style-type: none"> /etc/nsswitch.conf 	Lesson 12, Topic C
<ul style="list-style-type: none"> host 	Lesson 12, Topic C
<ul style="list-style-type: none"> dig 	Lesson 12, Topic C
<ul style="list-style-type: none"> getent 	Lesson 12, Topic C

110 Security

Exam Objective (LX0-104)	Course Lesson and Topic Reference
110.1 Perform security administration tasks	

Exam Objective (LX0-104)	Course Lesson and Topic Reference
110.1 Perform security administration tasks	
<ul style="list-style-type: none"> Audit a system to find files with the <code>suid/sgid</code> bit set 	Lesson 5, Topic D
<ul style="list-style-type: none"> Set or change user passwords and password aging information 	Lesson 2, Topic C
<ul style="list-style-type: none"> Being able to use <code>nmap</code> and <code>netstat</code> to discover open ports on a system 	Lesson 12, Topic B Lesson 14, Topic E
<ul style="list-style-type: none"> Set up limits on user logins, processes and memory usage 	Lesson 14, Topic B
<ul style="list-style-type: none"> Determine which users have logged in to the system or are currently logged in 	Lesson 1, Topic B
<ul style="list-style-type: none"> Basic <code>sudo</code> configuration and usage 	Lesson 14, Topic B
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> <code>find</code> 	Lesson 4, Topic B
<ul style="list-style-type: none"> <code>passwd</code> 	Lesson 2, Topic A
<ul style="list-style-type: none"> <code>fuser</code> 	Lesson 3, Topic C
<ul style="list-style-type: none"> <code>lsof</code> 	Lesson 16, Topic A
<ul style="list-style-type: none"> <code>nmap</code> 	Lesson 14, Topic E
<ul style="list-style-type: none"> <code>chage</code> 	Lesson 2, Topic C
<ul style="list-style-type: none"> <code>netstat</code> 	Lesson 12, Topic B
<ul style="list-style-type: none"> <code>sudo</code> 	Lesson 14, Topic B
<ul style="list-style-type: none"> <code>/etc/sudoers</code> 	Lesson 14, Topic B
<ul style="list-style-type: none"> <code>su</code> 	Lesson 14, Topic B
<ul style="list-style-type: none"> <code>usermod</code> 	Lesson 2, Topic C
<ul style="list-style-type: none"> <code>ulimit</code> 	Lesson 14, Topic B
<ul style="list-style-type: none"> <code>who, w, last</code> 	Lesson 8, Topic D

Exam Objective (LX0-104)	Course Lesson and Topic Reference
110.2 Setup host security	

Exam Objective (LX0-104)	Course Lesson and Topic Reference
110.2 Setup host security	
<ul style="list-style-type: none"> Awareness of shadow passwords and how they work 	Lesson 14, Topic B
<ul style="list-style-type: none"> Turn off network services not in use 	Lesson 12, Topic A
<ul style="list-style-type: none"> Understand the role of TCP wrappers 	Lesson 14, Topic A
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> /etc/nologin 	Lesson 1, Topic D
<ul style="list-style-type: none"> /etc/passwd 	Lesson 2, Topic A
<ul style="list-style-type: none"> /etc/shadow 	Lesson 2, Topic A
<ul style="list-style-type: none"> /etc/xinetd.d/* 	Lesson 13, Topic D
<ul style="list-style-type: none"> /etc/xinetd.conf 	Lesson 13, Topic B
<ul style="list-style-type: none"> /etc/inetd.d/* 	Lesson 13, Topic B
<ul style="list-style-type: none"> /etc/inetd.conf 	Lesson 12, Topic A
<ul style="list-style-type: none"> /etc/inittab 	Lesson 11, Topic A
<ul style="list-style-type: none"> /etc/init.d/* 	Lesson 11, Topic A
<ul style="list-style-type: none"> /etc/hosts.allow 	Lesson 14, Topic A Lesson 14, Topic B
<ul style="list-style-type: none"> /etc/hosts.deny 	Lesson 14, Topic B

Exam Objective (LX0-104)	Course Lesson and Topic Reference
110.3 Securing data with encryption	
<ul style="list-style-type: none"> Perform basic OpenSSH 2 client configuration and usage 	Lesson 12, Topic D
<ul style="list-style-type: none"> Understand the role of OpenSSH 2 server host keys 	Lesson 12, Topic D
<ul style="list-style-type: none"> Perform basic GnuPG configuration, usage and revocation 	Lesson 7, Topic D Lesson 14, Topic A
<ul style="list-style-type: none"> Understand SSH port tunnels (including X11 tunnels) 	Lesson 12, Topic D
The following is a partial list of the used files, terms, and utilities:	
<ul style="list-style-type: none"> ssh 	Lesson 12, Topic D

Exam Objective (LX0-104)	Course Lesson and Topic Reference
110.3 Securing data with encryption	
<ul style="list-style-type: none"> ssh-keygen 	Lesson 12, Topic D
<ul style="list-style-type: none"> ssh-agent 	Lesson 12, Topic D
<ul style="list-style-type: none"> ssh-add 	Lesson 12, Topic D
<ul style="list-style-type: none"> ~/.ssh/id_rsa and id_rsa.pub 	Lesson 12, Topic D
<ul style="list-style-type: none"> ~/.ssh/id_dsa and id_dsa.pub 	Lesson 12, Topic D
<ul style="list-style-type: none"> /etc/ssh/ssh_host_rsa_key and ssh_host_rsa_key.pub 	Lesson 12, Topic D
<ul style="list-style-type: none"> /etc/ssh/ssh_host_dsa_key and ssh_host_dsa_key.pub 	Lesson 12, Topic D
<ul style="list-style-type: none"> ~/.ssh/authorized_keys 	Lesson 12, Topic D
<ul style="list-style-type: none"> /etc/ssh_known_hosts 	Lesson 12, Topic D
<ul style="list-style-type: none"> gpg 	Lesson 7, Topic D Lesson 14, Topic A
<ul style="list-style-type: none"> ~/.gnupg/* 	Lesson 14, Topic A

C Syntax

Introduction

The following is a list of the most frequently used commands with their syntax. Candidates are encouraged to review the complete list and attain a working knowledge of all listed commands as a part of a comprehensive exam preparation program.

<i>Command</i>	<i>Syntax</i>
alias	alias {command}='{command} [options]'
apropos	apropos {keyword}
apt-get	apt-get [options] {command}
aspell	aspell [options]
at	at [options] {time}
awk	awk [options] {file}
bg	bg {%#}
bunzip2	bunzip2 {file name}
bzcat	bzcat {file name}
bzdiff	bzdiff {file name}
bzip2	bzip2 {file name}
bzip2recover	bzip2recover {file name}
bzless	bzless {file name}
bzmore	bzmore {file name}
cal	cal {month} {year}
cancel	cancel [command options]
cat	cat [command options] {file name}
cd	cd {absolute or relative path}
cp	cp [command options] {absolute or relative path of the file or directory to be copied}/{file or directory name} {absolute or relative path of the destination}
chown	chown {user name} {file name} OR chown {user name.group name} {file name} OR chown {user name.} {file name} OR chown {.group name} {file name}
chattr	chattr [-RV] [-v version] { mode } files
count	operator [count] {motion}
chmod	chmod [option] {mode} {file name}
createrepo	createrepo [options] <directory>
cron	cron [option] {mail command}
chgrp	chgrp {group} {file name}
chkconfig	chkconfig [option] {service name} {on off reset}
cp	cp [options] {absolute or relative path of the file or directory to be copied}/{file or directory name} {absolute or relative path of the destination}
date	date +[format]

Command	Syntax
dd	dd [operand]... OR dd [option]
dump	dump {-level #} {dump file} {filesystem/file/directory}
date	date +[format]
dumpe2fs	dumpe2fs [options] {block size} {device name}
diff	diff {file name 1} {file name 2}
dig	dig [command options] {query options} {Fully Qualified Domain Name IP address}
dpkg	dpkg [option] {action}
echo	echo {"string"}
e2fsck	e2fsck /dev/{filesystem}
e2label	e2label /dev/{device name} {partition number}
export	export variable
finger	finger [user name]
fdisk	fdisk [option] {device name}
fg	fg {%%#}
fsck	fsck -t {filesystem type} [options] OR fsck -r /dev/{filesystem}
find	find [options] {search locations} {search criteria} {actions}
file	file [options] {file name}
ftp	ftp [command options] {hostname}
gzip	gzip [command options] {file name}
groupdel	groupdel {group name}
groupmod	groupmod [-g gid [-o]] [-n new group name] [old group name]
groupadd	groupadd [options] {group name}
parted	parted [option] device {command [argument]}
grep	grep [command options] {keyword} {file name}
gpg	gpg [options] {command} {arguments}
groupadd	groupadd {group name}
groupdel	groupdel {group name}
groupmod	groupmod -g {GID}
history	history [options]
host	host [command options] {FQDN IP address}
hostname	hostname [options] {hostname}
inetd	inetd [option] [configuration file]
info	info {command}
insmod	insmod {file name} {module options}
ifconfig	ifconfig {interface name} {options or address}
ip	ip [options] {object} {command ! help}
iwconfig	iwconfig {interface name} {options or address}

Command	Syntax
iptables	iptables [-t table] {commands} {chain/rule specification} [options/parameters]
ls	ls [options] [absolute or relative path of the directory]
kill	kill [signal option] {PID}
killall	killall [signal option] {command}
klogd	klogd [options]
last	last [options]
lastlog	lastlog [options]
lp	lp [command options] {file name}
ln	ln [option] [-T] {target link name}
ls	ls [command options] [absolute or relative path of the directory]
logrotate	logrotate [options] {configuration file}
lsmod	lsmod
lpr	lpr [command options] {file name}
lpq	lpq [command options] {print queue name}
lprm	lprm {print job id}
lpc	lpc [parameter]
lpstat	lpstat [command options]
links	links [options] {URL}
lsattr	lsattr [-RVadv] [files...]
locate	locate [options] {string}
mv	mv {absolute or relative path}/{file or directory name} {absolute or relative path}/{new file or directory name}
modinfo	modinfo {module options}
man	man topic
mkinitrd	mkinitrd [options] {image name} {kernel version}
mkfs	mkfs [options] {filesystem}
mke2fs	mke2fs [options] {filesystem name}
mkdir	mkdir {directory name}
mount	mount [options] {device} {mountpoint}
mkswap	mkswap [option] device [size]
modprobe	modprobe [option] {module name}
mknod	mknod [option]... {name} {type} [major minor]
md5sum	md5sum --check {file name}
make	make {key file digital certificate}
mdadm	mdadm {mode} {raid device} [options] {component devices}
mv	mv {absolute or relative path}/{file or directory name} {absolute or relative path}/{new file or directory name}
netstat	netstat [options]
nice	nice -n {priority} {command}
nohup	nohup {command}
nslookup	nslookup {host name or FQDN}
nmap	nmap [scan type] [options] {target specification}

Command	Syntax
openssl	openssl <i>{command}</i> <i>[options]</i> <i>{arguments}</i>
parted	parted <i>[option]</i> device <i>{command [argument]}</i>
partprobe	partprobe <i>[options]</i> <i>[device]</i>
pr	pr <i>[command options]</i> <i>{file name}</i>
passwd	passwd <i>[user name]</i>
pwd	pwd <i>[option]</i>
ps	ps <i>[options]</i>
pstree	pstree <i>[options]</i>
pidof	pidof <i>[command options]</i> <i>{string}</i>
pgrep	pgrep <i>[command options]</i> <i>{process name}</i>
pkill	pkill <i>[signal option]</i> <i>{command}</i>
popd	popd <i>[options]</i>
pushd	pushd <i>[options]</i> <i>{directory name}</i>
renice	renice <i>{priority}</i> <i>[options]</i>
restore	restore <i>[options]</i> <i>{file}</i>
rm	rm <i>[command options]</i> <i>{absolute or relative path of file or directory}/{file or directory name}</i>
rmdir	rmdir <i>{directory name}</i>
rpm -q	rpm -q <i>{what_packages}</i> <i>{what_information}</i>
rpm -V	rpm -V <i>package_name</i>
rndc	rndc <i>[rndc options]</i> <i>{rndc command}</i>
rsync	rsync <i>{source file or folder}</i> <i>{destination file or folder}</i>
rdesktop	rdesktop <i>[options]</i> server[:port]
service	service <i>{service name}</i> <i>{options}</i>
sleep	sleep <i>{time}</i>
shutdown	shutdown [-t seconds] [-options] time <i>[warning message]</i>
sfdisk	sfdisk <i>[options]</i> device
swapon	swapon -e OR swapon -a
swapoff	swapoff -a
sysctl	sysctl <i>[command options]</i> <i>{kernel parameter}={value}</i>
syslogd	syslogd <i>[options]</i>

Command	Syntax
system-config-services	system-config-services
ssh-keygen	ssh-keygen <i>[options]</i>
sed	sed 'address/pattern/action' file name
smbclient	smbclient //machine/service
sftp	sftp hostname
sha1sum	sha1sum --check <i>{file name}</i>
sudo	sudo command-name command-options

Command	Syntax
tail	tail <i>[options]</i> {file name}
tar	tar <i>[archiving command options]</i> {destination file}.tar
	{source directory}
tee	tee <i>[options]</i> {file}
test	test {expressions}
telinit	telinit {runlevel}
top	top <i>[options]</i>
tr	tr {'character 1'} {'character 2'} < {file name}
traceroute	traceroute <i>[options]</i> {hostname ip address}
touch	touch {file name}
tune2fs	tune2fs <i>[options]</i> {device name}
tmpwatch	tmpwatch <i>[options]</i> {hours}
tcpdump	tcpdump <i>[option]</i> {expression}
uname	uname <i>[options]</i>
uptime	uptime OR uptime [-V]
useradd	useradd <i>[command options]</i> {user name}
userdel	userdel <i>[command options]</i> {user name}
usermod	usermod <i>[command options]</i> {user name}
umount	umount <i>[options]</i> {directory device}
uniq	uniq <i>[command options]</i> {file name}
unzip	unzip <i>[command options]</i> {file name}
umask	umask <i>number</i>
vim	vim <i>[options]</i> {file}
vncserver	vncserver {: <i>display number</i> } {- <i>option</i> }
vncviewer	vncviewer <i>[options]</i> {hostname ipaddress}{: <i>display</i> }

Command	Syntax
w	w <i>[options]</i> {user name}
wc	wc <i>[command options]</i> {file name}
vim	vim {file name}
whatis	whatis <i>command</i>
which	which {file name}
whereis	whereis [-bmsu] [-BMS directory... -f] file name ...
who	who <i>[options]</i>
whoami	whoami <i>[option]</i> ...
wireshark	wireshark <i>[options]</i>
wget	wget <i>[command options]</i> http://{hostname or IP of the destination}
xargs	xargs <i>[options]</i> {commands}
yum	yum <i>[options]</i> {command} {package name}

Command	Syntax
yumdownloader	yumdownloader <i>[options]</i> <i>{package name}</i>

D Additional Security Topics

Appendix Introduction

There are many topics on securing Red Hat® Enterprise Linux® servers and the services they provide. In this appendix, selected additional content is provided about Linux firewalls (via iptables), tools and techniques for security auditing, and Intrusion Detection Systems (IDS).

TOPIC A Enable Firewall Functionality

In this course, you secured user accounts as a level of security defense on your network. Now, you will look at packet filtering to provide firewall functionality to routers, gateways, and Linux servers and workstations. In this topic, you will implement iptables to provide firewall functionality by packet filtering.

When dealing with data navigating through your network, you will want to implement additional filtering. Understanding iptables will enable you to provide firewall functionalities by packet filtering to secure a Linux system. As a Linux administrator, you will need to provide a secure network and continuity of services, especially on large networks.

Firewalls

A **firewall** is a software program or a hardware device that protects a system or a network from unauthorized access by blocking unsolicited traffic. A firewall allows incoming or outgoing traffic that is specifically permitted by an administrator. It also allows incoming traffic that is sent in response to requests from internal hosts. Firewalls often provide logging features and alarms that track security problems and report them to the administrator. Firewalls use packet filtering and proxy servers to implement security on a network.

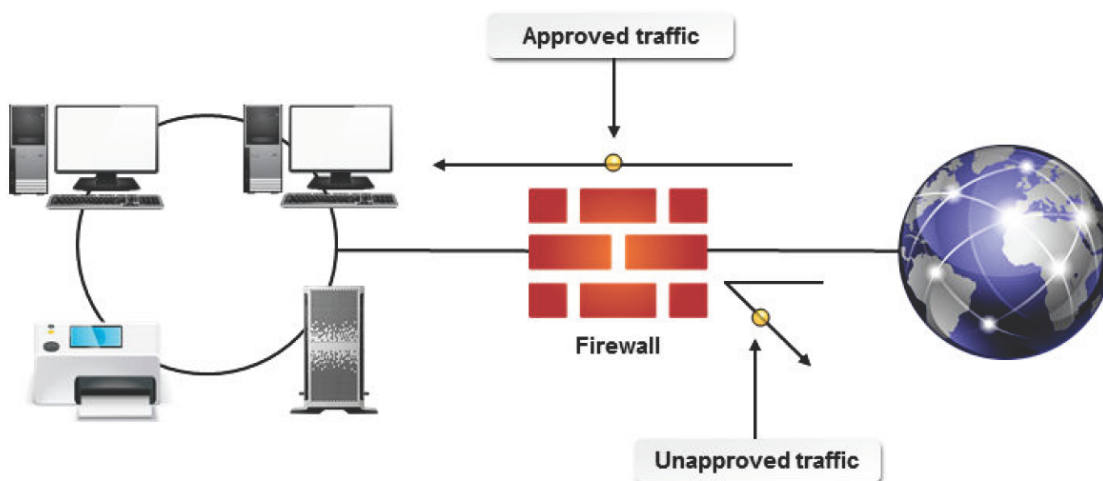


Figure D-1: Firewall protects a network by filtering incoming and outgoing traffic.

Software Firewalls

The word "firewall" is generally used to refer to software-based firewalls. Software firewalls can be useful for small home offices and businesses. The firewall provides many features that can be configured to suit various computing needs. Some features include:

- Enabling or disabling port security on certain ports.
- Filtering inbound and outbound communication. A user can set up rules or exceptions in the firewall settings to limit access to the web.
- Reporting and logging activity.

- Protecting systems from malware and spyware.
- Blocking pop-up messages.
- Assigning, forwarding, and triggering ports.

Hardware Firewalls

A hardware firewall is a hardware device, either stand-alone or built into most routers, that protects computers on a private network from unauthorized traffic. They are placed between the private network and the public network to manage inbound and outbound traffic and network access.

Firewall Positioning

A firewall can be positioned logically between the internal network and the external world. In addition, it can be positioned between internal corporate networks and on individual servers. It is recommended to configure the firewall to either deny or grant access, based on the rules assigned by the security administrator.

Packet Filtering

Packet filtering is a process of passing or blocking incoming and outgoing packets after inspecting each packet for user-defined content, such as an IP address. It was the first type of firewall that was used to protect networks by reading packet headers. Packet filtering was limited by the fact that it was designed to look only at packet header information. **Netfilter** is a framework that implements packet filtering in Linux, to manage firewalls and secure data. It contains a number of modules that inspect packet headers and filter packets with improper headers. The packets pass through various stages before they are sent to their destination. The packets are filtered at specific points and the filtered packets are sent to the next stage. Netfilter activities can be configured based on user requirements.

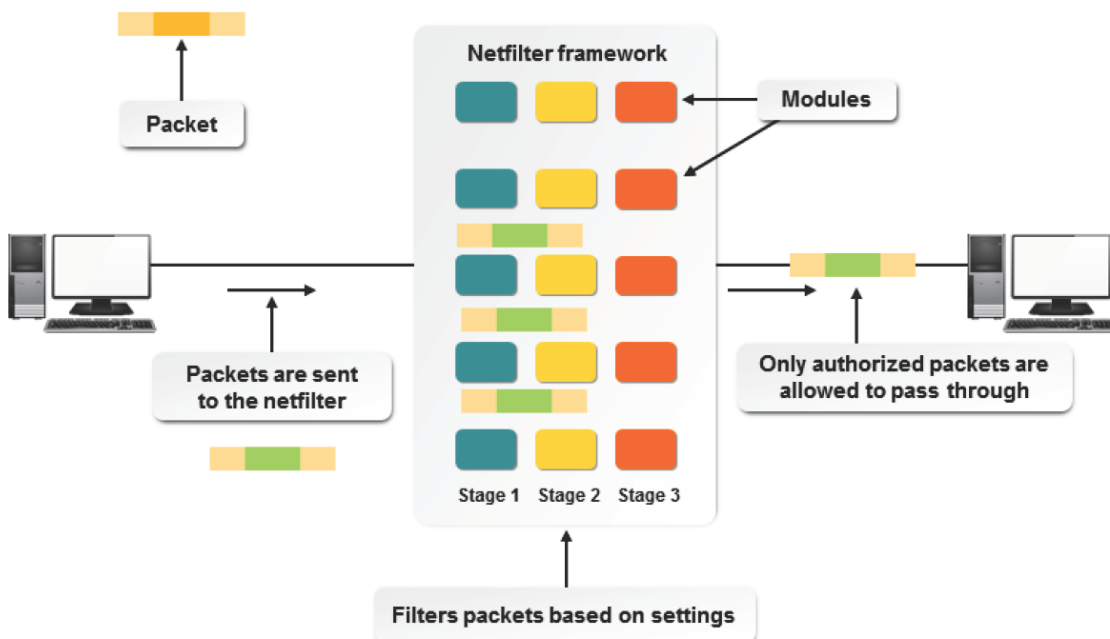


Figure D-2: The working of the netfilter framework.

Proxy Server Implementation

A proxy server, also known as an application gateway, is a much more secure and flexible firewall solution than a pure packet filter. The proxy software can be configured to intercept network traffic.

The proxy recognizes the request and sends the request to the server. In this way, the internal client never connects directly to the external server; thus, the proxy functions as the intermediary, communicating to both the

client and the server. The major advantage is that the proxy software can permit or deny traffic based on the actual data in the packet, and not simply the header.

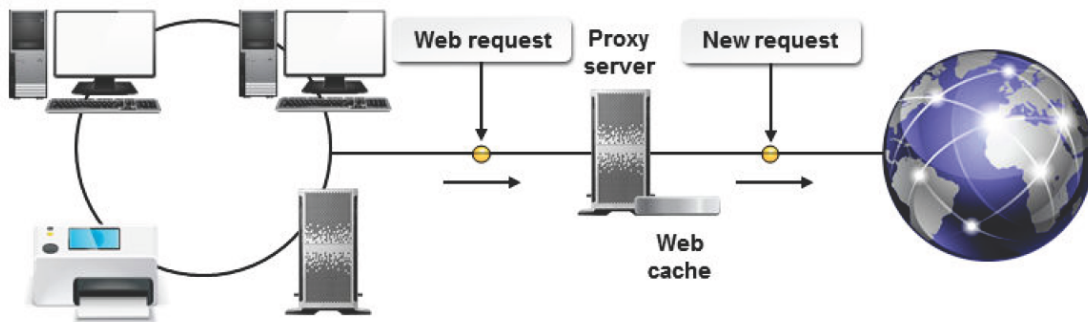


Figure D-3: The functioning of a proxy server on a network.

The iptables Program

The *iptables* program is a firewall program in Linux that provides protection to the internal network. This program uses rule sets, called *chains*, to implement *IP filtering*. Filter, nat, and mangle are the three iptables.

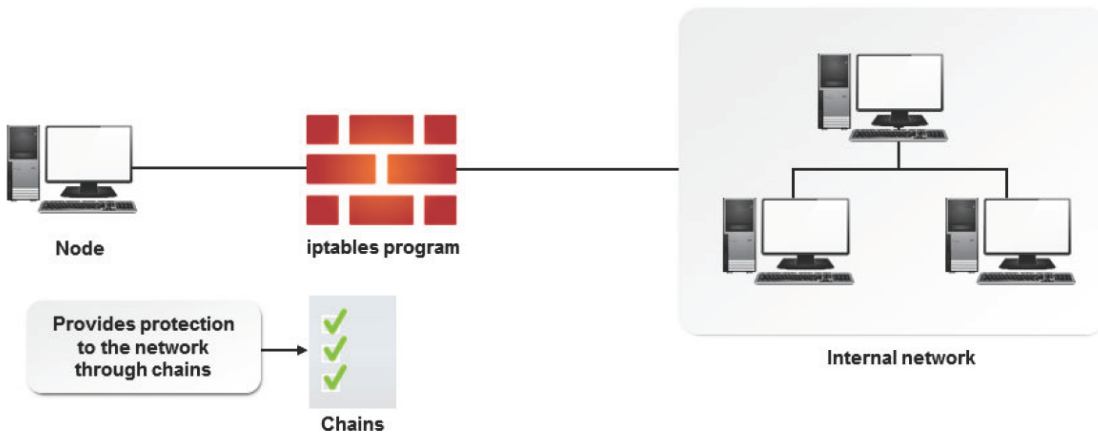


Figure D-4: Protecting a network through chains defined using the iptables program.

Syntax

The syntax of the iptables program is `iptables [-t table] {commands} {chain/rule specification} [options/parameters]`.

The ipchains Program

The *ipchains* program is a Linux tool for managing packet filtering on a Linux server. Support for ipchains is compiled directly into the Linux kernel, where the ipchains tool inserts and deletes rules from iptables. These rules define whether packets are permitted or denied from being sent or received by a Linux system.



Figure D-5: A list of all the rules in the ipchains.

iptables and ipchains

One of the differences between iptables and ipchains is that iptables can be configured to be a stateful packet filter. With an iptable, the headers within a packet are examined against a known set of rules (the chain) in sequence. If the packet matches a rule, a decision is made for that packet based on what is specified (the target). If a match is not found, then the packet is examined against the next rule in the sequence. This continues until all the rules are exhausted. At this point, iptables look to ipchains to make the default policy decision.

Each iptable contains specific chains. The filter table contains the INPUT, OUTPUT, and FORWARD chains. The nat table contains the PREROUTING, POSTROUTING, and OUTPUT chains, and is used on networks where only outgoing packets have to be filtered. The mangle table contains the INPUT, OUTPUT, FORWARD, PREROUTING, and POSTROUTING chains, and is used on large networks where both incoming and outgoing packets have to be filtered.

Syntax

To check your installation for ipchains, use the `rpm -q iptables` command.

Use of ipchains

ipchains are used to block access to privileged ports of a Linux server. By blocking access to all incoming traffic, an administrator can prevent network access to a Linux workstation connected to a network.

How to Implement iptables

Follow these general steps to implement iptables.

Configure iptables

To configure iptables:

1. Log in as **root**.
2. To start the iptables service, enter `service iptables start`.
3. To automatically start the iptables service at the system startup, enter `chkconfig iptables on`.

4. Manage the rules using iptables.

- To add a rule, enter iptables -A forward -p icmp -j ACCEPT.
- To remove a rule, enter iptables -D forward -p icmp -j ACCEPT.

TOPIC B Security Auditing

In the previous topic, you implemented iptables to provide firewall functionality on a Linux system by packet filtering. You will need to set up auditing for files and authentication in order to identify and trace possible security breaches. In this topic, you will implement security auditing.

A network may contain confidential data to which only selected people have access. This data may include financial reports, budgetary information, or personnel reviews. It is your responsibility to monitor the permissions and security levels on this data, so wouldn't you like to know if intruders are attempting to access data for which they do not have proper permission? When you implement security auditing, you can attain instant access to detect and record security-related events on your network.

How to Implement Security Auditing

Follow these general steps to implement security auditing.

Implement Security Auditing

To implement security auditing:

1. Set iptables to send possible intrusions to the log files by using the iptables command.
2. To save the iptables rule, enter the service iptables save command.
3. Direct the output from the iptables output to the **/var/log/iptables** file.

TOPIC C Identify an Intrusion Detection System

In the last topic, you implemented security auditing. To secure files that contain confidential data, you need to constantly monitor the network. In this topic, you will describe the Intrusion Detection System (IDS).

If your network is connected to the public network, chances are good that intruders may trace packets on the network and misuse services on your system. As a Linux administrator, you need to be aware of services on your network that will monitor any incoming and outgoing packets, and send an alert if any of the files or services have been corrupted or manipulated.

Network Monitoring Utilities

Various utilities are available for monitoring network devices and the systems connected to them.

<i>Utility</i>	<i>Used To</i>
nmap	Scan entire networks for various ports and the services running on them along with their status. The nmap utility can also be run as a GUI front end using the nmapfe command. This utility is primarily used to monitor remote network connections. There are a number of options for the nmap utility. These options help a user to get specific information about a network. The syntax of the nmap utility is nmap [scan type] [options] {target specification}.

<i>Utility</i>	<i>Used To</i>
tcpdump	Obtain packet information from a query string sent to the network interface. If the packet header matches the expression in the query, the packets are returned to the user. The syntax of the tcpdump utility is <code>tcpdump [options] {expression}</code> .
wireshark	Obtain packet information. It is a GUI-based utility. On running the wireshark command, the Wireshark Network Analyzer tool is displayed. Its functions are similar to the tcpdump utility.

The IDS

The **Intrusion Detection System (IDS)** is a security sensor that protects portals, networks, and files from hackers. The IDS monitors system files, log files, and packets passed on the network. It also checks the network data stream for unauthorized signatures and attacks. IDS software can also analyze data and alert administrators to potential security problems. An IDS can comprise a variety of hardware sensors, intrusion detection software, and IDS management software.

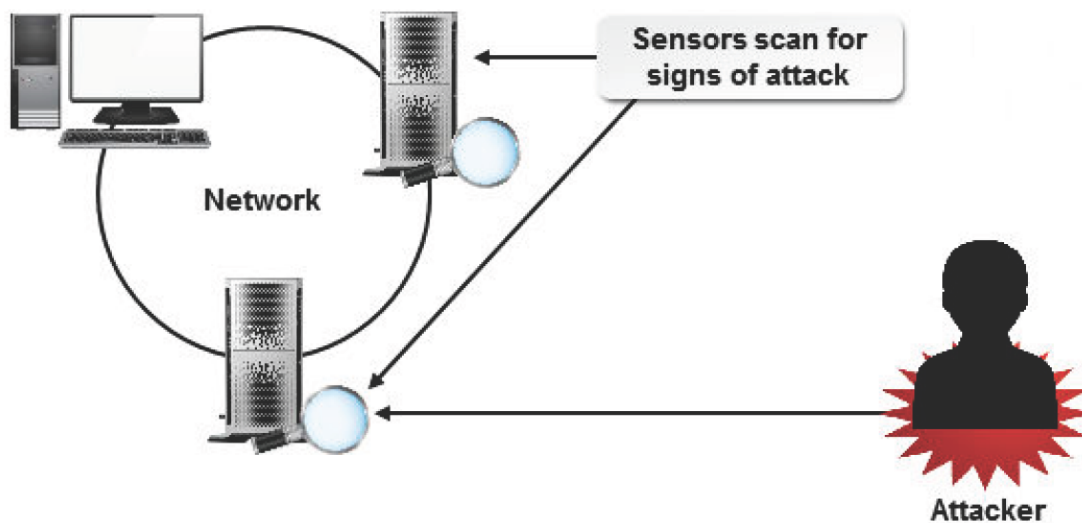


Figure D-6: An IDS protecting a network from a hacker.

Snort

Snort is an IDS that monitors each packet on the network. If a suspicious packet is detected, it passes an alert to the **syslog** file.

Passive and Active IDS

An IDS can be either passive or active. A **passive IDS** detects potential security breaches, logs the activity, and alerts security personnel. An **active IDS** does the same, and then takes the appropriate action to block a user from the suspicious activity. Some people consider the active IDS as a type of Intrusion Prevention System and not as a separate prevention system.

Tripwire

Tripwire is an intrusion detection tool that compares the content of a file or directory with a database that contains the locations of the file or directory and the dates modified. It also monitors the attributes of the file such as the binary signature, size, or expected change of size. If the file has been changed, then the Tripwire tool passes an alert to the administrator by email.

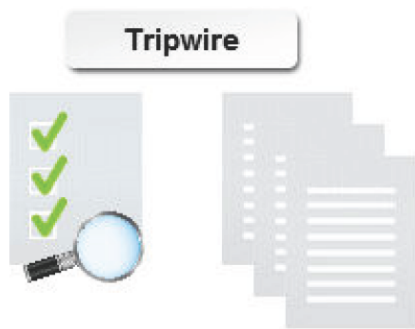


Figure D-7: Tripwire detects intrusion by comparing the content of a file and monitoring its attributes.

The Tripwire Database

The **Tripwire database** contains baselines, which are snapshots of files and directories noted at a specific time. When usage becomes abnormal, problems can be easily detected by comparing the latest files and directories with the earlier snapshots. After a Tripwire is installed and configured, the database should be initialized. To initialize the Tripwire database, execute the following command: `/usr/sbin/tripwire --init`. While initializing the database, Tripwire creates a collection of filesystem objects based on certain specifications mentioned in the policy file. An integrity check should be done after initializing the database. To view the entire database, enter the

command `/usr/sbin/twprint -m d --print-dbfile | less`.

The tw.cfg File

The **tw.cfg** file is the Tripwire configuration file. Before generating the **tw.cfg** file, you can modify the configuration options in the **twcfg.txt** file to suit your requirements. The following table lists some of the configuration options.

<i>Option</i>	<i>Used To</i>
POLFILE	Modify the location of the policy file, tw.pol .
DBFILE	Modify the location of the database file, [hostname].twd .
REPORTFILE	Modify the location of the report file, [hostname].[date].twr .
SITEKEYFILE	Modify the location of the site key file.
LOCALKEYFILE	Modify the location of the local key file.
EDITOR	Specify the editor called by Tripwire when the database is updated.
MAILPROGRAM	Specify the mail program used by Tripwire.
MAILMETHOD	Specify the mail protocol used by Tripwire.



Note: You cannot create the **tw.cfg** file if the Tripwire file locations are not properly specified in the **twcfg.txt** file.

Snort

Snort is a network IDS that monitors network traffic. Snort monitors each packet on the network.

If a suspicious packet is detected, it passes an alert to the **syslog** file and notifies the administrator through email or a pop-up window. It detects various network attack methods, including CGI attacks, denial-of-service, buffer overflow, and SMB probes.

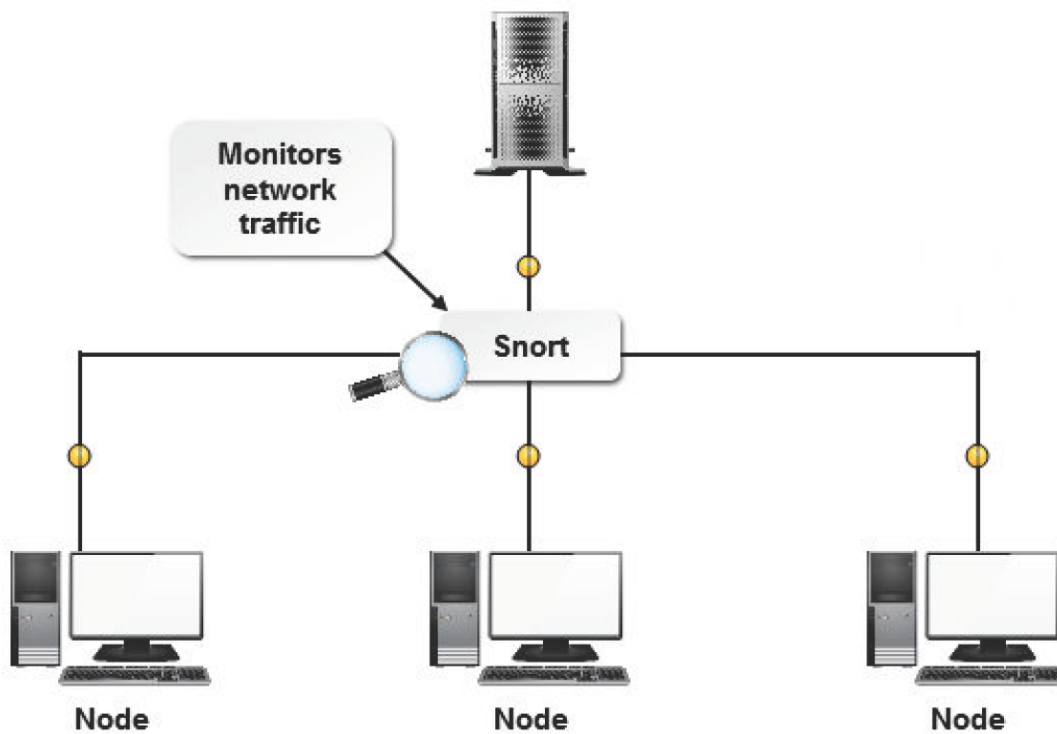


Figure D-8: Snort monitoring traffic on a network.

Snort Options

Some frequently used snort command options are listed in the table.

<i>Option</i>	<i>Used To</i>
-v	Run snort in verbose mode.
-i {interface}	Listen to packets on a specific interface.
-l {directory}	Specify the output logging directory.
-n {number of packets}	Specify the number of packets to be processed before exiting.
-D	Run snort in daemon mode and log all messages in the <code>/var/log/snort/alert</code> directory.

Portsentry

Portsentry is an IDS that detects and responds to port attacks. It is run as a daemon on TCP and UDP sockets to detect port scans on the system. If a port attack is detected, portsentry generates a log entry that contains the details of the hostname, the time of attack, the attacking host's IP address, and the TCP or the UDP port. It reports to the syslog daemon and alerts the administrator through email.



Note: Portsentry is generally used to drop the route to the scanning host. This prevents the attacking host from using any information it gained from the port scan.

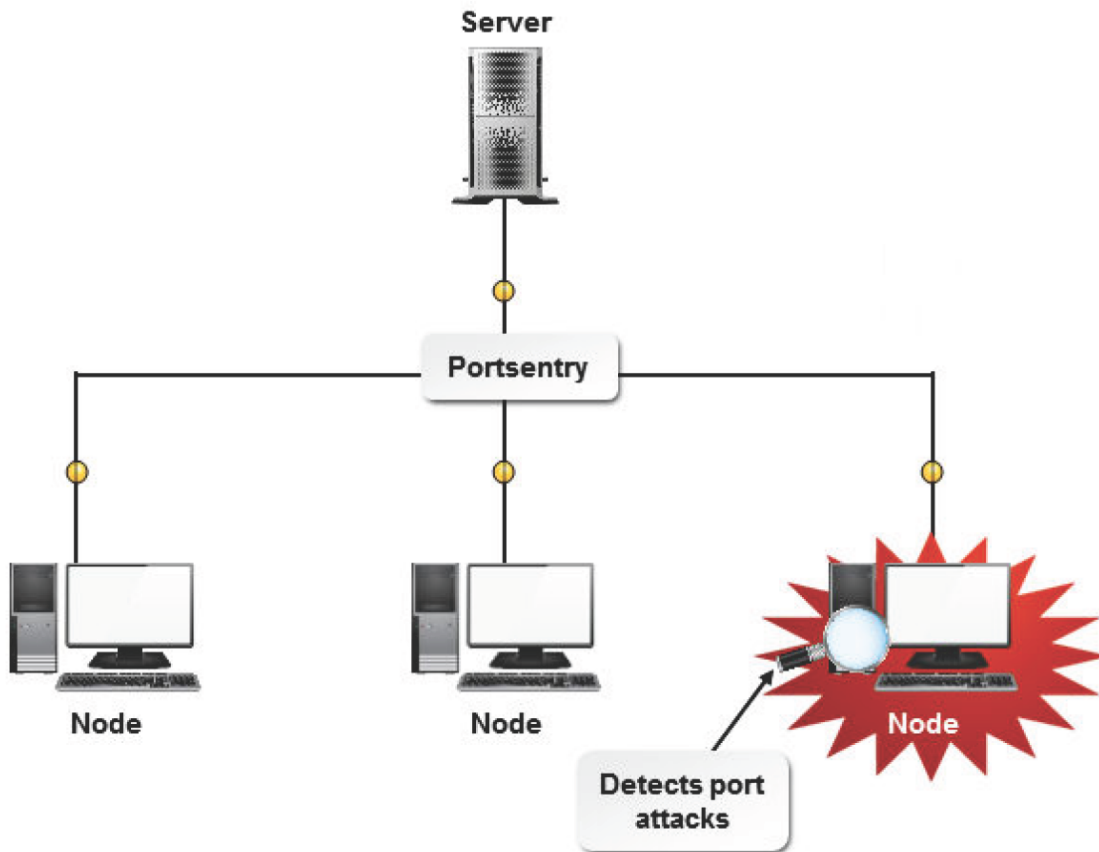


Figure D-9: Portentry detecting intrusion on a network.

Nessus

Nessus is an IDS that audits the security of remote hosts and the services running on the network.

Nessus performs vulnerability checks on the network and generates reports, listing all the security flaws and the possible ways to counter them. It consists of two parts, a server and a client. The server uses the `nessusd` daemon to maintain a vulnerability database for implementing security checks. The client uses this database and performs the security checks.

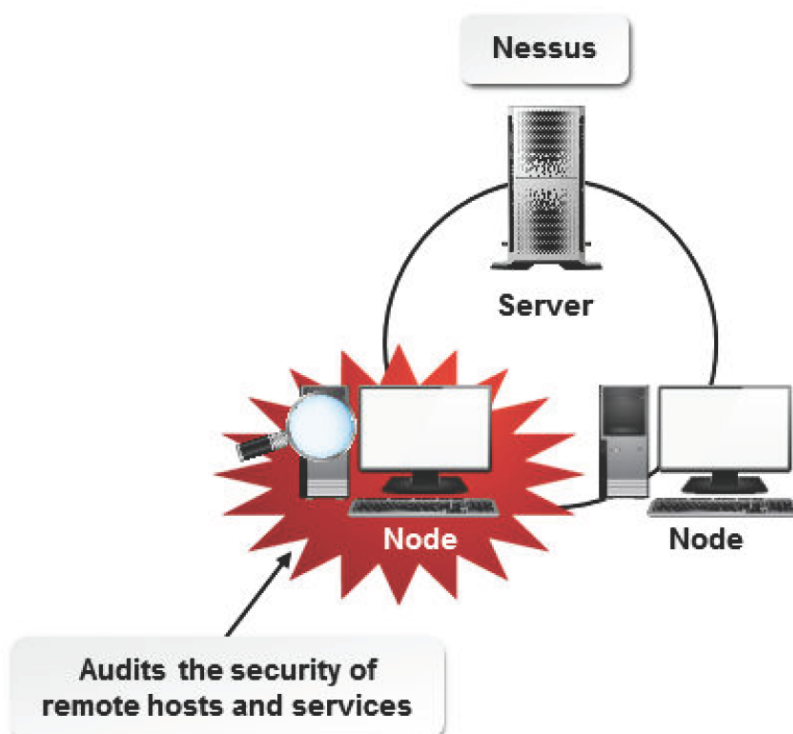


Figure D-10: Nessus auditing security on a network.

Solutions

ACTIVITY 1-1: Performing Basic Linux Tasks Review

1. What are the advantages of open source software over licensed software?

A: Answers will vary, but may include open source software enables users to access its source code and gives them the right to modify it. This ensures continuous improvement in the functionality of the software and increases the likelihood of bug detection.

2. What are the advantages of using Linux?

A: Answers will vary, but may include low cost, ease of licensing and customizing, and stability.

ACTIVITY 2-1: Managing User and Group Accounts Review

1. How is organizing users into groups useful to you?

A: Answers will vary, but may include organizing users into groups is useful because group accounts provide features such as collective access rights, collective ownership of files, and sharing of resources. This facilitates better management of user accounts.

2. Why is it essential to configure a user profile?

A: Answers will vary, but may include a user profile helps in uniquely identifying the user. It provides the user with a distinct identity and helps in differentiating one user from another.

ACTIVITY 3-1: Managing Partitions and the Linux Filesystem Review

1. When do you think formatting a partition is necessary? Why?

A: Answers will vary, but most frequently administrators format a partition when they add a new disk storage device to a system.

2. What filesystem types have you worked with? What advantages and disadvantages have you encountered with each?

A: Answers will vary, but may include a response in favor of the ext4 filesystem because it is an improved version of ext3 and ext2 and is the native Linux filesystem now. Some students may note that XFS has been selected as the new default filesystem for CentOS/ RHEL 7 for its speed, reliability, and advanced journaling features. Students may be familiar with older types like ext2 and their slower recovery time, whereas other students may have experienced that ext4 recovers faster than ext2. Some may note that formatting a new partition with XFS is orders of magnitude faster than formatting a partition with ext4.

ACTIVITY 4-1: Managing Files in Linux Review

1. Which tools will you use to search for files on your Linux system?

Why?

A: Answers will vary, but may include by using commands such as find, locate, and grep.

2. Which text editor in Linux do you prefer? Why?

A: Answers will vary, but may include nano—for its ease of learning, Vim—for its flexibility, and gedit—for its ease of use in the GUI.

ACTIVITY 5-1: Managing Linux Permissions and Ownership Review

1. What methods do you think you might use to preserve confidentiality of information on Linux systems?

A: Answers will vary, but may include confidentiality of information can be preserved by granting permissions at different levels, enabling access for different categories of users, and modifying ownership.

2. In what situation would you need to modify the default permissions?

A: Answers will vary, but may include default permissions should be modified when files require higher or lower levels of security.

ACTIVITY 6-1: Printing Files Review

1. Why would it be beneficial to have a local printer vs. a network printer?

A: Answers will vary, but may include a local printer can be accessed only from a single system to which it is connected, and often a local printer is more secure when output may contain confidential or sensitive information. On the other hand, a network printer is managed by a print server and can be accessed by many systems on the network and provide additional features and benefits not cost-effective for a local printer.

2. What do you think the best way to manage print queues will be for your organization?

A: Answers will vary, but may include print queues allow users to prioritize the printing of files and customize the order of the files to be printed, and may include the use of the lpq, lpstat, lpc, and lprm commands.

ACTIVITY 7-1: Managing Packages Review

1. What packages might you need to install in your organization? Which package management tools will you use most often?

A: Answers will vary, but may include packages such as yum, web servers, file servers, and databases.

Package management tools may include rpm, yum, dpkg, or comments about manually installing software via Makefiles and source code.

2. Why do you think it is important to create your own repositories?

A: Answers will vary, but may include by using repositories you can maintain a personalized and customized database of packages and remove any additional package. Private repositories enable effective package organization. Repositories can be created and organized according to data such as package name and source name. Local repositories may provide a mirror of the public repositories for faster access and increased efficiency.

ACTIVITY 8-1: Managing Kernel Services Review

1. **Why do modules affect the way a kernel is loaded? Do you expect that you will need to load any special modules into the kernel to fulfill your organization's requirements?**

A: Answers will vary, but may include kernel modules increase the flexibility of the kernel by extending kernel functionality to control hardware devices, drivers, and filesystems, and decrease the kernel's load by lowering the number of processes the kernel is involved in. Specific organization requirements that may require kernel modules to be loaded include special hardware device support or compatibility requirements.

2. **Why is process management important for operating systems?**

A: Answers will vary, but may include process management is necessary to manage system resources.

Too many unnecessary processes running on the system at the same time will result in poor system performance. In addition, process management enables the system administrator to identify harmful processes and prevent system corruption.

ACTIVITY 9-1: Working with the Bash Shell and Shell Scripts Review

1. **When might you use variables in your own shell scripts?**

A: Answers will vary, but may include variables are used to define changeable values, read user-defined input, and customize the shell environment.

2. **What are the various tasks that you might perform in your environment by running a shell script?**

A: Answers will vary, but may include setting reminders, sending mail, and customizing the system environment. Shell scripting is one of the more powerful tools in a Linux system administrator's toolbox as it enables the automation of many common processes and procedures.

ACTIVITY 10-1: Managing Jobs and Processes Review

1. **Which of the process management tools presented do you expect will be useful in your environment? Why?**

A: Answers will vary, but may include a system's performance can be improved by prioritizing processes, rescheduling processes, and killing unwanted processes via commands such as ps, kill, renice, top, nohup, and cron/crontab. Advanced users may have experience with these commands and/or using screen for system administration.

2. **Do you think automating system processes affect a system's performance? Why?**

A: Answers will vary, but may include the performance of a system when processes are automated depends on the processes themselves. When high-performance processes are automated, they will utilize more system resources than low-performance processes.

The system's performance also depends on the number of processes running on the system at a particular time and also the priority of processes.

ACTIVITY 11-1: Managing System Services Review

1. **How will you use system logs to troubleshoot system problems?**

A: Answers will vary, but may include when you are maintaining systems, you will always encounter issues that creep up suddenly. When the issues are left unattended, they could turn into major problems. System

logs collect data related to general system behavior that can be used to identify the problems.

2. Which level of access control and/or SELinux would you use in your organization and why?

A: Answers will vary, but students within highly-regulated environments like financial services and banking will likely use the highest level of SELinux, Enforcing Strict. Users in less security-focused environments may find that Discretionary Access Control (DAC) methods, which on SELinux are used in the Disabled or Permissive modes, are sufficient for their environment.

ACTIVITY 12-1: Configuring Network Services Review

1. What networking schemes are in use in your organization, and based on what you've learned in this lesson would you change anything?

A: Answers will vary, but may include Class-A, -B, or -C networks, CIDR for small subnetworks, DHCP, DNS, SSH, VNC, and rdesktop protocols.

2. When you need to securely access remote systems, which tools will you use and why?

A: Answers will vary, but may include the size of the network and the number of servers and clients must be taken into account. Most modern system administrators use SSH for remote Linux server access, and they may tunnel other protocols over SSH for security. VNC and rdesktop are frequently used to manage remote Windows and other GUI servers.

ACTIVITY 13-1: Configuring Basic Internet Services Review

1. Which mail retrieval protocol would you configure in your organization? Why?

A: Answers will vary, but may include configuring IMAP will retain a copy of messages on a server until it is deleted by the user. This helps you store an updated copy of messages on the server.

2. What services might you configure using Systemd in your environment?

A: Answers will vary, but may include email, file transfer, and web services, as well as custom applications.

ACTIVITY 14-1: Securing Linux Review

1. Which encryption method will you use to secure data? Why?

A: Answers will vary, but may include the encryption method to be used depends on the type and sensitivity of data to be transferred and the mode of communication. If data is to be transferred over the Internet, appropriate encryption methods, such as keys, must be used.

2. What password policies do you have currently in your organization and how do you think they can be improved?

A: Answers will vary, but may include requiring minimum password lengths; specific types of characters such as numbers, upper AND lowercase letters, and special characters; and requiring that passwords be changed on a scheduled basis.

ACTIVITY 15-1: Managing Hardware Review

1. Which of the hardware management tools do you expect you may use in your organization?

A: Answers will vary, but will likely include using USB devices and the df and du commands to check disk utilization.

2. Do you expect that you will implement disk quotas in your environment? Why or why not?

A: Answers will vary, but may include assigning disk quotas makes partition management easier because disks are split into manageable chunks. Disk quotas help avoid data overflow on a shared file server by issuing alerts if a user has used up his or her disk space.

ACTIVITY 16-1: Troubleshooting Linux Systems Review

1. How does troubleshooting in Linux differ from the troubleshooting approach you've taken with other systems?

A: Answers will vary, but may include troubleshooting via logs and kernel messages, using new tools such as lsusb and ifconfig, or the differences between online help resources for Linux and other operating systems.

2. Provide an example of a recent problem you encountered in your environment and how you were able to resolve it.

A: Answers will vary, but this is a great jumping off point to discuss some interesting and peculiar issues that can occur on networks, especially for new system administrators.

ACTIVITY 17-1: Installing Linux Review

1. How does the boot process affect the applications installed on a system? Why?

A: Answers will vary, but may include the boot process is responsible for the proper loading of the operating system and various applications that are installed on the computer. Therefore, any change in the boot process will affect the programs and applications that are dependent on it. Once the boot process is complete, the applications may not be dependent on the boot configuration because the applications start their own individual processes and may not be affected unless they are configured to start on system startup.

2. Which is the best runlevel for you to boot a system in your organization? Why?

A: Answers will vary, but may include the best runlevel to boot a system depends on the user's requirements. The default runlevel is runlevel 5 (graphical.target) when the **Server with GUI** option is selected during installation. If the user is technically sound or the purpose of the Linux server does not require local workstation GUI access, he or she may use runlevel 3 (multi-user.target) with the CLI only. If the user prefers the GUI, he or she may use runlevel 5 (graphical.target).

ACTIVITY 18-1: Configuring the GUI Review

1. Do you think using the Linux GUI in conjunction with the CLI will yield better results? Why?

A: Answers will vary, but may include using the GUI or the CLI depends on users' requirements. If users are technically sound and are able to use commands with ease, then they can use the CLI. If they want a more interactive interface, they can use the GUI. Therefore, combining both will make Linux accessible to all users.

2. In what way do you think customizing window managers is useful?

A: Answers will vary, but may include customizing window managers enables users to easily access frequently used applications. Users can modify the system to suit their requirements and comfort.

Glossary

.bash_history

The hidden file located in your home directory that contains a chronological listing of your last 500 shell commands.

.bashrc

The hidden file located in your home directory that contains commands that Bash reads and executes when an interactive shell that is not a login shell is started.

/etc/default/grub

A configuration file that sets the default GRUB menu settings.

/etc/group

A file that contains a list of groups.

/etc/grub.d

A directory that holds GRUB scripts.

/etc/issue

The login banner that is displayed to local users.

/etc/issue.net

The login banner that users see when they make a network connection, such as Telnet or SSH, with the system.

/etc/syslog.conf

The file that controls the location where syslogd records system logs.

3DES

A block cipher algorithm that can encrypt and decrypt data using a secret key.

absolute path

The specific location, including the domain name, irrespective of the working directory or combined paths.

ACL

(Access Control List) A list of permissions attached to an object.

active IDS

An IDS that detects a security breach according to the parameters it has been configured with, logs the activity, and then takes appropriate action to block the user from the suspicious activity.

activity light

A status indicator light that flickers when network packets are sent or received.

algorithm

A procedure or formula for solving a problem using a finite set of well-defined instructions for accomplishing some task that, given an initial state, will terminate in a corresponding recognizable end state.

alias

A command that is used to generate command line aliases. It is shorthand for a longer expression.

Anaconda installer

An installation program for installing Linux through text or graphical mode.

anacron

A daemon that executes commands at intervals, which are specified in days, without requiring the system to be running continuously.

application access controls

The daemons that are used to restrict access to certain important applications.

archiving

A method of storing data for later use by copying data from a system disk drive into a backup device.

argument

An argument is usually a file name or directory name that indicates the files on which the command will operate.

ARP

(Address Resolution Protocol) A network protocol that is used by IP to map network addresses to MAC addresses.

aspell

A utility that functions as a spell checker in Linux.

asymmetric encryption

An encryption type where a key pair is used for encryption.

at

A command that is used to execute a given set of commands only once, at a specified time.

ATAPI

(AT Attachment Packet Interface) A protocol that is used to control mass storage devices.

authentication

A process that verifies the identity of users.

automatic rotation

A system of regular rotation of logs to maintain a minimum log file size.

average time

See *true time*.

awk

A command that performs pattern matching.

background process

A program that allows the Linux shell to execute a command that runs a job in the background, enabling processes to run simultaneously.

Bash shell

(Bourne-Again SHell) A default shell in Linux that facilitates command line editing, command history, and shell scripting.

BIOS

(Basic Input/Output System) Low-level software that acts as the interface between the hardware and the operating system on a computer.

block special files

Large files that are used for data storage.

Blowfish

A symmetric block cipher that provides strong encryption and uses key sizes up to 56 bytes.

boot disk

A disk that contains the operating system files to start up a system.

boot loader

A program that loads the kernel from a hard drive, or boot disk, and then starts the operating system.

boot manager

See boot loader.

boot process

The process of starting or restarting your computer by loading the operating system from your hard drive.

broadcast address

A special IP address that is used to send messages to all hosts with the same network address.

browser mode

The mode in the Nautilus browser that enables you to display the selected folder in the same window.

cd

A command that is used to change directories.

CentOS

The CentOS Linux distribution is a stable, predictable, manageable and reproducible Linux Distribution derived from the sources of Red Hat Enterprise Linux (RHEL).

central network log server

A server that is used to implement centralized logging services.

Certificate Authority certificates

The digital certificates generated by a common and trusted Certificate Authority (CA) on receiving a certificate signature request (csr).

chains

A set of rules used by iptables to handle IP filtering.

CHAP

(Challenge Handshake Authentication Protocol) A security authentication protocol that encrypts the user name and password information using a key and transmits them over the network.

character special files

Small files that are used for streaming of data.

chattr

A command that is used to change the attributes of a file.

check for dependency

The first stage in the Debian Archive Package Installation process. In this stage, the package manager checks the specified Debian archive package for the numerous dependencies required.

child process

A process started by another process.

CIDR

(Classless Inter-Domain Routing) A method for allocating IP addresses that replaces the original addressing architecture of classful network design.

Class A

A subnet scheme that provides 16,777,214 nodes per network.

Class B

A subnet scheme that provides 65,534 nodes per network.

Class C

A subnet scheme that provides 254 nodes per network.

CLI

(Command Line Interface) A textual interface based on the operating system, where a user typically enters commands at the command prompt to instruct the computer to perform a specific task.

clock drift

The gradual variation in time that sets between the hardware clock and the system clock.

CMOS

(Complementary Metal Oxide Semiconductor) Pronounced as "see-moss." The most widely used type of integrated circuit for digital processors and memories. Virtually everything is configured through CMOS today.

command line interpreter

A program that implements the commands entered in a text interface.

command mode

A mode in Linux that allows users to perform different editing actions using single keystrokes.

command prompt

A sequence of one or more characters in a CLI that is used to indicate the interpreter's readiness to accept commands.

command substitution

The ability to reassign the output of a command as an argument to another command.

configure

The last stage in the Debian Archive Package Installation process. In this stage, the unpacked files can be configured with the default or customized values to suit your requirements.

console

See *terminal*.

control statement

An instruction that determines the direction a program takes depending on a condition.

copyleft

A concept that emphasizes the enforcement of public ownership of creative works.

count

A number that multiplies the effect of keystrokes in Vim.

cp

A command that is used to copy files.

cpio

A command that copies files to and from archives.

createrepo

A command that is used to create yum repositories.

cron

A daemon that runs in the background and executes specified tasks at a designated time or date.

cron job

A task scheduled via cron.

crontab

The file that contains instructions defining the tasks to be executed by a cron.

cryptographic hashes

An encryption method where arbitrary data is encapsulated within a fingerprint, which is a fixed string called the hash value, checksum, or message digest.

CUPS

(Common UNIX Printing System) A print management system designed for scheduling print jobs, processing administrative commands, and providing printer status information to local and remote programs.

cylinder

The aggregate of all tracks that reside in the same location on every disk surface.

daemon

A program that runs in the background without the need for human intervention.

database

An organized collection of information to facilitate easy storage and retrieval of data.

datagram

See packet.

dd

A command that allows you copy and convert files such that the converted files can be transferred to another type of media.

default gateway

A gateway that acts as a network segment's access point to all other external networks and the Internet.

default gateway address

The IP address assigned to the default gateway router.

delayed job

A job that can be run at some specified time after you issue the command.

DES

(Data Encryption Standard) An encryption method developed by IBM in 1977, which uses a 56-bit private key applied to each 64-bit block of data.

detached job

A job that can be set to run after you log out of the system.

device driver

A software program that enables a computer's operating system to identify the characteristics and functions of a hardware device, communicate with it, and control it.

device management layer

A layer in the kernel that manages devices by controlling device access and interfacing between user applications and hardware devices of the computer.

device node

An access point to the device drivers that is used while mapping service requests with device access.

device tree

A structure that lists all hardware devices installed on the computer and assigns device nodes to them. It is auto-generated by the computer's operating system.

df

A command to view free disk space.

DHCP

(Dynamic Host Control Protocol) A server that hands out IP addresses on an as-needed basis.

diff

A command that is used to compare individual text files or contents of directories.

directory service

A software system that stores and organizes information in a directory.

disk image

See *ISO image*.

disk quota

The specific amount of disk space that is allotted to a user for file storage on a computer.

display manager

A program that controls the look and feel of a desktop environment.

distribution

A collection of software based around the Linux kernel and a specific package management system that provides a complete operating system for installation.

distro

Another word for "distribution", see *distribution*.

DMA

(Direct Memory Address) A method by which hardware devices directly communicate with the memory to obtain memory allocation without going through the processor.

DNS

(Domain Name System) A distributed, hierarchical database system that maintains information about hostnames and their equivalent IP addresses on the Internet.

DNS resource record

A record that defines some parameters for a zone.

domain

A node in the hierarchical structure of the data stored in DNS.

domain name

A label given to a domain.

driver

A program that controls a device attached to a computer.

du

A command to view disk usage, including the size of directory trees and files.

dump

A command that dumps all the files in a filesystem into a tape or another file.

dumpe2fs

A utility that is primarily used for managing extended (ext2, ext3, and ext4) filesystems.

editing operators

The tools in command mode that can be used to manipulate text using simple keystrokes.

ELILO

A boot loader for UEFI machines. It supports flexible local booting from a FAT-32 filesystem.

Emacs

A flexible, powerful, and popular text editor used in Linux and UNIX.

encryption

The process where user access to information is controlled by configuring the data to appear in the form of codes that cannot be interpreted by unauthorized users.

entropy

A measure of randomness collected by a device, application, or system.

environment variable

A storage location in the environment of the operating system's command shell.

exec

A bash shell scripting command used to execute another command, replacing the current shell process with this new program's process (no new process is created).

execute mode

A mode in Linux that allows users to execute commands within the editor.

ext2

(second extended filesystem) A Linux-based filesystem that replaced the extended filesystem (ext).

ext3

(third extended filesystem) A Linux-based filesystem that has improved data recovery speed and integrity compared to ext2.

ext4

(fourth extended filesystem) A Linux-based filesystem that has improved journaling and increased file size support.

extended partition

A partition that does not contain any data and has a separate partition table.

FAT

(File Allocation Table) A filesystem compatible with multiple operating systems, including all versions of Windows, MS-DOS, and UNIX.

fdisk

A utility program that is used for creating, modifying, or deleting partitions on a disk drive.

FHS

(Filesystem Hierarchy Standard) A collaborative document that specifies a set of guidelines for the names of files and directories and their locations.

fields

Small segments of organized data in a database.

file owner

A user who creates a file or directory.

filesystem

A method used by an operating system to store, retrieve, organize, and manage files and directories in various mass storage devices.

filesystem integrity

The correctness and validity of a filesystem.

filesystem management layer

A layer in the kernel that manages the filesystem, which involves storing, organizing, and tracking files and data on a computer.

filter

A program that accepts an input or output request, verifies the data that matches the criterion specified in the request, and then processes it.

find

A command that is used to search a specific location for files and directories.

FIPS

(First nondestructive Interactive Partition Splitting) A utility that is used to resize the FAT partitions.

firewall

A software program or a hardware device that protects a system or a network from unauthorized access by blocking unsolicited traffic.

FireWire

A high-speed serial bus developed by Apple Computer, Inc. and Texas Instruments that allows various devices to be connected with a system. It was originally a trademarked term for IEEE 1394, but is now used interchangeably.

font path

A collection of paths in the filesystem where font files are stored.

for

A loop that executes a part of the script for a specific number of times.

foreground process

The program that the user is interacting with currently.

forward zone

A DNS zone that is used for mapping hostnames to IP addresses.

FQDN

(Fully Qualified Domain Name) A method by which systems are uniquely identified on the worldwide network.

fsck

A command that is used to check the integrity of a filesystem.

fstab

A configuration file that stores information about storage devices and partitions and where and how the partitions should be mounted.

FTP

(File Transfer Protocol) A protocol that is used to send and receive files from one system to another through the Internet.

function

A subprogram that executes an operation and returns a value on completion of the operation.

gateway

A device, software application, or system that converts data between incompatible systems.

GCC

(GNU Compiler Collection) A compiler system that supports various programming languages.

gdisk

(GPT fdisk) A command-line utility used to manipulate disk partitions that use the newer Globally Unique Identifier (GUID) Partition Table (GPT) format.

GDM

(GNOME Display Manager) The default display manager for Red Hat Linux.

gedit

A simple yet powerful GUI-based text editor used in the GNOME desktop.

getent

A utility used to display entries from databases supported by the Name Service Switch (nss) libraries.

global user profile

A description of the settings, preferences, bookmarks, stored messages, attributes, permissions, and other user items that users have access to, on whichever system they log in to.

globbing

A function that expands file names using a pattern-matching behavior.

GMT

(Greenwich Mean Time) The time at the prime meridian at Greenwich, England.

GNOME search tool

A graphical utility used for searching files.

GNOME system monitor

A graphical utility that is used to monitor the system processes, resources, and filesystems.

GNU Parted

A utility that can be used to create, destroy, and resize partitions.

GNU project

(GNU's Not UNIX) A project started by Richard Stallman to create a comprehensive computer operating system composed entirely of free software.

grace period

The time limit before the soft limit is enforced for a filesystem.

group

A collection of system users having the same access rights.

group database

Refers to the file named `/etc/group`, which contains a list of groups, each on a separate line.

GRUB 2

The newest version of the GRand Unified Bootloader (GRUB).

grub-install

A utility that installs the GRUB 2 bootloader in the MBR of the specified partition.

grub-mkconfig

A utility that generates `grub.cfg` files to configure the GRUB 2 bootloader.

grub2-install

See *grub-install*.

grub2-mkconfig

See *grub-mkconfig*.

GUI

(Graphical User Interface) A collection of icons, windows, and other graphical images on screen that help users interact with the operating system.

Gvim

The graphical version of the Vim editor.

gzip

(GNU zip) A compression utility that reduces the size of the named files.

HAL

(Hardware Abstraction Layer) A logical interface that enables software applications of a system to interact with hardware devices at an abstract level through system calls.

hard limit

The absolute limit on disk usage that a quota user has on a partition.

hard link

A reference to another file; it allows the file's data to have more than one name in different locations in the same filesystem.

history

A command that is used to view previously typed commands.

home directory

A directory where you are placed when you log in to a system.

honeypot

A system designed to attract attackers.

host computer

A computer that consists of two layers and controls data transfer to and from USB devices.

host controller hardware layer

A layer that converts data between the format used by the host computer and the physical format used by the USB. It is also known as the adapter layer.

HTTP

(Hyper Text Transfer Protocol) A protocol that is used to transfer hypertext files across the World Wide Web.

I/O address

(Input/Output address) An address that is used to identify the requests sent to or from a hardware device.

ICMP

(Internet Control Message Protocol) A protocol that is used to handle error and control messages.

IDS

(Intrusion Detection System) A security sensor that protects portals, networks, and files.

ifconfig

A command that is used for configuring network interfaces for Linux servers and workstations.

IMAP

(Internet Message Access Protocol) A protocol that is used to retrieve email messages over the TCP connection on port 143.

immutable flag

An extended attribute of a file that prevents the file from being modified.

index node table

A data structure that contains information about individual files in a filesystem.

inetd

A system service daemon that starts programs needed for accessing different Internet services.

init

The parent of all processes. It creates processes at system boot time from the /etc/inittab file.

initrd

The initial ramdisk that is temporarily mounted as the root filesystem for loading the startup programs and modules.

initrd image

An archived file containing all the essential files that are required for booting the operating system.

inittab

A file found in the /etc directory that stores details of various processes related to system initialization.

inode

(index node) A computer's reference for a file.

insert mode

A mode in Linux that allows users to insert text by typing.

insmod

A utility that installs a module into the currently running kernel.

ip

A utility used to configure networking in Linux.

IP address

A unique address that identifies a host on the Internet.

IP class

A block of IP addresses that can be assigned to businesses or governments, based on the size and need.

IP filtering

A mechanism employed for processing, dropping, logging, or forwarding data packets received by a system.

ipchains

A Linux tool for managing packet filtering on a Linux server.

IPng

(IP Next Generation). See *IPv6*.

iptables

A firewall program in Linux that provides protection to the internal network.

IPv4

(IP Version 4) An older version of IP, which is being replaced by IPv6 with extended features.

IPv6

(IP Version 6) A new version of IP, which is being implemented on the Internet. Also called IP Next Generation (IPng).

IRQ

(Interrupt ReQuests) A signal sent by a hardware device to the kernel requesting processing time to perform an operation.

ISO 9660

A filesystem standard designed by the International Organization for Standardization (ISO) for DVDs and other optical media.

ISO image

An archive file format for files that are to be written to optical discs such as CDs and DVDs.

iwconfig

A command that is used for configuring wireless network interfaces for Linux servers and workstations.

JFS

(Journaled File System) A 64-bit filesystem created by IBM that is designed to handle power failures and system crashes efficiently.

jobs table

A table containing information about processes running in the background.

join

A query that is used to combine values in two or more tables in a relational database.

journalctl

A utility that manages log files created by the Journal component of Systemd.

journaling filesystem

A method that is used by an operating system to quickly recover after an unexpected interruption, such as a system crash.

KDM

(K Display Manager) The display manager for KDE or K Desktop Environment.

Kerberos

A network authentication service that is used by client/server applications.

kernel

The central core of the Linux (or UNIX) operating system that manages all the computer's physical devices.

kernel module

A system-level function that extends the functionality of the kernel.

key

A block of information that authorizes users to access data.

klogd

(kernel logging daemon) A daemon that tracks kernel messages by prioritizing them.

KWrite

A flexible GUI-based text editor used in KDE.

LAN

(Local Area Network) A network that connects computers in a small geographical area such as a floor or a building.

lastlog

A command that displays the latest login details of all users.

LDAP

(Lightweight Directory Access Protocol) A communication protocol that defines the transport and format of messages used by a client to access the directory service.

leap second

An adjustment to UTC to account for Earth's irregular rotation.

LED

(Light Emitting Diode) An electrical component that is frequently used as an indicator light on network adapters and other types of network equipment.

libwrap.so.0

The library that is used for compiling a TCP wrapped service.

LightDM

The default display manager for Ubuntu that starts the X servers, user sessions, and login screen.

link light

A status indicator light that indicates whether or not the NIC can receive a signal from the network.

Linux

An open source computer operating system derived from UNIX.

Linux documentation

The material that provides information on various Linux commands and blocks of code.

Linux kernel

The core constituent of the Linux operating system that manages all other resources on the system. It performs functions such as sharing resources and allocating of memory, input and output operations, security settings, and user access.

Linux rescue environment

A stand-alone Linux program for troubleshooting a corrupt Linux installation.

ln

A command that is used to create a link to a file.

load average

The average number of processes waiting to run on the system for the last 1 minute, 5 minutes, and 15 minutes.

local or private repositories

The repositories stored on a system.

locate

A command that performs a quick search for any specified string in file names and paths stored in the mlocate database.

log file

A file that stores log messages.

log file analysis

The process of examining the messages generated by logging daemons in log files.

logging service

A daemon used to track logs or errors that are created in a system or kernel.

logical partition

A partition created within an extended partition.

logrotate

A command that is used to compress, delete, or mail log files.

logwatch

A utility that is used to monitor logs.

loop

A programming construct that supports repetitive execution of one or more statements.

lpc

A command that is used to manage print jobs.

lpr

A command that is used to submit files for printing.

ls

A command that is used to list files in the current working directory.

lsattr

A command that is used to list the attributes of a file.

lsmod

A utility that displays the currently loaded kernel modules, their sizes, usage details, and dependent modules.

LVM

(Logical Volume Manager) A software tool that is used to manage the disk storage on a computer system.

MAC address

(Media Access Control address) See *physical address*.

mail protocol

A method of distributing email messages from a mail server.

mail queue

A waiting area for email messages that need to be processed by a computer.

major number

A number stored as part of the structure of a device node. It identifies the device driver that controls a particular device.

MAN

(Metropolitan Area Network) A network that connects computers in a broad geographical area such as a city and its suburbs.

manual pages

The pages containing the complete documentation that is specific to every command, presented in simple ASCII text format.

MBR

(Master Boot Record) The first physical sector on a hard drive that contains the code that is used to load the operating system or boot loader into memory.

MD5

(Message-Digest algorithm 5) A command-line utility that generates and verifies message digests.

MDA

(Mail Delivery Agent) A program that delivers an incoming email message to the respective user's mailbox.

memory management layer

A layer in the kernel that manages the computer's memory.

memory usage

The sum of all the programs in the memory of an operating system.

menu.lst

A GRUB configuration file that lists all the kernels available in the system along with their partition numbers, boot information, and details of which kernel is booted.

message digest

A compact digital signature for an arbitrary stream of binary data.

minor number

A number stored as part of the structure of a device node. It identifies a particular device installed on the computer.

mkdir

A command that is used to create a directory.

mke2fs

A utility that is used to create ext2, ext3, and ext4 filesystems.

mkfs

A command that is used to build a Linux filesystem on a device, usually a hard disk partition.

mkinitrd

A command that is used to create the initial ramdisk image for preloading the kernel modules.

mknod

A command that allows you to create device files that are not present, using the major and minor node numbers of a device.

mkswap

A system administration command that is used to create swap space on a disk partition.

modinfo

A utility that displays information about a particular kernel module such as the file name of the module, license, description, author's name, module version number, dependent modules, and other parameters or attributes.

modprobe

A utility that is used to add or remove modules from a kernel.

modular kernel

A kernel in which only a minimal set of essential modules are built-in. It is also known as a micro kernel or a dynamic kernel.

monolithic kernel

A kernel in which all the required modules, such as device drivers or filesystems, are built-in.

motd

(Message of the day) The /etc/motd file that is displayed to all users after a successful login.

motions

Single-key shortcuts that are used to navigate through files in command mode.

mount point

An access point to information stored on a local or remote storage device.

MRA

(Mail Retrieval Agent) A retrieval agent that retrieves messages from the spool.

MTA

(Mail Transfer Agent) A mail transport program on the Internet that is used to send email messages through SMTP.

MUA

(Mail User Agent) A program that is used to read and compose email messages.

multitasking

The capability to perform more than one task at a time for each user.

mv

A command that is used to move or rename files and directories.

MySQL

An open source RDBMS used for managing data.

mysqld

(MySQL daemon) The MySQL server that manages the MySQL server service.

named.conf

A configuration file that is used to manage the BIND service.

nano

A small, user-friendly text editor that evolved from the Pico editor.

Nessus

An IDS that audits the security of remote hosts and services running on the network.

nested loops

Loops within other loops.

netcat

A utility for reading from and writing to network connections via the command-line using the TCP or UDP networking protocols.

netfilter

A framework that implements packet filtering in Linux for managing firewalls and securing data.

network

A group of computers connected together to communicate with each other and share resources.

network interface

A point of connection between two systems.

network protocol

A set of rules that enable communication and data transfer among network devices.

NIC

(Network Interface Card) A small circuit board device that is installed on a computer to enable a computer to connect to a network.

nice

A command that lets you run a command at a priority lower than the command's normal priority.

nice value

The priority of a process.

NTFS

(New Technology File System) A proprietary filesystem developed by Microsoft that is used in Windows environments.

NTP

(Network Time Protocol) A standard Internet protocol for synchronizing the internal system clock with a server or network clock.

ntp.conf

A file containing the configuration options for the NTP server.

online repositories

The repositories found on the Internet.

open source software

Software that allows users to access and modify its source code.

OpenSSH

A free version of the SSH protocol that ensures secure communication by encrypting data transmitted over the Internet.

package

A collection of classes, functions, or procedures that can be imported as a unit.

package integrity

The method of checking packages based on criteria, such as file structure, format, and checksum, along with verifying whether or not the packages are installed.

package manager

A tool that enables the installation, verification, upgrading, or removal of packages.

packet

A formatted unit of data being sent across a network.

packet filtering

A process of passing or blocking incoming and outgoing packets after inspecting each packet for user-defined content.

PAP

(Password Authentication Protocol) A security authentication protocol for logging in to a network.

parent directory

A directory that is one level above your current working directory.

parent process

A process that creates another process.

partition

A section of the hard disk that logically acts as a separate disk.

partition management

The process of creating, destroying, and manipulating partitions to optimize system performance.

partition utility

A program that is used to manage partitions on the hard disk.

partprobe

A program that is used to update the kernel with changes in the partition tables.

passive IDS

An IDS that detects potential security breaches, logs the activity, and alerts security personnel.

password policy

A set of guiding principles that help form effective passwords.

patch

A command that updates text files with changes according to instructions contained in a patch file.

path

An address that specifies a location in the filesystem.

PATH

See *PATH variable*.

PATH variable

A system variable that allows you to specify the directories that need to be searched while executing a command.

PDL

(Page Description Language) A computer language that is understood by a printer.

Perl

(Practical Extraction and Reporting Language) A programming language that is used to write scripts.

permissions

Security information that Linux uses to determine access to users to manipulate a particular file.

persistent configuration

A configuration where the kernel settings are permanent.

physical address

A globally unique hexadecimal number burned onto every NIC by the manufacturer.

physical bus

A set of USB cables that link the controller with the peripherals.

physical network interface

An interface that is implemented using a hardware device.

ping

A command that is used to test network connectivity.

pipe

An operator that combines commands.

PKC

(Public Key Cryptography) An encryption method that uses a public and private key pair.

POP3

(Post Office Protocol version 3) A protocol that is used to retrieve email messages over the TCP connection on port 110.

popd

A command that is used to remove entries from a stack of directories.

port

An access point to a logical connection. It serves as a point of exit and entry for data channels on a network.

portsentry

An IDS that is used to detect and respond to port attacks.

POST

(Power-On Self Test) A series of built-in diagnostics that are performed when the computer is started. Proprietary codes are generated to indicate test results.

PostScript

A Page Description Language (PDL) that tells a printer how to display text or graphics on a page.

PPID

(Parent Process ID) A number assigned to a process that spawns other processes.

primary partition

A partition in which the swap filesystem and the boot partition are normally created.

print queue

A temporary storage area that sorts incoming print jobs.

print server

A computer that provides users on a network access to a central printer.

printer software

The program that enables a printing device to print text or graphics on media.

private key

The key retained on the local system and not transmitted to the destination server.

PRNG

(Pseudo Random Number Generation) The algorithm that governs random number generation.

process

An instance of a running program that performs a data processing task.

process management layer

A layer in the kernel that handles different processes by allocating separate execution space on the processor and ensuring that the running of one process does not interfere with the other processes.

process monitoring

A mode of tracking the processes running on a system to determine the performance and reliability of the system.

process table

A record that summarizes the current running processes on a system.

profile file

A file that contains the commands to tailor a user's login session according to the requirements of the user.

program

A set of instructions describing how to carry out a task.

programming constructs

The parts of a program that define the order in which the instructions in a program are executed.

proxy server

A server that acts as an intermediary between a web browser and a web server, containing data to which the user requires access.

public key

The key transmitted to the destination server along with the request.

pushd

A command that is used to add a directory at the top of a stack of directories or rotate a stack of directories.

pwd

A command that is used to display the current working directory relative to the root directory.

quota report

A report created by the system to view the usage of disk space by each user.

RADIUS

(Remote Authentication Dial In User Service) A networking protocol that facilitates centralized authentication, authorization, and accounting over a network.

RAID

(Redundant Array of Independent Disks) A method that is used to store the same data in different locations on multiple hard disks of a server or a standalone disk storage system.

ramdisk

A portion of memory that is allocated and used as a partition.

random number generation

A method of encrypting data where the kernel is used to generate random numbers that are assigned to files before transfer.

RDBMS

(Relational Database Management System) A system that enables you to store and organize logically related data.

redirector

An operator that redefines the standard input or the standard output.

regular expressions

Strings of characters that denote a word, a set of words, or a sentence.

ReiserFS

A filesystem that can efficiently handle small file sizes (less than one kilobyte).

relational database

A database that stores logically related data consistently in the form of related tables.

relative path

The path relative to the current working directory.

remote X sessions

The sessions where a user on a remote workstation is able to view the X Window of the host.

renice

A command that enables you to alter the scheduling priority of a running process.

repomd

The XML metadata based on the rpm.

repository

The name of the database that holds the source code and compilations.

restore

A command that enables you to restore files or filesystems from backups made with the dump command.

reverse zone

A DNS zone that is used for mapping IP addresses to hostnames.

rm

A command that is used to delete files and directories.

rmdir

A command that is used to delete directories.

RMON

(Remote Monitoring) A network management protocol that allows network information to be gathered at a single workstation.

root disk

A disk that contains directories, which contain files required to run a Linux system.

root user

A user who has access rights to all files and resources on the system.

router

A networking device that connects multiple networks together.

routing

The process of selecting the best route for moving a packet from its source to destination on a network.

routing table

A table of network addresses that is used by routers to forward packets over networks.

RPM

(RPM Package Manager) A tool for maintaining packages.

rpm2cpio

A utility that enables you to make cpio backup of individual files from rpm packages.

rsyslog

A utility that tracks system logs. This is a higher-performance system for log processing as compared to the traditional syslogd utility.

RTC

(Real Time Clock or the hardware clock) The clock that keeps track of the time when the system is turned off and not when the system is on.

runlevel

A setting that specifies which group of processes runs on your system. Levels include single user, multiuser, reboot, and halt, among others.

Samba

A suite of network sharing tools that help in the sharing of files and printers on a heterogeneous network, which consists of computers running on different operating systems.

SASL

(Simple Authentication and Security Layer) A framework that provides authentication and data security to Internet protocols.

SCI layer

(System Call Interface) An abstraction layer that handles function calls sent from user applications to the kernel.

screen

This GNU command is a full-screen window manager that multiplexes a physical terminal between several processes, typically interactive shells.

SCSI

(Small Computer Systems Interface) A set of standards for connecting peripheral devices to a computer.

search path

A sequence of various directory paths that is used by the shell to locate files.

sector

The smallest unit of storage read or written onto a disk.

security context

The collection of all security settings pertaining to processes, files, and directories.

sed

A command line program that can modify text files according to command line parameters.

self-signed certificates

The certificates generated by a user and contains the public key of the user as the signature.

SELinux

(Security-Enhanced Linux) A security enhancement feature that implements various security policies on Linux operating systems.

seq

A command to print a sequence of numbers on the standard output.

service

A set of applications that perform tasks in the background.

service access controls

The daemons that are used to restrict access to certain important services.

sfdisk

A utility that is used to manipulate partitions.

SHA-2

(Secure Hash Algorithm version 2) A set of cryptographic hash functions designed by the U.S. National Security Agency (NSA) that includes the commonly used SHA-256 and SHA-512 hash functions.

shadow password file

A highly protected file that is used for storing each user's encrypted passwords.

shared library

A file containing routines that are used by applications.

shell

A component that interacts directly with users and functions as the command interpreter for the Linux system.

shell spawning

A process that allows a shell to create a clone of itself.

shutdown

The command that is used to shutdown or restart a system. This closes files and performs other tasks necessary to safely shutdown a system.

signals

The messages sent to a process to perform a certain action.

skel directory

A location where the default files and directories that need to be copied to the new user's home directory are stored.

slapd

(Stand-alone LDAP Daemon) An LDAP directory service which allows users to create and provide their own directory service that can be connected to the global LDAP directory service.

SMTP

(Simple Mail Transfer Protocol) A protocol that defines a set of rules to enable interaction between a program sending an email message to a server, and a program receiving an email message from a client.

SNMP

(Simple Network Management Protocol) A network management protocol that enables you to remotely monitor and configure network components such as bridges, routers, network cards, and switches.

snort

A network IDS that monitors the network traffic.

soft limit

The maximum amount of disk usage that a quota user has on a partition.

source

A bash shell scripting command that executes the content of the file pass as argument in the current shell.

spatial mode

The default mode of the Nautilus browser.

speed light

A status indicator light that indicates which speed the network adapter is operating at.

spool

A temporary storage space for files waiting to be printed.

spooler

A component that temporarily stores files waiting to be printed.

spooling

The procedure by which print jobs can be temporarily stored.

SSH

(Secure Shell) A network protocol that controls the secure flow of data among computers on a network.

ssh-agent

A program that holds private keys for public key authentication.

ssh-keygen

A command that generates, manages, and converts authentication keys.

SSL

(Secure Socket Layer) A protocol that ensures secure transactions between web servers and browsers.

standard error

A stream in Linux that is used as the destination for error messages.

standard input

The source for command input.

standard output

The destination for command output.

STDERR

The symbol that is used to refer to the standard error file.

STDIN

The symbol that is used to refer to the standard input file.

STDOUT

The symbol that is used to refer to the standard output file.

sticky bit

A permission bit that provides protection for files in a directory.

su

(substitute or switch user) A command that allows you to change the ownership of a login session without logging out.

subnet mask

A filter that tells the server whether an IP address is on the local network or a remote network.

subnets

A logical subsection of a large network.

sudo

(super user do) A command that allows users to run programs with the security privileges of the root user.

SUID script

A program that overrides normal permissions and runs with the permissions of the owner of the program.

superblock

A data structure that is stored on a disk and contains control information for a filesystem.

swap

A portion of the hard disk used when Linux runs out of physical memory and requires more.

swap space

A portion of the hard disk that is used in situations when Linux runs out of physical memory and needs more of it.

symbolic link

A reference to a file or directory that allows you to access mounted filesystems from a different directory.

symmetric encryption

An encryption type where only a single key is used to authorize information during encryption and decryption.

sysctl

A command that is used to view or set the kernel parameters at runtime.

syslog-ng

A utility that tracks system logs. This is a higher-performance system for log processing as compared to the traditional syslogd utility.

syslogd

A utility that tracks system logs.

system initialization

The first process that starts when the system is booted.

system load

A measurement of the amount of work done by a computer over a given period of time.

system logs

The records of system activities that are tracked by the syslogd utility.

system time

The time maintained by a computer's internal clock.

Systemd

An event-based init daemon that replaces Upstart in some Linux distributions.

tab completion

A feature that facilitates auto completion of commands and file names by pressing Tab.

tar

A command that creates archives of data.

TCP wrappers

The protection layers that define the host computers that are allowed to connect to some network services and those that are not.

TCP/IP

(Transmission Control Protocol/Internet Protocol) A protocol that is used to transfer packets of data from one system to another on a network.

Telnet

A terminal emulation protocol that enables a user on a site to simulate a session on a remote host.

terminal

A computer interface for text entry and display, where information is displayed as an array of preselected characters.

test constructs

The programming constructs that test for a condition and then act according to the result of the test.

text editor

An application that allows you to view, create, or modify the contents of text files.

text stream

A sequence of one or more lines of text that can be written to be read on a text-based display.

tmpwatch

A command that is used to delete files that have not been accessed for some time.

touch

A command that is used to modify an existing file to change the time of access or modification time of the file to the current time, or create an empty file.

tr

(translate) A command that is used to translate strings from the standard input to the standard output.

traceroute

A command that is used to print the route that packets take to reach their destination.

transactional configuration

A configuration where the kernel settings are updated for a required service.

Tripwire

An intrusion detection tool that compares the content of a file or a directory with a database that contains the locations of a file or directory and the dates modified.

Tripwire database

A database that contains baselines, which are snapshots of files and directories noted at a specific time.

troubleshooting

The recognition, diagnosis, and resolution of a problem.

troubleshooting model

A standardized approach to the troubleshooting process.

true time

The average time on a number of high accuracy clocks around the world.

tune2fs

A utility that helps tuning parameters associated with a Linux filesystem.

tunneling

A protocol in which one protocol is layered over the other, for a layered model.

TZ

The Time Zone environment variable set in /etc/profile for the system. Set with tzconfig or tzselect.

udev

A device manager that manages the automatic detection and configuration of hardware devices.

UDP

(User Datagram Protocol) A transport protocol that is part of the TCP/IP suite of protocols.

UID

(User ID) A unique ID number assigned to every user when an account is created.

umask

A command that automatically alters the default permissions on newly created files and directories.

uniq

A command that is used to display only unique lines from a sorted file.

UNIX

A well-known operating system originally developed by AT&T's Bell Labs in the 1970s.

unpack

The second stage in the Debian Archive Package Installation process. In this stage, the Debian archive package and its dependent packages are unpacked into the filesystem of the hard disk.

unzip

This GNU command lists, tests, and extracts compressed files in a ZIP archive.

UPG

(User Private Group) A unique group created by default whenever a new user account is created.

Upstart

An event-based init daemon that does not track runlevels.

USB

(Universal Serial Bus) A hardware interface standard designed to provide connections for numerous peripherals.

USB device

(Universal Serial Bus device) A peripheral device that can communicate with a host computer.

user account

A collection of information that defines a user on a system.

user profile

A description of the settings, preferences, bookmarks, stored messages, and other user items that characterize a user.

UTC

(Coordinated Universal Time) A time scale that forms the official measure of time in the world.

variable

A symbolic name that represents a value.

verbose

A mode that displays varying levels of status messages as the program is processed.

vfat

A 32-bit filesystem that supports long file names.

vi

The standard UNIX text editor.

Vim

A default Linux text editor especially meant for programming.

virtual network interface

An interface that is implemented through software support.

visual mode

A mode in Linux that allows users to highlight or select text for copying, deleting, and so on.

VNC

(Virtual Network Computing) A platform-independent system through which a user can control a remote system.

WAN

(Wide Area Network) A network that connects computers in a wide geographical area such as across the country or around the world.

wc

(word count) A command that is used to count the number of lines, words, and characters of text files.

whereis

A command that is used to locate the details associated with a command.

while

A loop that enables you to repeat a set of instructions for a fixed number of times while a specific condition is met.

wildcard

A special character that is used to substitute characters in a string.

window manager

See *display manager*.

X

See *X Windows*.

X client

An application that is written with the aid of the Xlib library, which gives programs access to any X server.

X forwarding

A mechanism by which programs are run in one machine and the X window output is displayed in another machine.

X protocol

The standard protocol used by clients and servers in the X Window system.

X server

A program that implements the GUI service provided by the X Window system.

X Windows

A client/server, multiuser system that resides on top of the operating system.

X-station

A terminal that is connected over a network and engineered to run the X Window system remotely.

X.Org

A free version of the X Window GUI system for Linux.

X11

See *X Windows*.

xargs

A command that allows you to construct and execute command lines.

xdm

(X Display Manager) A basic display manager that allows the user to only log in to the system.

Xfs

(X font server) A service that provides fonts to the X.Org server and the X client applications that connect to it.

XFS

This is a 64-bit, high-performance journaling filesystem that provides fast recovery and can handle large files efficiently. XFS is the default filesystem for CentOS Linux 7 installations.

xinetd

A daemon that controls system services on a network.

XTerm

A screen for typing system commands for the X Window system.

YUM

(Yellow dog Updater, Modified) A package manager similar to RPM.

zone

A point of delegation in a DNS tree structure.

Index

\$HISTFILESIZE [264](#)

A

Access Control List, *See* ACL

access controls

 service and application [419](#)

access control types [342](#)

accessibility

 keyboard options [523](#)

 options [523](#)

accessibility based themes [524](#)

ACL [171](#)

Address Resolution Protocol, *See* ARP

algorithms [426](#)

alien [211](#)

Anaconda installer [482](#)

archiving [135](#)

argument [12](#)

ARP [473](#)

assistive technologies [523](#)

authentication methods [425](#)

autoconf [213](#)

automatic rotation [331](#)

average time [314](#)

See also true time

B

background processes [287](#)

backup strategy guidelines [141](#)

Bash shell

 functions [250](#)

basic architecture of USB driver

 host computer [451](#)

 host controller hardware layer [451](#)

 physical bus [451](#)

 upper software layer [451](#)

 USB devices [451](#)

basic filesystem commands [87](#)

Basic Input/Output System, *See* BIOS

binaries [92](#)

BIND attack [474](#)

biometric authentication [425](#)

BIOS [484](#)

block special files [78](#)

boot disks [465](#)

booting devices [485](#)

boot loaders

 components [488](#)

 overview [38](#)

 types [489](#)

boot managers [488](#)

See also boot loaders

boot process [484](#), [492](#)

Bourne-Again SHell, *See* Bash shell

definition

broadcast addresses [357](#)

browser mode [81](#)

built-in help options [30](#)

bzip2 file archiving utilities [140](#)

C

cat command options [24](#)

CDPATH [262](#)

cells [365](#)

central network log server [330](#)

character special files [78](#)

check for dependency [209](#)

child process [265](#)

chmod command

- modes [159](#)
- options [159](#)
- chroot mode [462](#)
- CIDR [355](#)
- class-based networks [355](#)
- classless addressing [355](#)
- CLI [9](#)
- clock drift [317](#)
- CMOS [485](#)
- Command Line Interface, *See* CLI
- command line interpreter [9](#)
- command prompt [9](#)
- commands
 - `:.help` [110](#)
 - `:q` [110](#)
 - `$SHELL` [10](#)
 - `#!/bin/bash` [256](#)
 - alias [263](#)
 - apropos [29](#)
 - apt-get [210](#)
 - aptitude [211](#)
 - aspell [114](#)
 - at [310](#)
 - awk [338](#)
 - bash [10](#)
 - cal [13](#)
 - cat [23](#)
 - chattr [168](#)
 - chkconfig [325](#)
 - chmod [158](#), [265](#)
 - chown [57](#), [165](#)
 - cpio [135](#)
 - csh [10](#)
 - date [12](#)
 - dd [136](#), [466](#)
 - debugfs [100](#)
 - df [447](#)
 - diff [112](#)
 - dig [383](#)
 - du [447](#)
 - dump [136](#)
 - dumpe2fs [99](#)
 - e2fsck [98](#)
 - e2label [65](#)
 - echo [10](#), [21](#), [261](#)
 - exec [257](#)
 - exit [10](#)
 - export [260](#)
 - fdisk [69](#), [482](#)
 - file [78](#)
 - find [121](#)

- finger [47](#)
- for [280](#)
- free [242](#)
- fsck [97](#)
- gdisk [75](#)
- gedit {file name} [53](#)
- getent [385](#)
- GNU parted [74](#)
- gpg [428](#)
- grep [120](#), [336](#)
- groupadd [48](#)
- groupdel [58](#)
- groupmod [58](#)
- grub2-mkconfig [500](#)
- grub-mkconfig [500](#)
- gzip [138](#)
- head [11](#)
- history [252](#)
- host [384](#)
- hostname [17](#)
- id [47](#), [438](#)
- if [279](#)
- if...else [279](#)
- ifconfig [362](#)
- ifdown [363](#)
- ifup [363](#)
- inetd [327](#)
- ip [363](#), [370](#)
- iwconfig [365](#)
- jobs [288](#)
- journalctl [335](#)
- last [19](#)
- lastlog [336](#)
- less [11](#)
- less /etc/passwd [11](#)
- ln [132](#)
- locate [119](#)
- logrotate [308](#), [331](#)
- logwatch [309](#)
- lpc [183](#)
- lpd [35](#)
- lpr [182](#)
- ls [51](#)
- ls -a [51](#)
- lsattr [169](#)
- m4 [409](#)
- man [28](#)
- md5sum [428](#)
- mkdir [57](#)
- mke2fs [72](#)
- mkfs [71](#)

mkinitrd [232](#)
mknod [237](#)
mkswap [93](#)
more [11](#)
more /etc/passwd [11](#)
mount [91](#)
netcat [374](#)
netstat [373](#)
nice [299](#)
nohup [302](#)
nslookup [385](#)
ntpdate [318](#)
openssl [428](#)
partprobe [76](#)
passwd [45](#)
passwd -l [57](#)
patch [113](#)
ping [351](#)
pmap [242](#)
pr [183](#)
ps [292](#)
pstree [294](#)
pwd [84](#)
quotacheck [454](#)
rdesktop [398](#)
rdev [465](#)
read [270](#)
renice [300](#)
restore [142](#)
rm [437](#)
route [371](#)
rpm [431](#)
rpm2cpio [199](#)
rpm -Fvh *.rpm [204](#)
rslog [333](#)
sar [243](#)
scp [393](#)
screen [303](#)
sfdisk [73](#)
sftp [393](#)
sha1sum [428](#)
sha256sum [428](#)
sha512sum [428](#)
shutdown [38](#)
sleep [22](#)
ssh-keygen [390](#)
startx [512](#)
su [437](#)
sudo [438](#)
swapoff [93](#)
swapon [93](#)

sysctl [228](#)
syslogd [331](#)
syslog-ng [334](#)
system-config-date
[317](#)
tail [11](#), [338](#)
tar [137](#)
tee [273](#)
test [256](#)
tmpwatch [307](#)
top [298](#)
tr [114](#)
traceroute [373](#)
tune2fs [98](#)
umask [162](#)
umount [91](#)
uniq [114](#)
unset [53](#)
unzip [140](#)
uptime [15](#)
useradd [44](#)
usermod [56](#)
vim [108](#)
vimdiff [113](#)
vimtutor [110](#)
vmstat [242](#)
vncserver [396](#)
vncviewer [397](#)
w [18](#)
wall [20](#)
wc [113](#)
whatis [28](#)
whereis [120](#)
which [24](#), [53](#)
while [281](#)
who [16](#)
whoami [17](#)
xargs [273](#)
xauth [513](#)
xfs_admin [99](#)
xfs_repair [98](#)
xhost [513](#)
xinetd [327](#)
xinit [512](#)
xvidtune [509](#)
xz [139](#)
yum [431](#)
zip [140](#)

command substitution [274](#)

Common UNIX Printing System, *See* CUPS

Complementary Metal Oxide Semiconductor,
See CMOS
configure [209](#)
console [14](#)
 See also terminal
control statements [276](#)
copyleft [3](#)
core system variables [464](#)
counts [111](#)
cron jobs [306](#)
crontab [306](#)
cryptographic hashes [427](#)
CUPS [177](#)
cylinder [73](#)

D

daemons
 anacron [311](#)
 cron [306](#)
 httpd [419](#)
 klogd [332](#)
 logging service [329](#)
 ntpd [314](#)
 squid [419](#)
 xinetd [416](#)
databases [143](#)
Data Encryption Standard, *See* DES
datagrams [370](#)
date/time format [318](#)
date Command characters [13](#)
dbus [234](#)
D-Bus [238](#)
Debian installation process [209](#)
Debian package management commands [210](#)
DEB tools [210](#)
default environment variables [262](#)
default gateway
 default gateway address [373](#)
default permissions [162](#)
default user accounts [45](#)
DES [426](#)
device drivers [234](#)
device nodes [235](#)
device tree [235](#)
DHCP
 overview [378](#)
 process [379](#)
digital certificate types
 Certificate Authority [430](#)
 Self-signed [430](#)
directories

 /boot/grub [494](#)
 /boot/grub2 [497](#)
 /etc/cups [188](#)
 /etc/init.d [325](#)
 /etc/localtime [318](#)
 /etc/ntp [314](#)
 /etc/sysconfig [327](#)
 /etc/xinetd.d [418](#)
 /lib/modules [223](#)
 /proc [227](#)
 /sys [228](#), [234](#)
 /usr/lib/rpm/* [197](#)
 /usr/share/doc [30](#)
 /usr/share/zoneinfo/ [318](#)
 /usr subdirectories [79](#)
 /var/lib/dpkg/* [211](#)
 RPMS [197](#)
 skel [52](#)
directory
 current working [83](#)
 home [82](#)
 parent [84](#)
disk image [90](#)
 See also ISO image
disk quotas [453](#)
display managers
 GDM [517](#)
 KDM [517](#)
 xdm [517](#)
distributions [6](#)
distros [6](#)
DNS [380](#)
DNS resource records [387](#)
domain name resolution process [383](#)
domain names [381](#)
Domain Name System, *See* DNS
Dynamic Host Control Protocol, *See* DHCP

E

editing operators [111](#)
ELILO [489](#)
Emacs [107](#)
email process [410](#)
encryption
 asymmetric [429](#)
 overview of [426](#)
 symmetric [428](#)
encryption solutions
 3DES [426](#)
 Blowfish [426](#)
 MD5 [426](#)

- SHA-2 [426](#)
- entropy [237](#)
- environment configuration problems [463](#)
- environment files [436](#)
- environment variables
 - overview of [262](#)
- PATH [266](#)
- execute mode commands [109](#)
- expressions [276](#)

F

- fdisk utility options [70](#)
- FHS [78](#)
- fields [143](#)
- file browsers [80](#)
- file compression utilities [139](#)
- file naming conventions [80](#)
- file owner [158](#)
- files
 - /etc/crontab [310](#)
 - /etc/default/grub [498](#)
 - /etc/group [49](#)
 - /etc/grub.d/ [499](#)
 - /etc/hosts.equiv [187](#)
 - /etc/hosts.lpd [187](#)
 - /etc/issue [142](#)
 - /etc/issue.net [142](#)
 - /etc/login.defs [436](#)
 - /etc/passwd [46](#)
 - /etc/shadow [47](#)
 - /etc/skel [52](#)
 - /etc/ssh/ssh_config [391](#)
 - /etc/syslog.conf [333](#)
 - /etc/timezone [318](#)
 - /etc/xinetd.conf [417](#)
 - /proc/modules [223](#)
 - /proc/version [227](#)
 - /var/log/lastlog [336](#)
 - /var/log/messages [333](#)
 - .rhosts [393](#)
 - .shosts [393](#)
 - ~/.bash_profile [52](#)
 - apt.conf [211](#)
 - boot.iso [465](#)
 - diskboot.img [465](#)
 - fstab [70](#)
 - grub.cfg [497](#)
 - grub.conf [494](#)
 - inittab [324](#)
 - known_hosts [392](#), [425](#)
 - libwrap.so [418](#)

- log [329](#)
- makefile [212](#)
- menu.lst [495](#)
- modprobe.conf [225](#)
- named.conf [386](#)
- ntp.conf [315](#)
- NTP drift [314](#)
- printers.conf [188](#)
- sysctl.conf [228](#)
- xorg.conf [508](#)

Filesystem Hierarchy Standard, *See* FHS

filesystem integrity [97](#)

filesystems

- definition [64](#)
- labels [65](#)
- types [66](#)

File Transfer Protocol, *See* FTP

filters [128](#)

find command conditions [123](#)

FIPS [484](#)

firewalls

- hardware [569](#)

FireWire [451](#)

First nondestructive Interactive Partition Splitting program, *See* FIPS

font path [510](#)

foreground processes [287](#)

FQDN [382](#)

Free Software Foundation, *See* FSF

FSF [3](#)

FTP [351](#)

Fully Qualified Domain Name, *See* FQDN

functions [277](#)

G

gateways [373](#)

GCC [227](#)

GDE [518](#)

gedit [107](#)

General Public License, *See* GPL

global user profiles [52](#)

globbing [252](#)

GMT [315](#)

GNOME desktop environment, *See* GDE

GNOME On-Screen Keyboard, *See* GOK

GNOME system monitor [243](#)

GNU Compiler Collection, *See* GCC

GNU project [3](#)

GOK [523](#)

GPL [4](#)

GRand Unified Bootloader, *See* GRUB

Graphical User Interface, *See* GUI

Greenwich Mean Time, *See* GMT

group database [49](#)

group management [58](#)

groups [48](#)

GRUB

commands [496](#)

configuration [494](#)

menu-specific commands [497](#)

overview of [489](#)

GRUB 2

configuration [497](#)

overview of [489](#), [493](#)

GUI [8](#)

Gvim [107](#)

H

HAL

utilities [238](#)

hardware

components [446](#)

device types [236](#)

problems [469](#)

removable [450](#)

resources [446](#)

Hardware Abstraction Layer, *See* HAL

hardware communication channels

Direct Memory Address (DMA) [237](#)

Input/Output (I/O) Addresses [237](#)

Interrupt ReQuests (IRQ) [237](#)

HCI [451](#)

honeypot [474](#)

Host Controller Interface, *See* HCI

HTTP [351](#)

HyperText Transfer Protocol, *See* HTTP

I

IANA [360](#)

ICMP [351](#)

IDS

passive vs. active [575](#)

ifconfig command options [363](#)

IMAP [407](#)

immutable flag [170](#)

index node [131](#)

See also inode

index node table [131](#)

init [35](#)

init process [287](#)

initrd [231](#)

initrd image [232](#)

inodes [131](#)

input and output redirection [115](#)

installation

Anaconda installer [482](#)

documentation [502](#)

hardware compatibility [480](#)

methods [502](#)

partitioning [482](#)

Internet Assigned Numbers Authority, *See*

IANA

Internet Control Message Protocol, *See* ICMP

Internet Message Access Protocol, *See* IMAP

IP address classes

Class A [354](#)

Class B [354](#)

Class C [354](#)

IP addresses

allocation [379](#)

overview [352](#)

IP classes [354](#)

ip command options [364](#)

IP Next Generation, *See* IPng

IPng [352](#)

IP spoofing [474](#)

IPv4 [352](#)

IPv6 [352](#)

IP Version 4, *See* IPv4

IP Version 6, *See* IPv6

ISO image [90](#)

iwconfig command options [365](#)

J

job control tools [290](#)

jobs

delayed [302](#)

detached [302](#)

table [288](#)

joins

inner [150](#)

outer [150](#)

journaling filesystems [97](#)

K

KDE desktop environment [518](#)

Kerberos [425](#)

kernel

definition [220](#)

layers [221](#)

modular [222](#)

- monolithic [222](#)
- versions and modules [221](#)
- kernel configuration
 - persistent [226](#)
 - transactional [227](#)
- kernel module [223](#)
- kernel module utilities [224](#)
- kernel options [226](#)
- kernel state monitoring utilities [240](#)
- key files [391](#)
- keys
 - private [424](#)
 - public [424](#)
- kill commands [296](#)
- KWrite [107](#)

L

- LAN [350](#)
- leap seconds [315](#)
- LED [362](#)
- libwrap.so.0 [432](#)
- LightDM desktop manager [520](#)
- links
 - hard [133](#)
 - symbolic [133](#)
- Linux
 - distributions [6](#)
 - documentation [27](#)
 - uses [5](#)
- Linux kernel [220](#)
- Linux rescue environment [461](#)
- Linux User Groups, *See* LUGs
- LMTP [408](#)
- load average [240](#)
- Local Area Network, *See* LAN
- locale settings [315](#)
- Local Mail Transport Protocol, *See* LMTP
- local storage devices [96](#)
- log file analysis [335](#)
- logger [332](#)
- Logical Volume Manager, *See* LVM
- login levels [436](#)
- loopback device [452](#)
- loops [279](#)
- LUGs [31](#)
- LVM [482](#)

M

- MAC address [361](#)
- Mail Delivery Agents, *See* MDAs

- mail forwarding [410](#)
- mail protocols [406](#)
- mail queues [408](#)
- Mail Retrieval Agents, *See* MRAs
- Mail Transfer Agents, *See* MTAs
- Mail User Agents, *See* MUAs
- major number [235](#)
- makefile commands [212](#)
- MAN [350](#)
- man command options [28](#)
- manual pages [28](#)
- MariaDB
 - commands [147](#)
 - configuration file [145](#)
 - overview of [144](#)
 - service [146](#)
- Master Boot Record, *See* MBR
- MBR [491](#)
- MDAs [410](#)
- Media Access Control (MAC) address [361](#)
 - See also* physical address
- memory monitoring utilities [241](#)
- memory usage [439](#)
- message digests [426](#)
- message of the day, *See* motd
- Metropolitan Area Network, *See* MAN
- minor number [235](#)
- modprobe [225](#)
- motd [141](#)
- motions [110](#)
- mount command options [91](#)
- mount points [90](#)
- MRAs [410](#)
- MTAs
 - overview of [408](#)
 - types [408](#)
- MUAs [410](#)
- multiple SSH connections [393](#)
- multitasking [288](#)

N

- nano [107](#)
- Network Interface Card, *See* NIC
- network interfaces
 - physical [360](#)
 - virtual [360](#)
- network issues [472](#)
- network protocols
 - types [351](#)
- networks
 - definition [350](#)

- security problems [473](#)
- security vulnerabilities [474](#)
- types [350](#)

Network Time Protocol, *See* NTP

NIC

- characteristics [361](#)
- overview [361](#)

nice value [299](#)

NTP [314](#)

O

online help [31](#)

open source software [2](#)

OpenSSH [390](#)

Orca [524](#)

OSI model

- and Linux+ troubleshooting [461](#)

P

package

- definition [194](#)
- dependencies [195](#)
- managers [194](#)
- verification error codes [202](#)

package integrity [431](#)

packets [370](#)

packet-switching technology [370](#)

Page Description Language, *See* PDL

Parent Process ID, *See* PPID

partitionless installation [484](#)

partition management [73](#)

partitions

- overview of [67](#)
- types of [68](#)
- utilities [482](#)

partprobe program [76](#)

password policies [439](#)

passwords [45](#)

paths

- absolute [86](#)
- relative [86](#)

PATH variable [24](#)

PDL [176](#)

Perl [330](#)

permission levels [158](#)

permissions [156](#)

physical address [361](#)

PID [286](#)

ping command options [366](#)

pipe [272](#)

PKC [389](#)

POP3 [407](#)

port forwarding [395](#)

port ranges [360](#)

ports [358](#)

POST [484](#)

Post Office Protocol version 3, *See* POP3

PostScript [176](#)

Power-On Self Test, *See* POST

PPID [293](#)

Practical Extraction and Reporting Language,
See Perl

pr command options [184](#)

printer commands [182](#)

printer software [176](#)

print process [178](#)

print queues [179](#)

print servers [186](#)

private networks [353](#)

PRNG [427](#)

processes

- child [293](#)
- parent [293](#)

Process ID, *See* PID

process identification commands [295](#)

process monitoring [243](#)

process states [297](#)

process table [291](#)

profile file [52](#)

program [288](#)

programming constructs [277](#)

proxy servers [393](#)

ps command options [292](#)

Pseudo Random Number Generation, *See*
PRNG

Public Key Cryptography, *See* PKC

Q

quota management commands [454](#)

quota reports

- generation commands [454](#)
- grace period [454](#)
- hard limit [454](#)
- soft limit [454](#)

R

RADIUS [431](#)

RAID [482](#)

ramdisks [465](#)

ramdisk word [465](#)

- random number generation [427](#)
- RDBMS [144](#)
- Real Time Clock, *See* RTC
- redirectors [271](#)
- Redundant Array of Independent Disks, *See* RAID
- regular expressions [126](#)
- Relational Database Management System, *See* RDBMS
- relational databases [143](#)
- Remote Authentication Dial In User Service, *See* RADIUS
- Remote Monitoring, *See* RMON
- remote printer permissions [187](#)
- remote printing [187](#)
- remote X sessions
 - commands [514](#)
 - overview of [513](#)
- removable hardware [450](#)
- repair filesystems [98](#)
- repartitioning
 - destructive [483](#)
 - nondestructive [483](#)
- repositories
 - local or private [205](#)
 - online [205](#)
- Request To Send, *See* RTS
- rescue environment utilities [462](#)
- resolver files [385](#)
- RMON [400](#)
- rogue public keys [430](#)
- root disks [465](#)
- root user [82](#)
- routers [369](#)
- routing [369](#)
- routing tables [371](#)
- RPM
 - commands [198](#)
 - components [198](#)
 - overview [196](#)
 - queries [199](#)
 - verification [201](#)
- RPM Package Manager, *See* RPM
- RTC [317](#)
- RTS [365](#)
- runlevel [35](#)

S

- Samba [188](#)
- SCSI
 - IDs [447](#)
 - overview of [446](#)
 - search paths [266](#)
 - sectors [490](#)
 - Secure Shell, *See* SSH
 - security context [343](#)
 - Security-Enhanced Linux, *See* SELinux
 - security policies [344](#)
 - SELinux [343](#)
 - server keys [392](#)
 - services [34](#)
 - Set Group ID, *See* SGID
 - Set User ID, *See* SUID
 - SGID [168](#)
 - shadow password file [439](#)
 - shadow passwords [47](#)
 - shared libraries [214](#)
 - shell commands [11](#)
 - shells [9](#)
 - shell scripts [255](#)
 - shell spawning [265](#)
 - signals [296](#)
 - Simple Mail Transfer Protocol, *See* SMTP
 - Simple Network Management Protocol, *See* SNMP
 - single-user mode [464](#)
 - Small Computer Systems Interface, *See* SCSI
 - SMTP [406](#)
 - SNMP [399](#)
 - software acquisition [7](#)
 - spatial mode [81](#)
 - special devices [236](#)
 - special permission commands [171](#)
 - special permissions [168](#)
 - spool [179](#)
 - spooling [179](#)
 - SSH [389](#)
 - ssh-agent [392](#)
 - SSH protocol versions [392](#)
 - standard directories [79](#)
 - standard error [271](#)
 - See also* STDERR
 - standard groups [48](#)
 - standard input [270](#)
 - See also* STDIN
 - standard output [270](#)
 - See also* STDOUT
 - status indicator lights
 - activity light [362](#)
 - link light [362](#)
 - speed light [362](#)
 - STDERR [271](#)

- STDIN [270](#)
- STDOUT [270](#)
- sticky bits [170](#)
- subdomains [382](#)
- subnet masks [353](#)
- subnets [365](#)
- SUID [168](#)
- SUID scripts [265](#)
- superblocks [490](#)
- swap space [92](#)
- switch modes [109](#)
- systemctl [37](#)
- system initialization [324](#)
- system load [241](#)
- system logs [329](#)
- system security monitoring tools [473](#)
- system time [317](#)

T

- tab completion [252](#)
- tarballs [213](#)
- TCP [351](#)
- TCP wrappers [432](#)
- Telnet [416](#)
- terminal [14](#)
- test constructs [277](#)
- text editor [106](#)
- text editors list [107](#)
- text streams [128](#)
- textutil commands [129](#)
- Transmission Control Protocol, *See* TCP
- troubleshooting
 - boot process [462](#)
 - hardware problems [469](#)
 - models [460](#)
 - network issues [472](#)
 - network security [473](#)
 - strategies [460](#)
 - tools [468](#)
- true time [314](#)
- tunneling [393](#)

U

- udev [234](#)
- UDP [351](#)
- UID [44](#)
- Universal Serial Bus, *See* USB
- Universal Time Coordinated, *See* UTC
- UNIX [3](#)
- unpack [209](#)

- UPG [49](#)
- Upgrade/Freshen Packages [204](#)
- USB
 - devices [450](#)
 - drivers [451](#)
- user accounts [44](#)
- User Datagram Protocol, *See* UDP
- userdel [55](#)
- User ID, *See* UID
- user-level security
 - ways to improve [439](#)
- User Private Group, *See* UPG
- user profiles [51](#)
- UTC [315](#)

V

- variables [260](#), [278](#)
- verbose [71](#)
- Vi [107](#)
- Vim [107](#)
- Vim help options [110](#)
- Vim modes [108](#)
- virtual desktops [511](#)
- Virtual Network Computing, *See* VNC
- VNC [395](#)

W

- WAN [350](#)
- Wide Area Network, *See* WAN
- wildcards [251](#)
- window managers [517](#)
 - See also* display managers

X

- X [508](#)
 - See also* X Windows
- X.Org [508](#)
- X11 [508](#)
 - See also* X Windows
- X clients [510](#)
- X Display Manager, *See* xdm
- xdm [517](#)
- X font servers, *See* Xfs
- X forwarding [394](#)
- Xfs [510](#)
- xfs tools [100](#)
- xinetd access controls [419](#)
- XOrg runlevels [512](#)
- X protocol [509](#)

X servers [508](#)
X-stations [514](#)
X Windows [508](#)

Y

Yellow dog Updater, Modified, *See* YUM

YUM

commands [207](#)

Z

zero-filled files [466](#)

zones

forward [382](#)

reverse [383](#)

