CompTIA.

The Official CompTIA

Linux+ Study Guide

Exam XK0-004

Official CompTIA Content Series for CompTIA Performance Certifications

The Official CompTIA® Linux+® Study Guide (Exam XK0-004)

COURSE EDITION: 1.0

Acknowledgements

Jason Nufryk, Author
Damon Garn, Author
Brian Sullivan, Media Designer
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

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+®, and the CompTIA logo are registered trademarks of CompTIA, Inc., in the U.S. and other countries.

CentOS® 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.

Ubuntu® is a registered trademark of Canonical Ltd., in the U.S. and other countries; the Canonical products and services discussed or described may be trademarks of Canonical Ltd.

All other product and service names used may be common law or registered trademarks of their respective proprietors.

COPYRIGHT NOTICE

Table of Contents

About This Guide

For many years, Linux has dominated the server install base in the business world—and it will continue to do so in the foreseeable future. Linux's popularity has led to a greater need for information technology (IT) professionals who can manage servers that run some form of the Linux kernel. *The Official CompTIA® Linux+® Study Guide* builds on your existing experience with systems operations and administration to provide you with the knowledge and skills required to configure, manage, operate, and troubleshoot a Linux environment by using security best practices, scripting, and automation.

Guide Description

Target Student

This guide is designed for IT professionals whose primary job responsibility is the management of servers and other devices running the Linux operating system. A typical student should have at least nine months of hands-on Linux experience and at least one and a half years of IT experience in other computing environments. The target student should wish to expand their skillset to support their career in Linux system administration and operation.

This guide is also designed for students who are seeking the CompTIA Linux+ certification and who want to prepare for Exam XK0-004. The Linux+ certification can validate the student's understanding and skill in configuring, monitoring, and supporting Linux systems.

Guide Prerequisites

To ensure your success, you should have at least foundational experience with general systems administration procedures, some hands-on exposure to one or more Linux distributions, as well as knowledge of computing hardware and basic networking and cybersecurity concepts.

You can obtain this level of skills and knowledge by taking the following official CompTIA courses:

- *The Official CompTIA® A+® Core 1 and 2 Study Guide (Exams 220-1001 and 220-1002)*
- *The Official CompTIA® Network+® Study Guide (Exam N10-007)*
- *The Official CompTIA® Security+® Study Guide (Exam SY0-501)*

Note: These prerequisites might differ significantly from the prerequisites for the CompTIA certification exams. For the most up-to-date information about the exam prerequisites, complete the form on this page: **https://certification.comptia.org/training/exam-objectives**

Guide Objectives

In this guide, you will configure, operate, and troubleshoot Linux systems.

You will:

- Perform basic Linux tasks.
- Manage users and groups.
- Manage permissions and ownership.
- Manage storage.
- Manage files and directories.
- Manage kernel modules.
- Manage the Linux boot process.
- Manage system components.
- Manage devices.
- Manage networking.
- Manage packages and software.

- Secure Linux systems.
- Write and execute Bash shell scripts.
- Automate tasks.
- Plan and perform a Linux installation.

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 activities designed to enable you to solidify your understanding of the informational material presented in the guide. Information is 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. In many electronic versions of the book, you can click links on key words in the content to move to the associated glossary definition, and on page references in the index to move to that term in the content. To return to the previous location in the document after clicking a link, use the appropriate functionality in your PDF viewing software.

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.

Student Icon	Student Icon Descriptive Text
	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.

Lesson 1

Performing Basic Linux Tasks

LESSON TIME: 1 HOUR, 30 MINUTES

LESSON INTRODUCTION

There is a great amount of depth to the Linux® operating system. Rather than dive right into the specifics, you'll do well to get a high-level overview of what it is you're about to work with. Also, by operating Linux in some fundamental ways, you'll be better prepared for the journey ahead.

LESSON OBJECTIVES

In this lesson, you will:

- Identify the high-level design concepts that make up the Linux operating system.
- Use fundamental Linux shell commands to get started with the command-line interface (CLI).
- Use various resources to find help on the Linux operating system.

Topic A

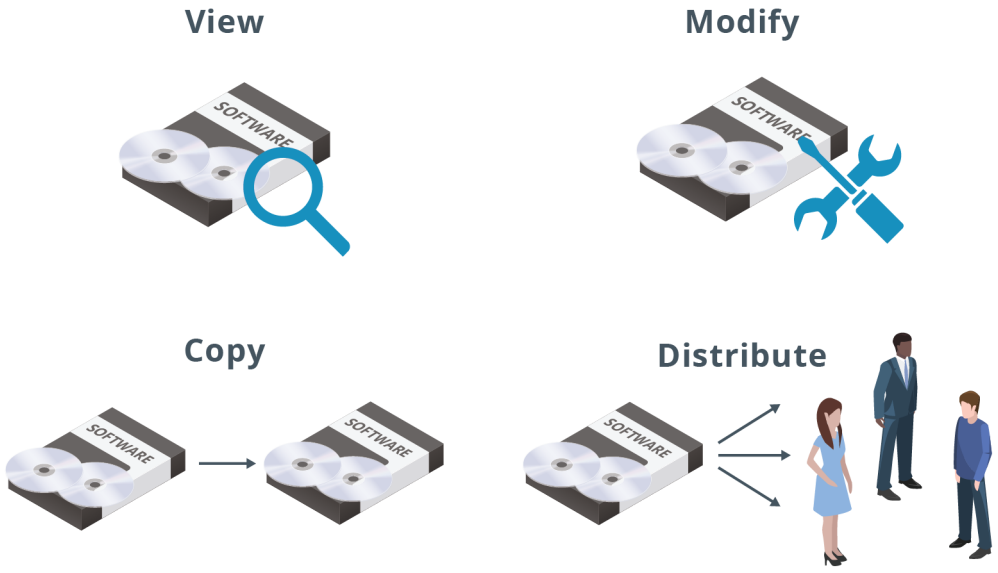
Identify the Linux Design Philosophy

Unlike other operating systems you may have worked with, the history of Linux is driven by a philosophy of openness and transparency. This philosophy has informed the design of Linux as an operating system, as well as its application in the business world. It's important for you to understand this philosophy and how it will ultimately have an impact on your day-to-day operations.

OPEN SOURCE SOFTWARE

Open source software (OSS) refers to computer code that any user is permitted to view, copy, and modify for any reason, as well as distribute to anyone. Users are granted these rights when the author of the software releases the source code under one of several open source licenses. The opposite of OSS is proprietary software—software that is released under a license that imposes restrictions on one or more of the rights just mentioned (view, copy, modify, distribute).

OSS provides several advantages to users, administrators, and programmers alike. Perhaps the most important element of OSS is that it encourages the ongoing improvement of software in a collaborative, community-driven environment. Individuals or groups of developers may build upon another developer's work to create enhanced or customized software, all while avoiding legal issues. There are many examples of OSS, one of which is the Linux kernel.

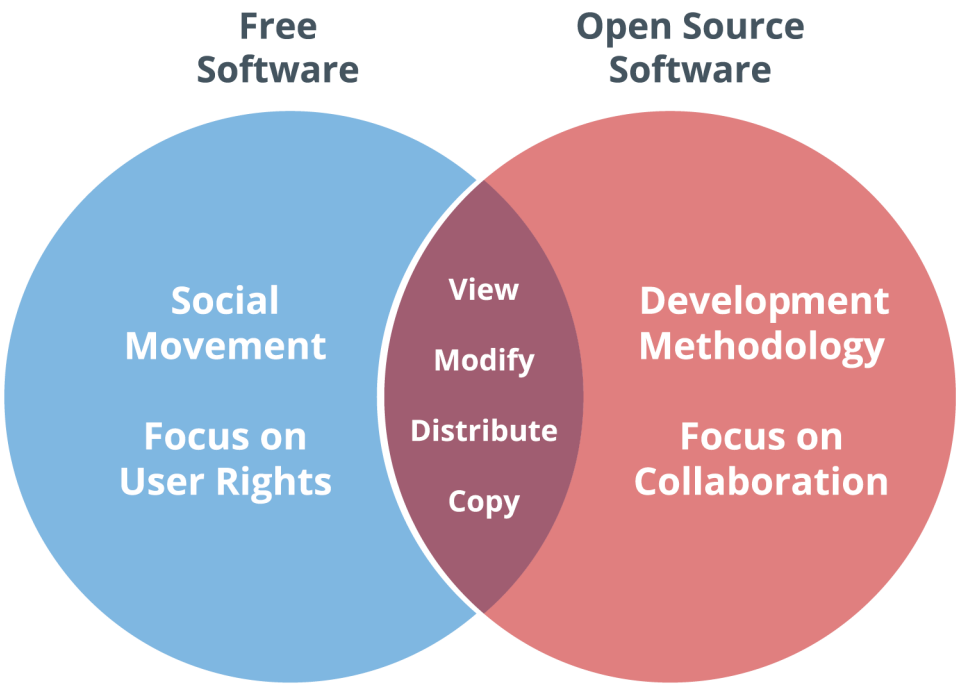


The fundamental permissions granted by open source software.

FREE SOFTWARE VS. OPEN SOURCE SOFTWARE

The term **free software** is often used interchangeably with OSS, and both share a great deal of overlap. However, some draw a distinction between the two. Richard Stallman, the founder of the free software movement, argues that the terms share different values—in his words, "Open source is a development methodology; free software is a social movement." In other words, OSS is most concerned with fostering open collaboration, whereas free software is most concerned with upholding users' rights. Note that "free" is being used in the sense of freedom, not in the sense of having no monetary cost.

In an attempt to resolve this terminology dispute, some developers have adopted the term **free and open source software (FOSS)** to describe any project that embodies the values of both movements.



A Venn diagram of free software and open source software.

FREE AND OPEN SOURCE LICENSES

There are several FOSS licenses that a developer can release their software under. These licenses may differ when it comes to additional restrictions or compatibility with other licenses. The following are some examples of FOSS licenses:

- Apache License
- Berkeley Software Distribution (BSD) license family
- Creative Commons Zero (CC0)
- Eclipse Public License (EPL)
- GNU General Public License (GPL)
- Massachusetts Institute of Technology (MIT) License
- Mozilla Public License (MPL)

THE GNU PROJECT

The **GNU Project** is a free software project led by Richard Stallman. Announced in 1983, it was the first project of its kind, and effectively launched the free software movement. The GNU Project's goal is to create an operating system that is composed

of entirely free software. By the early 1990s, the GNU Project had not completed a full OS, but had created several programs. Stallman wrote much of the GNU software himself, including the GNU C Compiler (GCC) and the Emacs text editor. Later, several programmers worked together to develop more utilities that are compatible with GNU software.

Stallman chose the recursive acronym "GNU's Not Unix" to show that GNU software was similar in design to the proprietary Unix operating system, but did not actually contain any Unix code.

Note: *The "G" in GNU is included in the pronunciation of the term "guh-NOO."*

FREE SOFTWARE FOUNDATION AND THE GNU GPL

Shortly after kicking off the GNU Project, Stallman founded the nonprofit **Free Software Foundation (FSF)** to promote the philosophy behind free software. Stallman and the FSF created the **GNU General Public License (GPL)** to support the GNU Project. One of the requirements of the GPL is that any derivative of a GPL work must also be distributed under that same license—a concept known as **copyleft**.

THE UNIX PHILOSOPHY

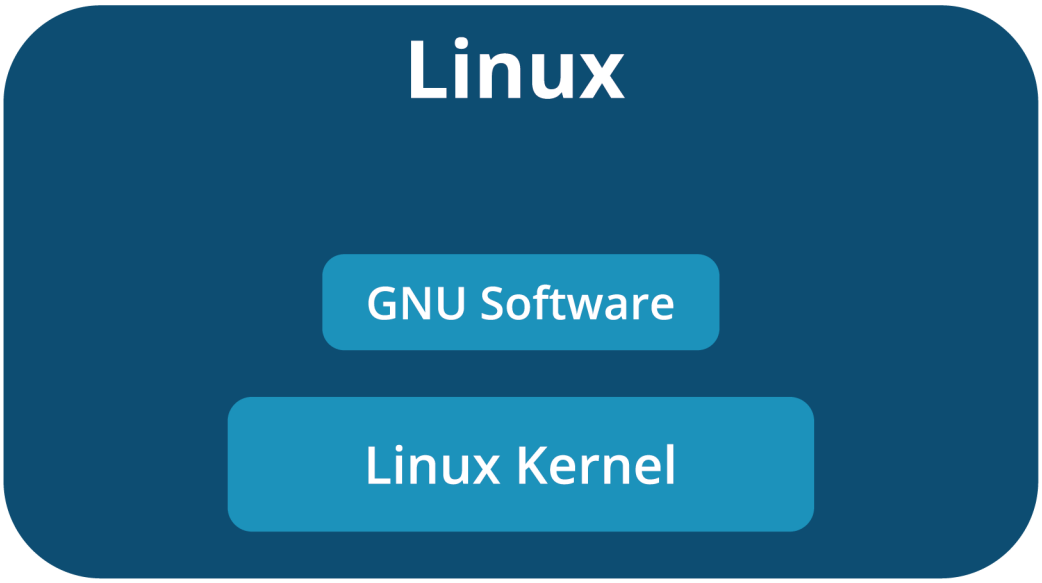
Because GNU software is based on the design of Unix®, it tends to conform to the Unix philosophy. The **Unix philosophy** is a set of best practices and approaches to software development that emphasize simplicity and modularity. This philosophy was created by the lead developers of the Unix operating system and has been summarized in many ways. Some of the key high-level points include:

- **Do one thing and do it well**. Rather than writing a monolithic software tool that accomplishes a variety of disparate tasks, write individual tools that fulfill a specific function.
- **Worse is better**. Software that is limited in functionality ("worse") is often preferable ("better") because it tends to be easier to use and maintain.
- **Support interactivity**. Write individual tools so that they work well with other tools.
- **Handle input/output streams**. Feeding one tool's output as input into another tool is a universally desirable feature.

THE LINUX OPERATING SYSTEM FAMILY

Linux is a family of operating systems based on the Linux kernel, the central core of the OS that manages all components in the system. The Linux kernel was developed by Finnish programmer Linus Torvalds in 1991, while he was a student at the University of Helsinki.

FOSS, the GNU Project, and the Unix design philosophy have all played an important role in the development of Linux. Torvalds released the Linux kernel under version 2 of the GPL. Most distributions of Linux add GNU software written by Stallman and other free software components on top of the Linux kernel. In other words, Linux is the first complete operating system family to qualify as FOSS, and like GNU software, it follows the principles of simplicity and modularity set forth in the Unix design philosophy.



Fundamentally, Linux is a combination of the Linux kernel and GNU software.

GNU/LINUX

Because most members of the Linux OS family incorporate GNU utilities along with the Linux kernel, the FSF prefers to call the OS family "GNU/Linux" rather than just "Linux." This supports the idea that the Linux kernel was the "missing piece" to the GNU Project, and gives credit to GNU for its tools and the free software movement itself. However, Torvalds and others disagree with this assessment, and the name "Linux" is by far the most common way of referring to the OS family.

ADVANTAGES OF USING LINUX

To summarize, the following are some of the major advantages of using Linux:

- Its FOSS nature promotes transparency.
- Its design emphasizes simplicity and modularity.
- It is highly customizable.
- It is highly reliable and stable.
- It has strong integration with several major programming languages, like C, C++, Python®, Ruby, etc.
- It places an emphasis on security and privacy.
- Most distributions are free of monetary cost.
- It has a passionate community willing to provide support.

DISADVANTAGES OF USING LINUX

No system is perfect, including Linux. The following are some potential disadvantages:

- It has a sharper learning curve than other general purpose operating systems like Windows® and macOS®.
- Desktop software is not as well-supported as it is in other operating systems like Windows and macOS.
- There is no definitive or official version, which can be confusing to new users.
- With some exceptions, there is no official vendor-provided support.

LINUX DISTRIBUTIONS

As a family of operating systems, there is no official OS called "Linux." Instead, there are distinct members of the family called **Linux distributions**, or **distros**. All Linux distros are based on the Linux kernel; they differ primarily in what additional software they add on top of the kernel to create a fully functional OS, as well as the version of the kernel they run. There are also differences in community, rate of release, and other factors. Choosing a distribution is a matter of identifying which one most closely aligns with your business needs as well as your familiarity with its tools.

LIST OF LINUX DISTRIBUTIONS

There are hundreds of distros available. The following table includes some of the most historic and/or popular ones.

Distro	Notes
Slackware	This is the oldest distro that is still actively maintained, initially released in 1993. It is intended to be the most "Unix-like" distro and is most appropriate for advanced Linux users.
Debian	This is another early distro, and one that is composed entirely of free software. It is also the basis for many other derivative distros.
Ubuntu®	This is one of the most popular distros for general use and is based on Debian.
Kali Linux	This Debian-based distro is used for penetration testing and other cybersecurity tasks.
Red Hat® Enterprise Linux® (RHEL)	Unlike most distros that are available free of charge, RHEL comes with paid customer support provided by Red Hat, Inc. This distro is common for servers in corporate environments.
CentOS®	This distro is essentially the no-cost version of RHEL. The functionality is nearly identical, but no customer support is provided.
Fedora®	This distro is sponsored by Red Hat and is the upstream source of RHEL. It has multiple editions that each target different computing roles, and tends to run the latest version of the Linux kernel.
openSUSE	This distro is sponsored by SUSE Linux GmbH and targets several computing roles. It strives to incorporate only tools that qualify as FOSS.
SUSE® Linux Enterprise Server (SLES)	This distro shares a code base with openSUSE, but is intended to be more stable than openSUSE. Like RHEL, SLES comes with paid customer support.
Arch Linux™	This distro is targeted at experienced users and focuses on simplicity as its driving philosophy. Its online documentation repository, ArchWiki, is one of the most comprehensive sources of Linux documentation.

Note: *Many Linux distros include proprietary software and are not entirely FOSS.*

Note: *For more information about Linux distros, visit* ***https://distrowatch.com****.*

MORE ON CENTOS

The CentOS Linux distribution is a stable, predictable, manageable, and reproducible platform derived from the sources of RHEL. CentOS is maintained by the CentOS Project, a community-driven free software effort that has its own governing board. The members of the CentOS Project work independently of the RHEL team. However, CentOS benefits from Red Hat's ongoing contributions and investment, and the CentOS trademark is owned by Red Hat.

This course uses CentOS 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 does not contain Red Hat's product or certifications, although it is built from the same sources as the upstream enterprise products. More details about this are available in the CentOS FAQ here: **https://wiki.centos.org/FAQ/General**.

For production environments, the licensed and fully supported RHEL product is recommended.

USES FOR LINUX

One of the main advantages of Linux is that it is highly extensible. As a result, Linux has been applied to many different computing roles. The following table describes these roles.

Role	Notes
Servers	Servers provide an array of services to multiple users over a network. Market share figures for servers are not always easy to pin down, but most sources agree that Linux dominates the server market. More than 90% of the top one million domains are run on Linux web servers, and more than 90% of public cloud servers on the Amazon Elastic Compute Cloud (EC2) platform run Linux.
Workstations	Workstations are more powerful versions of a home desktop and are typically geared toward technical work such as software development. Linux is more common on workstations than it is on PCs, but it still lags behind Windows and macOS in this area.
Mainframes and supercomputers	Mainframes are large, highly advanced computers that excel at processing many transactions at once. Supercomputers are machines at the cutting edge of computing technology that excel at performing complex calculations as quickly as possible. Linux dominates both of these markets—the top 500 fastest supercomputers in the world as of 2018 all run Linux.
Mobile devices	Mobile devices typically include smartphones and tablets, or any device that is small enough to fit in the hand. The mobile operating system Android™ is based on the Linux kernel and has over 80% of the global market share for mobile devices.

Role	Notes
Personal computers (PC)	This typically includes desktops and laptops, or any non-mobile device that is geared toward non-technical home use. This is one of the few markets where Linux has seen minimal penetration; around 2% of users run Linux as a desktop platform.
Embedded systems	Embedded systems are computer hardware and software systems that have a specific function within a larger system. These larger systems can include everything from home appliances like microwaves to large industrial machines. Embedded systems running Linux comprise the majority of the market.

Topic B

Enter Shell Commands

The design of the Linux operating system emphasizes a particular kind of user interface; one in which the user types text commands into a prompt in order to interact with the system. This differs from the primarily visual operating systems like Windows and macOS. Therefore, one of the most crucial skills in Linux administration is becoming comfortable at entering text commands. In this topic, you'll enter a few basic commands to become more familiar with the process.

THE CLI

The **command-line interface (CLI)** is a text-based interface between the user and the operating system that accepts input in the form of commands. The CLI presents a command prompt to the user, and the user enters a command to interact with the system in a variety of ways. Working at the command-line is an important tool in any administrator's arsenal. Developers and administrators often use the CLI, whereas regular users will typically rely on a graphical user interface (GUI).

Comfort at the command-line is essential for administrators. Command-line administration is an assumed skill in Linux. The GUI is not a required component of Linux. In fact, Linux includes many GUIs. Any or none of those may be installed on the system. Assuming you want to maximize the use of hardware for the system's specified purpose, you can perform all administration at the CLI.

Note: *In Linux, the CLI is case-sensitive.*

CLI ADVANTAGES AND CHALLENGES

Some advantages to using the CLI include:

- It's faster for the system to process.
- It's faster for administrators to enter information.
- Commands can be stored in text files called scripts that you can execute with one command, resulting in a long series of activities by the system.
- Scripts can be scheduled to execute on a regular basis.
- Additional options are available in the CLI that may not be present in the GUI.

Likewise, there are some challenges to using the CLI:

- It's more difficult to learn than a GUI.
- Commands have many options and are not always consistent.
- It's often mistakenly thought of as legacy.
- There are many command-line environments among the Linux, Unix, macOS, and Windows platforms.

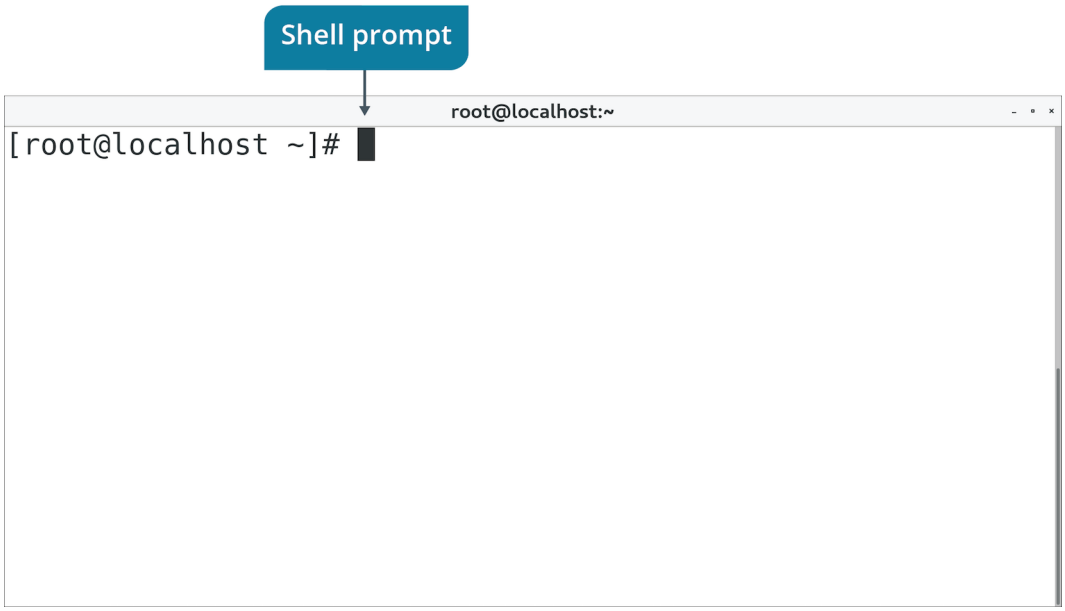
SHELLS

A **shell** envelops the core portion of the operating system—referred to as the kernel—permitting the user to pass commands and information to the kernel. The kernel is also able to respond back to the user via the shell. The shell can be thought of as an

interpreter between human and kernel languages. Linux users issue commands in the shell to tell the operating system what to do. The operating system responds back to the user with the results.

A shell can be implemented as either a CLI or a graphical user interface (GUI). The following table lists some common CLI-based shells that are used in Linux.

Shell	Description
Bourne shell (`sh`)	This is the original Unix shell that is still available on Linux systems, though not widely used.
Bash (`bash`)	This is the default Linux shell and a replacement for the Bourne shell. Its full name comes from the term *Bourne-again shell*.
C shell (`csh`)	This shell is based on the C programming language and was designed to support C language development environments.
KornShell (`ksh`)	This shell uses the features of the C shell with the syntax of the Bourne shell. It is common on Unix systems.



A shell prompt in a GUI.

MORE ON BASH

As a Linux user, it is essential to be comfortable using the default Bash shell. Virtually every Linux distribution will use this shell as the translator between the user and the system. It is possible to install and use other shells in Linux if users are more comfortable with them, but the default will almost always be Bash.

Some characteristics of Bash include:

- It's been around for a very long time, so it is well documented with many existing scripts.
- It's commonly used in Linux and macOS (where it is also the default) and with various other operating systems.
- It's not always consistent, since there have been a great many modifications by various groups since its creation.
- It includes history and tab completion features.

- It's very flexible.

BASH SYNTAX

Command-line administration includes the idea of "syntax," or the proper way of structuring a command and any supporting information. The many CLIs have their own unique ways of entering information. You need to understand this syntax to be able to effectively communicate with the interface.

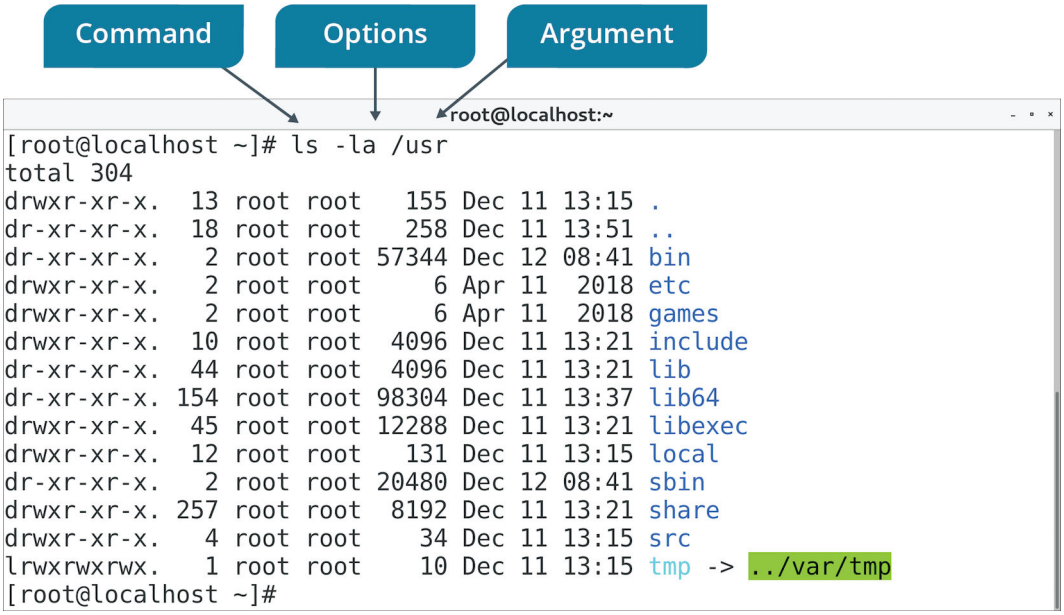
Bash shell syntax contains three main components: the command, options to modify the command, and an argument for the command to act upon. It is common for new users to Bash to forget the spaces between the three components.

The basic syntax of Bash is therefore: `command [-options] [arguments]`

Note: *For command syntax, this course uses the convention by which square brackets denote an optional component and curly brackets denote a required component. In some cases, a command can be run without options or arguments and still work.*

The following table lists an example of each type of basic syntax format using the `ls` command.

Syntax Format	Command	Description
Command only	`ls`	Lists directory contents with default output.
Command with options	`ls -la`	Lists directory contents in long format (`-l`) and showing "hidden" files (`-a`).
Command with an argument	`ls /var/log`	Lists directory contents of `/var/log` directory with default output.
Command with options and an argument	`ls -la /var/log`	Lists directory contents of `/var/log` directory in long format and showing "hidden" files.



The ls -la command displaying the list of files in the /usr directory.

ERRORS

If you fail to enter a command in the proper syntax, Bash will return an error. Typically, these error messages are descriptive and will help you to understand what Bash expects. For "command not found" errors, check for typos in the command. For "no such file or directory" errors, check for typos in the directory, file, or file path names.

BASIC BASH COMMANDS

There are several commands that are important for accomplishing basic tasks. Many of these commands will be covered in greater depth throughout the course but are included in this topic to provide basic command vocabulary and to get you started with hands-on practice at the Bash shell.

Command	Description	Examples
`echo`	Repeats input back to the user on the screen. Commonly used to send information to the user in a script.	`echo 'Good morning!'` returns "Good morning!" at the CLI.
`ls`	Lists the contents of a directory. Can be given options to view permissions, hidden files, etc.	• `ls` lists contents of current directory. • `ls -a` includes hidden files. • `ls -l` outputs in long format. • `ls /var/log` lists contents of specified directory.
`pwd`	Displays the current working directory you are in.	`pwd` returns the path to your current working directory.
`cd`	Changes your current working directory.	• `cd /var/log` changes your current directory to `/var/log` • `cd /etc` changes your current directory to `/etc`
`touch`	Updates timestamp on an existing file, but can also be used to create an empty file.	`touch file1` updates the timestamp on `file1` if it exists; creates `file1` if it doesn't.
`cp`	Copies a file or directory to another location.	`cp file1 file2` copies the contents of `file1` to `file2`.
`mkdir`	Creates a directory.	`mkdir newdir` creates a new directory called `newdir`.

FILE VIEWING COMMANDS

Linux system configurations are held in text files, so you'll need to be able to view the contents of those files.

The `cat` command is used to view the contents of a file without the option to edit that file. An example of using `cat` is `cat file1` to show the contents of `file1` on the screen.

The `less` command is used to view the contents of a file when those contents won't fit entirely on one screen. This command breaks the content output into pages that you can scroll through at the CLI. An example of using less is `less file1` to break

the contents of `file1` into multiple pages when its contents are lengthy enough to go past a single screen. Press **Page Up** and **Page Down** to scroll screens, and press **q** to exit the command.

FILE EDITING COMMANDS

Just as you'll need to view the contents of text files, you'll also need to edit them.

Command	Description	Examples
`vim`	Starts a powerful text editor and the default for Linux.	1. `vim file1` to open a text file in command mode. 2. Press **i** to enter insert mode. 3. Press **Esc** to leave insert mode. 4. `:wq` to save the file and quit.
`nano`	Starts a simple, user-friendly text editor. It may not be installed on all distros.	1. `nano file1` to open a text file. 2. Enter text directly in the interface. 3. Press **Ctrl+O** to save changes. 4. Press **Ctrl+X** to quit.
`gedit`	Starts a GUI text editor that is easy to use. Requires a desktop environment to be installed.	1. Select **Applications→Accessories→Text Editor**. 2. Enter text directly in the interface. 3. Use the menu to save and quit.

POWER MANAGEMENT COMMANDS

Periodically, it may be necessary to reboot or shut down the system. There are several commands to accomplish this, but for now you will focus on the `shutdown` command. Some examples of the `shutdown` command include:

- `shutdown -h now` shuts down the system with no time delay.
- `shutdown -h -t 90` shuts down the system in 90 seconds.
- `shutdown -r now` reboots the system with no time delay.

Note: *The `reboot` command essentially performs the same task as the `shutdown -r now` command.*

THE sleep COMMAND

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

SUPERUSER COMMANDS

In Linux, the user with administrator credentials is the superuser. The superuser is typically named root. It is generally a bad practice to log onto the system as the superuser, and you should get in the habit of logging in with a non-privileged account. However, there will be times when you need to assume the privileges of the superuser in order to perform an administrative task.

The `su -` command ("substitute user") switches user credentials, and `su - root` switches credentials to the root user. The system will prompt you to enter the root

user's password for authorization purposes. Once you are logged in as root, you will be able to perform tasks that you were previously unable to.

SHELL HISTORY

The Bash shell keeps a history file of all commands entered. You can reference this file and repeat commands, increasing efficiency and consistency of entered commands.

Examples of using shell history include:

- The `history` command outputs the most recently entered commands in a list format.
- The **Up Arrow** and **Down Arrow** keys cycle through the command history. Press **Enter** to reissue the desired command from the history.

TAB COMPLETION

The Bash shell supports **tab completion**, enabling users to type in enough of a command, file name, or directory name to be unique, then filling in the remainder of the entry. This feature reduces typographical errors and speeds up the entering of commands.

Examples of using tab completion include:

- Typing `his` and pressing **Tab** will automatically fill the rest of the `history` command.
- Typing `cd /home/user1/Aug` and pressing **Tab** will automatically fill the directory path to `cd /home/user1/AugustProjects` assuming such a directory already exists.

SHELL TIPS AND TRICKS

While the command-line interface can be intimidating, there are several ways of making it easier to work with and more efficient. As you get more comfortable with Bash, you may find you prefer working at the CLI much of the time.

Here are a few tips to help make working at the command-line easier:

- **Tab completion**: Get in the habit of using tab completion for speed and to minimize typos.
- **Use history instead of re-writing long commands**: When you make a typographical error in a command or file name, do not manually re-type the entire line. Repeat the line with the mistake by hitting the **Up Arrow** key one time, and then use the **Left** and **Right Arrow** keys to move to the mistake so that you can correct it.
- **Read the command backward**: When troubleshooting your commands, start from the right and read to the left. This method makes it a great deal easier to notice missing or duplicate characters.
- **Clear the screen**: Enter the `clear` command to clear the CLI of all text. This is useful when you're starting a new task and want to eliminate any distracting information from past command entries.

HOW TO ENTER SHELL COMMANDS

Use the following procedures to enter shell commands.

ENTER SHELL COMMANDS

To enter shell commands:

1. If necessary, sign in with an account on the system.

2. If necessary, open a terminal.
3. Type the desired command at the prompt:
   - Type the command by itself in the form `command`
   - Type the command with the desired options in the form `command [options]`
   - Type the command with the desired options and arguments in the form `command [options] [arguments]`
4. Press **Enter** to issue the command.
5. Observe the results.

USE TAB COMPLETION

To use tab completion:

1. At a terminal, begin typing the desired command. To complete the command immediately, you must type enough of the command for it to be unique.
2. Press **Tab** to fill in the rest of the command, path, or file.
3. If necessary, press **Tab** again to see all paths and files that match the text you've typed so far.
4. If necessary, continue typing and pressing **Tab** to narrow down the desired command, path, or file.
5. When the appropriate command, path, or file is listed, press **Enter**.

Topic C

Get Help with Linux

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

LINUX DOCUMENTATION

Documentation is a necessity in any major computing project, and Linux is no different. Documentation helps users of the Linux operating system, no matter their role or experience level, to perform a wide range of activities and resolve a wide range of issues. However, just like there is not one official form of Linux, neither is there a single, authoritative source of documentation. Documentation is spread across multiple sources that appear in multiple forms, each one suited to a particular context.

Some of major sources of Linux documentation include:

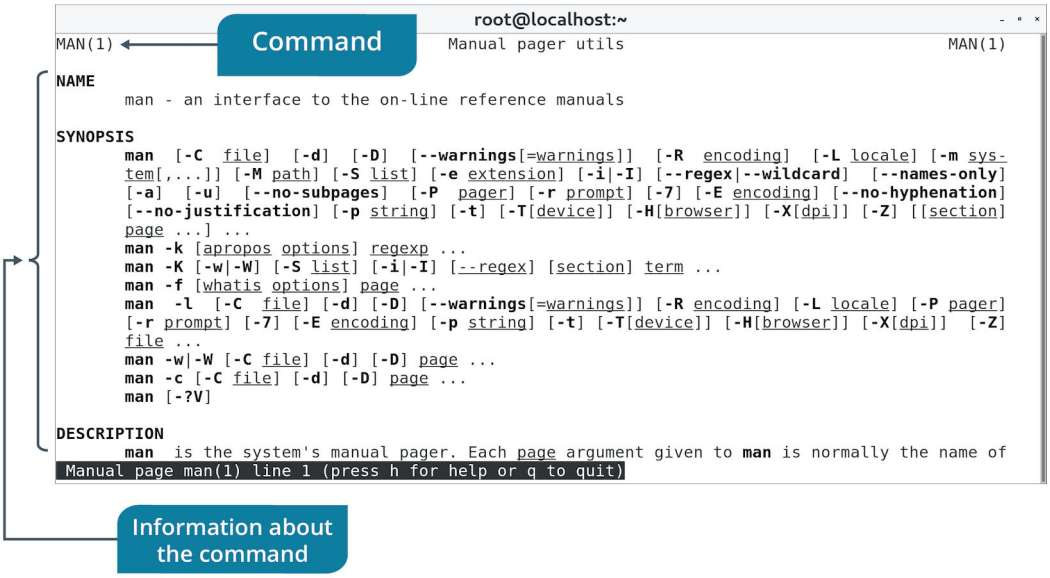
- Manual pages
- Built-in help commands
- Online documentation projects
- Usenet newsgroups
- Internet mailing lists
- Question and answer websites
- Forums and social media
- Books and other print resources

MANUAL PAGES

Linux **manual pages**, or man pages, contain the complete documentation that is specific to each Linux command. The man pages are available on Linux systems by default. The man page for a specific command is displayed using the `man` command. 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.

Man pages are perhaps the most immediate source of help available when you need to learn more about what a command does or how to operate it. They are especially useful in situations where Internet access is not available. However, man pages can be difficult to parse for someone not familiar with how they're formatted.

Note: *Man pages are presented in simple ASCII text format for ease of access.*



The man page for the man command.

SYNTAX

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

SYNOPSIS FORMAT

Most of the components of a man page are self-explanatory, but the **SYNOPSIS** component can be somewhat confusing to new users. This part of a man page provides the syntax of the command along with some example use cases. These use cases are formatted as such:

- **bold** text should be typed exactly as shown.
- *italic* text should be replaced with the appropriate argument. Note that this may be formatted differently on certain systems, like underlined text or colored text.
- `[-abc]` indicates that all arguments within the brackets are optional.
- `-a|-b` indicates that the arguments on either side of the pipe (`|`) cannot be used together.
- *italic* text with `...` (ellipsis) after it indicates that the argument can be repeated.
- `[italic]` text with `...` after it indicates that the entire expression within the brackets can be repeated.

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.

Option	Description
`-t`	Formats the man pages to enable printing.

MAN PAGE SECTIONS

Man pages for commands may be listed under one or more sections. A section defines what category the command belongs to. 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 provided 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 `/dev`)
5	File formats and conventions
6	Games and screensavers
7	Miscellaneous
8	System administration commands and daemons

MAN PAGES NAVIGATION

You can navigate through Linux man pages using a number of keys. These keys are described 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.

OTHER BUILT-IN HELP OPTIONS

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

Help Option	Description
`apropos`	Searches the **NAME** section of all man pages for the keyword that the user provides. The **NAME** section usually contains a brief, one-sentence description of the command after the name itself. The `apropos` command is therefore useful when you'd like to perform some task, but don't know the name of the appropriate command(s). The syntax of this command is `apropos {keyword}`

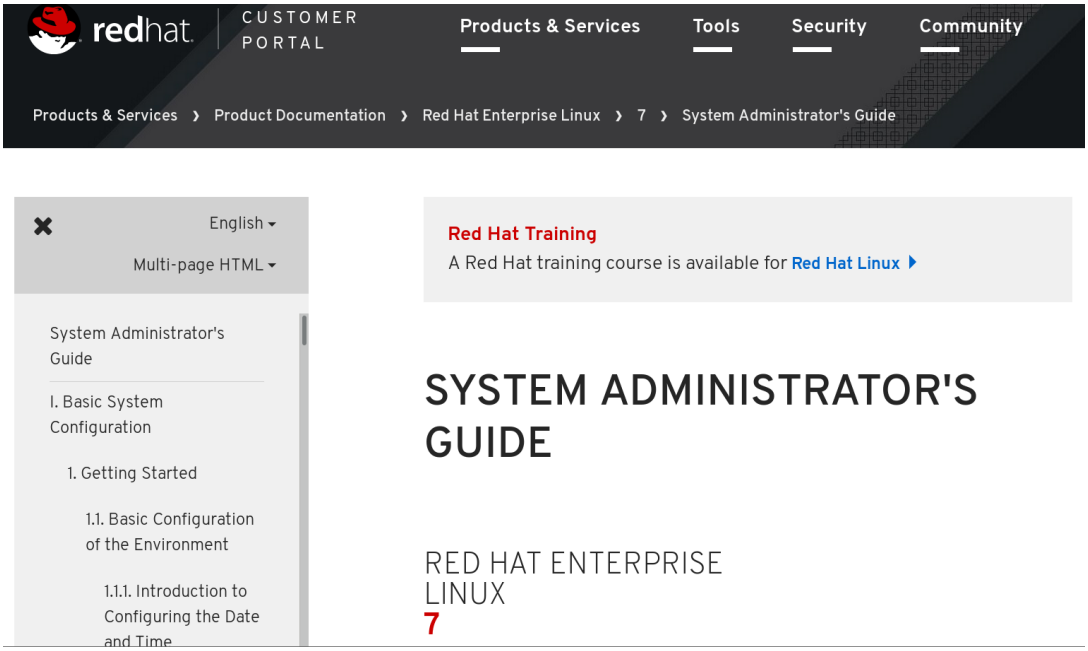
Help Option	Description
`whatis`	Displays the brief description of the given command, including commands that appear in multiple sections. This is essentially the opposite of `apropos`—you know the command name, but want to figure out what it does. The syntax of this command is `whatis {command}`
`info`	Displays the info page of a command. Info pages are essentially an alternative to man pages, and are favored by the GNU project. Although they have features such as hyperlinking and can easily generate HTML pages and PDF files, info pages are not as widely adopted as man pages. The syntax of this command is `info {command}`
`--help`	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 `<command> --help` or `<command> -h`
`/usr/share/doc/`	This directory contains documentation for libraries, system utilities, and other software packages installed on the system. Each software package has its own subfolder in this directory, and each subfolder usually contains separate documents that list the authors, licensing information, installation instructions, general README, etc.

ONLINE DOCUMENTATION

The Internet is one of the best places to go to for help with Linux. There is not one single online source that is necessarily better than the others; the choice often comes down to what type of help you're looking for. The following table lists some of the best online sources for Linux documentation.

Online Source	Description
Official distro documentation	Some of the more popular Linux distributions provide their users with documentation that applies specifically to that distro. This documentation often includes the release notes of different versions and updates, an installation guide, a virtualization guide, and more. For example, the official documentation for Ubuntu is available at **https://help.ubuntu.com/**.
Linux Documentation Project (LDP)	The LDP is an all-volunteer project that provides a comprehensive resource for Linux documentation. The documentation comes in many forms, from HOWTOs that are step-by-step procedures for performing a specific task in Linux, to Guides, which include suggestions and best practices on a variety of Linux topics. The LDP also includes FAQs, man pages, and other types of documentation. The LDP is available at **https://www.tldp.org/**.
Linux man-pages project	This is the official source of man pages for Linux kernel and C library interfaces that are available to user space programs, i.e., code that runs outside of the kernel. This project is available at **https://www.kernel.org/doc/man-pages/**.

Online Source	Description
GNU coreutils manual	This is the official source of documentation for the GNU core utilities, or coreutils. The coreutils is the package of GNU software tools that is compiled with the Linux kernel, along with other software, to form a Linux distribution. This manual is available at **https://www.gnu.org/software/coreutils/manual/**.



The online documentation for the RHEL 7 distribution.

INTERACTIVE HELP

Online documentation is a quick and easy reference point, but it's not always the best source for answering your Linux questions, especially if those questions are complex or apply to unconventional scenarios. These questions are often best answered by interacting with other people directly, whether in real-time or asynchronously. The following table lists some of the major sources of interactive help on the Internet for Linux issues.

Online Source	Description
Usenet newsgroups	Usenet newsgroups are online discussion repositories similar to bulletin board systems, and are some of the oldest forms of Internet-based discussion. Users post text or files to a newsgroup, and other members of the group can view and respond to these posts, which are organized in a threaded manner. Newsgroups tend to be focused on specific subject matter; some newsgroups focused on providing Linux help include `comp.os.linux.help`, `comp.os.linux.answers`, and `comp.os.linux.admin`.

Online Source	Description
Mailing lists	Internet-based mailing lists are similar to newsgroups in that they involve threaded discussions among members of a specific community. "Posts" are essentially email messages sent to every user on a distribution list. You can find distro-specific mailing lists for several major distros, or you can subscribe to more general Linux mailing lists, such as those listed at **https://lists.linuxfoundation.org/mailman/listinfo**.
Q&A websites	Question and answer websites enable users to post a question on some topic, and other users can answer or otherwise comment on the question. Many of these sites have functionality for users to "like" or "upvote" questions they feel are useful, or answers they feel offer a good solution. The most popular Q&A site for IT topics is Stack Exchange, which has a Unix & Linux category available at **https://unix.stackexchange.com/**.
Forums and social media	Internet forums are typically threaded posts, similar to newsgroups or mailing lists, but use web protocols to deliver the content. Some forums encourage users to post questions to the community, and other users respond to the posts with guidance. Aside from distro-specific forums, one popular forum for Linux support is available at **https://www.linuxquestions.org/**. In addition, some social media sites, like Reddit, offer a forum-like discussion platform for soliciting help on specific topics—for example, **https://www.reddit.com/r/linuxquestions/**.

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. Navigate through the man pages:
   - Press the **Up Arrow** or **Down Arrow** keys to navigate within a page.
   - Press **Page Up** or **Page Down** keys to navigate through several pages.
   - Search for some specific topic in the man pages as required.
     a. Enter `/<search string>` to search through the man page for the specified string.
     b. If necessary, press **n** to locate the next occurrence of the string in the man page.
4. Press **q** to close the man page for the specified command.

FIND THE APPROPRIATE COMMAND

To find the appropriate command:

1. If necessary, with root user privileges, enter `mandb` to update the database. This is necessary if you receive a "nothing appropriate" result after running the `apropos` command.
2. Enter `apropos {keyword}` where `{keyword}` is the keyword that describes the task you are trying to perform.
3. Examine the list of results for an appropriate command.

IDENTIFY THE PURPOSE OF A COMMAND

To identify the purpose of a command:

1. Enter `whatis {command}` where `{command}` is the command you want to identify the purpose of.
2. Examine and navigate through the results.

DISPLAY THE INFO PAGES OF A COMMAND

To display the info pages of a command:

1. Enter `info {command}` to read the info documents for the specified command.
2. Examine and navigate through the results.

DISPLAY THE OPTIONS OF A COMMAND

To display the options of a command:

- Enter `command --help` to display the command syntax and a list of options.

Activity 1-1

Performing Basic Linux Tasks Review

SCENARIO

Answer the following review questions.

1. **How is Linux used in your organization? Are there any existing systems that could benefit from switching to Linux?**

2. **What is your preferred method for getting help with Linux, and why?**

Summary

In this lesson, you identified the fundamental concepts behind the Linux operating system, as well as performed basic tasks to get you started in a Linux environment. This is the foundation on which you will build your skills in managing Linux systems.

ISBN-13 978-1-6427-4155-1
ISBN-10 1-6427-4155-8