CompTIA Linux+

01- Introducing Linux

Ahmed Sultan
Senior Technical Instructor
ahmedsultan.me/about

Outlines

1.1- Identify Linux Characteristics

1.2- Understand Bash Interaction with Linux

1.3- Use Help in Linux

CHARACTERISTICS OF FREE AND OPEN-SOURCE SOFTWARE

- Much of today’s software is closed-source software, or proprietary software that’s released under copyright law.
- These laws restrict intellectual property, such as closed-source software, from duplication and re-release by competitors.
- Open-source software takes a different approach.
- In general, software licensed as open-source can be duplicated, shared, and modified, and the modified versions can be released to consumers.
- Code licensed as **free and open-source software (FOSS)** can be used and changed without cost.

THE HISTORY AND PHILOSOPHY OF UNIX AND LINUX

- **Unix** is one of the oldest operating systems still in use.
- It was created in **1969**, and it was not released as open-source software.
- Instead, Unix versions were associated with many different tech organizations, including **IBM**, **Hewlett-Packard**, and **AT&T**.
- These various Unix versions are referred to as Unix “flavors" and were proprietary to each company.

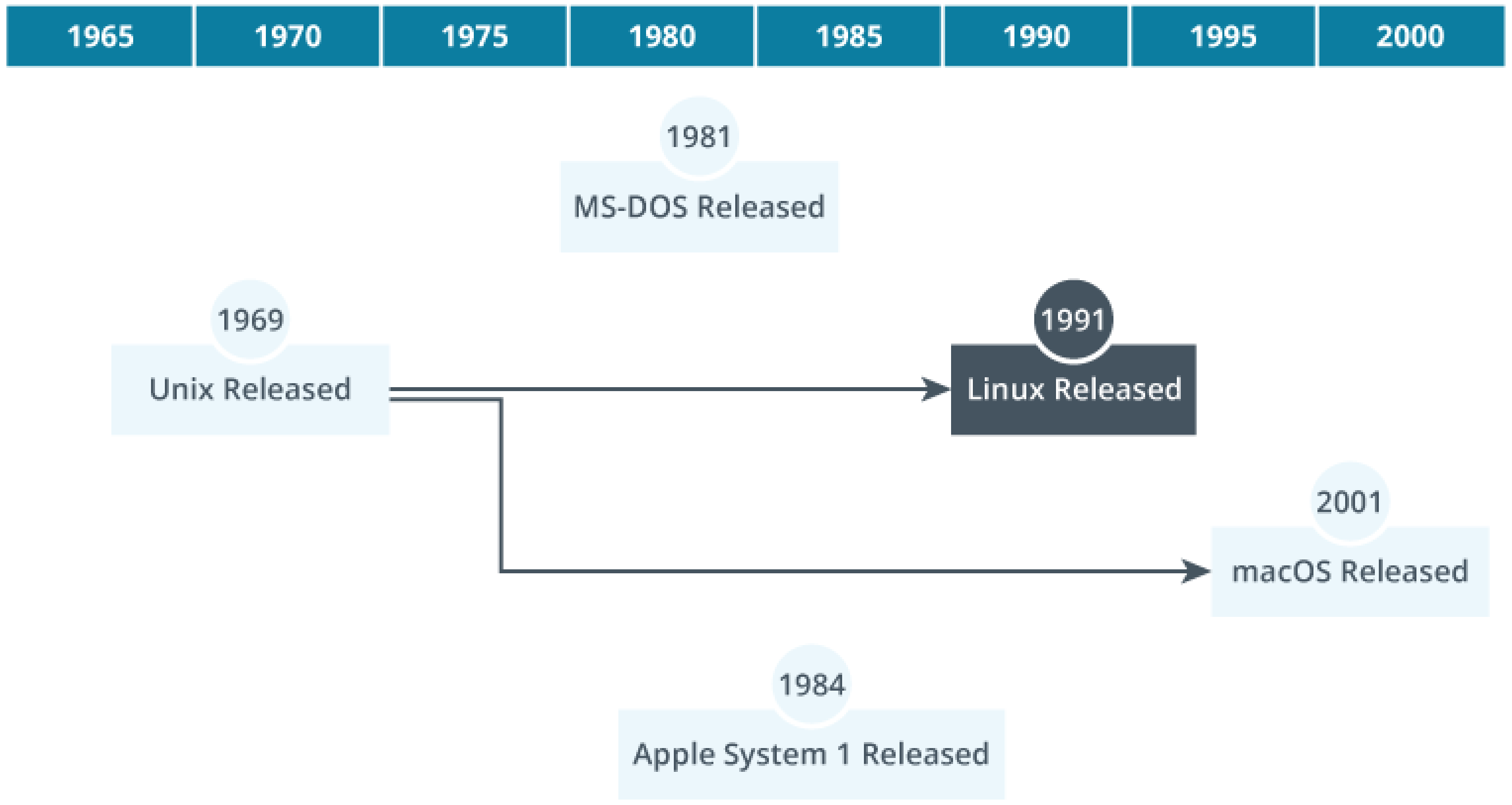
THE HISTORY AND PHILOSOPHY OF UNIX AND LINUX (cont.)

- In **1991**, Linus Torvalds created a new Unix-like operating system kernel.
- He released this kernel, which he called **Linux**, under the GPL license.
- The Linux kernel, as well as much of the software released with it, is open-source; it can be modified, shared freely, and re-released.
- This collaborative approach allows Linux to grow and evolve rapidly.
- As a result of this approach, there are now more than 200 Linux versions, or **distributions** (abbreviated "**distros**").

THE HISTORY AND PHILOSOPHY OF UNIX AND LINUX (cont.)

- Of the **three** primary operating systems in the marketplace today (**Linux**, **macOS**, and **Windows**), two can trace their roots back to Unix.
- The **macOS** kernel evolved from a Unix flavor named **BSD** and shares many of the same standards and some software as Linux.
- However, Apple's OS is not **FOSS**.
- Microsoft Windows also uses a proprietary kernel with a more restrictive licensing method.

THE HISTORY AND PHILOSOPHY OF UNIX AND LINUX (cont.)



TRAITS OF THE LINUX OPERATING SYSTEM

- Like any operating system, Linux has characteristics that may or may not fit the needs of a given organization.
- Here are a few general considerations:
  - ✓ Free: No licensing fees or tracking associated with most Linux distributions.
  - ✓ Security: Because of the open-source nature of Linux and its associated software, many developers can and do review code for vulnerabilities. Such vulnerabilities tend to be addressed quickly.
  - ✓ Support: Community-driven support may provide easy, efficient, and cost-effective solutions. However, support may be limited to the community, without a strong corporate support structure implemented by the distribution’s vendor.

TRAITS OF THE LINUX OPERATING SYSTEM (cont.)

- Here are a few general considerations (cont.)
  - ✓ Performance: Linux often provides greater performance and stability compared to other operating systems.
  - ✓ Hardware requirements: Linux may consume fewer hardware resources, making it easier to retain older systems for longer.
  - ✓ Hardware flexibility: Linux runs on a wide variety of hardware platforms, adding to its flexibility in areas such as Internet of Things (IoT). Specialized hardware may require specific drivers that may not exist for Linux.
  - ✓ Distribution creation: If existing Linux distributions do not fit your needs, you are welcome (and encouraged) to create your own.

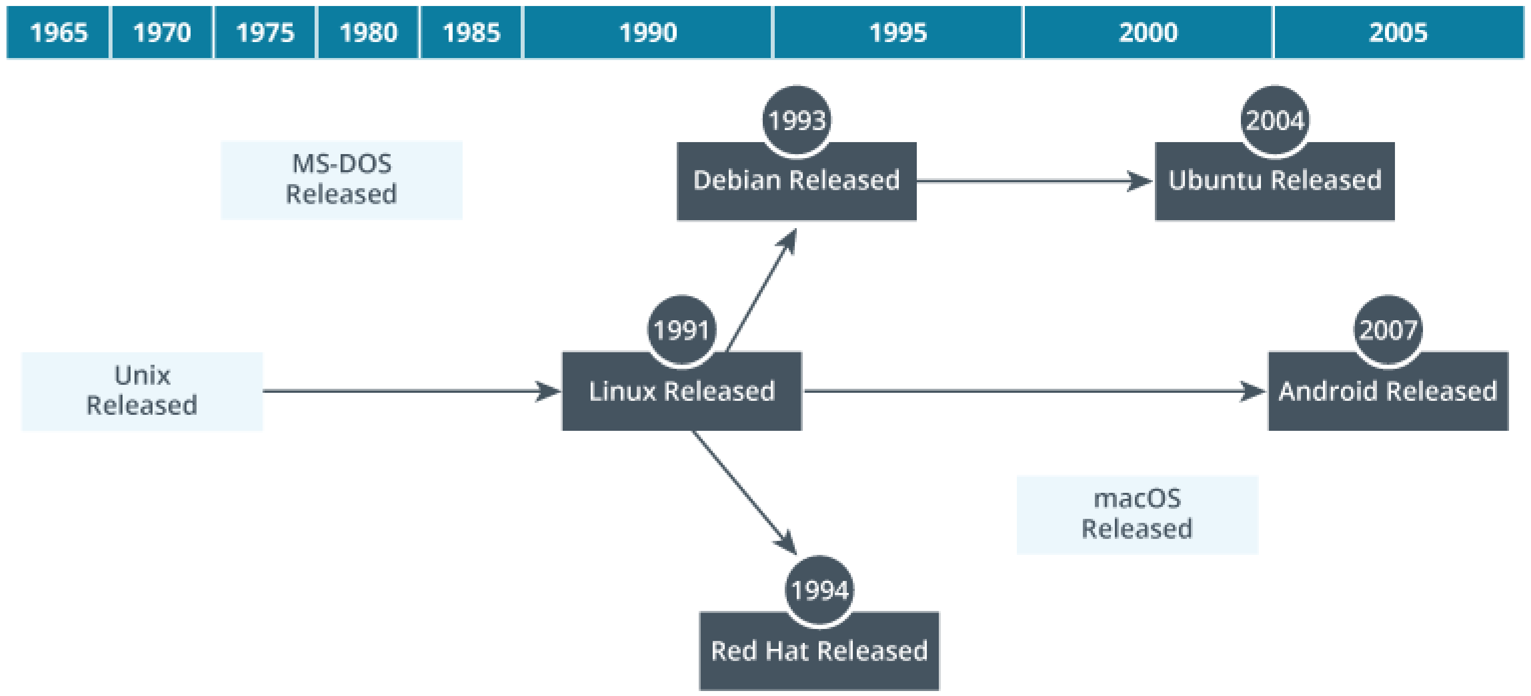
UNDERSTAND LINUX DISTRIBUTIONS

- Because anyone can create and release their own version of Linux, there are thousands of different options.
- These individual releases are called **distributions** (or "**distros**" for short).
- Distributions are purpose-specific versions of Linux that address a specific need, such as system security or application hosting.
- Many distributions trace their history back to one of two specific Linux distributions: **Red Hat Linux** or **Debian Linux**.
- One of the main differentiators between these two distros is how they manage software.
- Those distros derived from Red Hat Linux use different software managers than those derived from Debian Linux.

UNDERSTAND LINUX DISTRIBUTIONS (cont.)

- Some of the most common distros include:
  - ✓ Fedora Linux
  - ✓ Ubuntu Desktop, Server, Core
  - ✓ Red Hat Enterprise Linux (RHEL)
  - ✓ Linux Mint
  - ✓ Debian
  - ✓ openSUSE

UNDERSTAND LINUX DISTRIBUTIONS (cont.)



UNDERSTAND LINUX DISTRIBUTIONS (cont.)

- Many of these distributions fulfill specific roles in the marketplace, including desktop or workstation computer, server, IoT device, mobile device, or other functions.
- While mobile and IoT implementations are common, the focus of this course is on server deployments.
- Some distributions contain end-user applications, such as word processors or presentation software.
- Others contain server services, such as web services or file storage.
- Still other distributions include security software or creative applications, such as music editing.

UNDERSTAND LINUX DISTRIBUTIONS (cont.)

- Linux server deployments are put to use in the following ways:
  - ✓Webserver: Hosts one or more websites.
  - ✓Name resolution: Hosts Domain Name System (DNS) name resolution services.
  - ✓File: Stores business data, usually in some form of text document.
  - ✓Print: Manages the print process and access to print services.
  - ✓Log: Centralizes and stores log files from other systems.
  - ✓Virtualization/container: Hosts virtual machine or container virtualization software.
  - ✓Database: Hosts one or more databases.
  - ✓Cluster: Works with other cluster nodes to host high-performance, fault-tolerant services.

THE COMMAND-LINE INTERFACE (CLI)

- One distinguishing characteristic of Linux compared to other operating systems is its reliance on the **command-line interface (CLI)**.
- Linux administrators frequently use the **CLI** for everyday tasks, while administrators of other platforms often use graphical user interface (GUI) utilities.
- A GUI consumes a great many hardware resources, specifically memory and processor time.
- On a server, these resources should be dedicated to the service provided, such as handling database queries or managing print jobs.
- Desktop systems might need a user-friendly GUI but servers usually do not.

THE COMMAND-LINE INTERFACE (CLI) (cont.)

- CLI advantages:
  - ✓ Quicker: It’s usually quicker to execute a series of commands at the CLI (assuming you know the commands).
  - ✓ Performance: CLI environments consume fewer hardware resources, leaving those resources free to support the server’s purpose.
  - ✓ Scriptable: CLI commands can be written into a text file, which the system then reads and executes in a consistent, efficient, repeatable, and scheduled manner.

THE COMMAND-LINE INTERFACE (CLI) (cont.)

- Command-line interfaces are available in Linux, Windows, and macOS.
- Users type commands using a specific syntax, and the system processes the commands.
- At first, such input may seem intimidating or difficult, but CLI environments get easier with use.
- These environments are usually faster and offer automation options that are not available in GUIs.

THE COMMAND-LINE INTERFACE (CLI) (cont.)

- Shells provide the CLI.
- Each shell has its own syntax, or way of structuring commands.
- Common Linux shells:
  - ✓Bash: Default Linux shell
  - ✓ksh: Korn shell
  - ✓zsh: Z shell

COMMON GUIs

- Just as there are many different Linux distributions, there are also many different Linux graphical environments.
- Windows and macOS users have one GUI available to them—whatever graphical environment Microsoft and Apple choose to provide.
- Linux users have the freedom to install zero, one, or many GUI environments and switch between them.

COMMON GUIs (cont.)

- These GUIs are usually distinguished by two characteristics: **user-friendly interface** and **performance**.
- Some users like the look and feel of a particular GUI over others.
- In addition, some GUIs consume more processor time and memory than others do.
- Luckily, many options are available in the Linux world.
- Common GUI environments include **GNOME**, **KDE Plasma**, **Cinnamon** , and **MATE** .

1.1- Identify Linux Characteristics

1.2- Understand Bash Interaction with Linux

1.3- Use Help in Linux

COMMAND SHELLS

- The **CLI** is provided by software called a **shell**.
- The shell accepts user input, processes the input for syntax, and provides output back to the user.
- The default shell for most Linux distributions is **Bash**, and this is the shell that sysadmins should be prepared to work with.

COMMAND SHELLS (cont.)

- Other common Linux shells include **ksh**, or **KornShell**, which is common among Unix servers; **Zsh**, or **Z Shell**, with quite powerful scripting capabilities; and **Fish**, or **f**riendly **i**nteractive **sh**ell, an interface that provides a user-friendly experience and web-based configurations.
- By way of comparison, **Windows Server** also uses shells: the traditional, DOS-like **cmd.exe** shell and **Microsoft PowerShell**.
- The current (at the time of this writing) default shell for macOS is the **Zsh**.

BASH CHARACTERISTICS AND SYNTAX

- Commands must be entered into Bash using a specific structure, or syntax.
- Each component of the syntax has a name to make it easier to understand.
- The syntax components are:
  - ✓Command: The primary instruction given to the system.
  - ✓Subcommand: A secondary, more detailed instruction supporting the primary command.
  - ✓Option: A command modifier that slightly changes the way a command is processed.
  - ✓Argument: The object on which the command acts. For example, in a command to delete a file, the argument is the name of the file to be deleted.

BASH CHARACTERISTICS AND SYNTAX (cont.)

There are two basic syntax forms, **normal command** and **command-subcommand**.

BASH CHARACTERISTICS AND SYNTAX (cont.)

- **Normal Command Syntax**
  - ✓ The normal command syntax relies on the **three** primary components of the Bash syntax: **the command**, **options** to modify the command, and an **argument** for the command to act upon.

Normal Command Syntax for the ls Command	Purpose
`ls`	List directory contents
`ls -la`	List all (`-a`) directory contents in long format (`-l`)
`ls /var/log`	List the contents of the `/var/log` directory
`ls -la /var/log`	List all contents of the `/var/log` directory in long format

BASH CHARACTERISTICS AND SYNTAX (cont.)

- **Command-Subcommand Syntax**
  - ✓ Many Linux commands support subcommands to specify particular information that the sysadmin needs.
  - ✓ The sysadmin enters the primary **command**, then follows it with a space and a **subcommand**, and then a space and argument.

Command-Subcommand Syntax for the ip Command	Purpose
`ip addr`	Display all IP addresses for all interfaces
`ip addr show eth0`	Display only IP address information for the eth0 interface
`ip help`	Display basic help about the `ip` command
`ip link help`	Display help about the `ip link` subcommand

USE BASIC BASH COMMANDS

- There are many Bash commands.
- Some of the most often-used commands deal with **file management functions**, such as displaying files and file contents, moving from one directory (or folder) to another, or editing files.
- The upcoming commands exemplify the Bash syntax and enable users to begin working with the files and directories that make up Linux.
- These commands are used throughout this course and will quickly become familiar.

USE BASIC BASH COMMANDS (cont.)

Command	Purpose	Example with Options	Result
`ls`	List the contents of the current directory	`ls /tmp`	List the contents of the /tmp directory
`touch`	Create a new empty file or update the timestamp on an existing file	`touch newfile.txt`	Create a new file named newfile.txt
`cd`	Change from one directory to another	`cd /etc`	Changes the current directory to /etc
`cat`	Display the contents of a text file on the screen	`cat data.txt`	Display the contents of the data.txt file

USE BASIC BASH COMMANDS (cont.)

`less`	Display the contents of a file in windows that fit on the screen	`less data.txt`	Display the contents of the data.txt file screen at a time when the file would not normally fit on one screen
`tree`	Display the directory structure in a tree format	`tree /etc`	Display the subdirectories and files in the /etc directory in a tree structure
`shutdown`	Shut down the system	`shutdown -r now`	Restart the system immediately

USE BASIC BASH COMMANDS (cont.)

- Two common commands do not use options to generate an output.
- Use ***whoami*** to display the current user, and use ***pwd*** to display the present working directory.

USE BASIC BASH COMMANDS (cont.)

- Bash supports **tab completion**.
- Users can type in enough of a command to make it unique from any other command.
- Select the **Tab** key, and Bash automatically completes the command.
- This feature also works with file and directory names.
- Tab completion reduces typographical errors and increases speed at the CLI.

USE BASIC BASH COMMANDS (cont.)

- Bash also keeps a record of previously entered commands in a **history** file.
- This file can be referenced and used to repeat or edit commands.
- The simplest way to work with history is by using the **Up** and **Down Arrow keys**.
- Select the **Up Arrow key** one time to recall the most recently used command.
- You can cycle through the command history by pressing Up Arrow or Down Arrow multiple times.
- Select Enter once the appropriate command is displayed.

USE BASIC BASH COMMANDS (cont.)

- Typing the ***history*** command displays the contents of the history file.
- Each entry in the file is numbered.
- Type **!** and the **command number** executes that command.

INTRODUCING VIM AND NANO

- Linux stores its configurations in text files.
- When a **sysadmin** needs to change system settings, these text files must be edited.
- There are many familiar text editors in GUIs, but what about Linux systems that do not have a graphical interface available, such as Linux server installations?
- Two standard text editors exist that are run from the CLI and do not need a mouse or graphical interface: **Vim** and **Nano**.
- Here is a very brief overview of using these two editors.

VIM

- **Vim** is very powerful and complex. It uses three different modes, where each mode maps keyboard keys to different functions.
- For example, in **Insert mode** the keyboard acts as normal, inserting text into the file.
- If you’re in Insert mode and type "abc," those three characters appear in the file’s content.

VIM (cont.)

- In **Command mode**, pressing a key on the keyboard issues commands to Vim instead of entering text in the file.
- Selecting the ***i*** key tells Vim to switch from **Command mode** to **Insert mode**.
- The third mode is **Execute**.
- This mode is entered by selecting the colon character, ***:*** , and it provides a command prompt to Vim where additional commands can be issued.
- For example, ***:wq*** places Vim in Execute mode, writes the files to the disk (save), and then quits Vim (q).
- The many modes and commands can make Vim a little confusing.
- Strive to understand four basic functions: **create/open**, **edit**, **save**, **close**.

VIM (cont.)

Function	Command	Result
Create/Open	`vim filename`	Create a new empty file, or open an existing file in Vim
Edit	`i`	Enter Insert mode, and begin making edits
Save	`ESC` and `:w`	Move out of Insert mode and into Command mode, and then save (write) changes
Close	`ESC` and `:q`	Move to Command mode, and then exit

NANO

- **Nano** is a popular and common alternative to Vim.
- It’s simpler but less powerful.
- However, in many cases, sysadmins don't need the power offered by Vim, which makes Nano a useful choice for basic editing functions.
- Nano does not have modes.
- Pressing keys on the keyboard inserts text into the file, just as expected with most editors.
- To save and close the file, use keyboard shortcuts using the **Ctrl** meta key.
- For example, ***Ctrl+O*** saves the file, and ***Ctrl+X*** exits the file.

NANO (cont.)

- Some Linux distributions install both **Vim** and **Nano** by default, while others will include only one or the other.
- It is essential for you to be able to use both editors at a very basic level (open, edit, save, close) so that you are capable of editing files with whichever tool is available.

INTRODUCING SU AND SUDO

- There are **three** types of accounts on Linux systems: **root**, **standard user**, and **service**.
- The administrator account in Linux is called **root**.
- Logging in to the system with administrator access is frowned upon.
- The security best practice is to log on with a standard user account, and then, if necessary, switch your user account to root.
- The command to accomplish this is ***su***.
- Type ***su root*** to switch from the standard user to root.
- Type ***exit*** to leave the root user and return to the standard user.

IDENTIFY COMMON DIRECTORIES

- With so many Linux distributions available, administrators rely on the Filesystem Hierarchy Standard to understand the default location of particular resources.
- There are three common directories that administrators work with on a regular basis.
  - # /home/username: Each standard user has a specific and private directory used to store personal files, profile settings, and other data. These user directories are subdirectories of /home.
  - # /etc: Most system configuration files are stored in the /etc directory.
  - # /var/log: Log files for the system and applications are stored in the /var/log directory.

1.1- Identify Linux Characteristics

1.2- Understand Bash Interaction with Linux

1.3- Use Help in Linux

LINUX DOCUMENTATION

- There are several ways of getting help in Linux.
- The most common are the **manual pages**, referred to as "**man pages**" for short.
- There is built-in documentation for the system and some applications, too.
- Many online resources also exist, and they are often the most up to date.
- Because there are so many commands, and because each command has so many options, it’s very common to use the man pages as a quick reference for displaying the available options.

MANUAL PAGES

- It’s common for new Linux users to ask for help and then be asked, “Did you check the man pages?” That’s because man pages are the primary reference for standard Linux commands.
- The man pages provide syntax information and usage examples.
- Perhaps most important, the available options are displayed.
- Because of the number of options for each command, and the fact that many options differ from command to command, the man pages provide an essential quick reference.
- The syntax for using man pages is ***man {command}***
- For example, to display help for the *ls* command, type ***man ls***.

BUILT-IN DOCUMENTATION

- Most commands include help references.
- Add the ***-h*** option, or ***help*** after the command to display this reference material.
- The ***whatis*** command provides a brief description of the specified command.
- The syntax for **whatis** is ***whatis {command}***
- Finally, built-in documentation can be found at **/usr/share/doc**.
- This directory contains some Linux and application help files.
- Not all applications store documentation at this location, but it’s worth checking.

ONLINE DOCUMENTATION

- There is a great deal of information available online that covers Linux administration, applications, security configurations, and network services.
- This documentation may be provided by vendors, community groups, online forums, article repositories, and other sites.
  - ✓ Linux distribution vendors: Vendors such as **Red Hat** and **Ubuntu** have large repositories of reference information.
  - ✓ Linux application vendors: Vendors for products such as **Apache web server**, **Vim**, and **Firefox** provide many references for their applications.
  - ✓ Linux Documentation Project: This is a community project dedicated to providing documentation for Linux, including how-to documents, man pages, and guides.