

UNIT-8

Linux Commands

8.1 THE OPERATING SYSTEM

We use computers freely, but most of us never bother to know what's inside the box. Why should we? We also use TV and never care to find out how this box manages to convert invisible radio waves to real-life colorful pictures. We use spreadsheets and word processors without knowing how these programs access the machine's resources.

Every time you switch on your computer, you see a screen where you can perform different activities like write, browse the internet or watch a video. What is it that makes the computer hardware work like that? How does the processor on your computer know that you are asking it to run a mp3 file?

Well, it is the operating system or the kernel which does this work. A kernel is a program at the heart of any operating system that takes care of fundamental functionalities, like letting hardware communicate with software.

So, to work on your computer, you need an Operating System (OS). In fact, you are using one as you read this on your computer. Now, you may have used popular OS's like Windows, Apple OS X but here we will learn what Linux is and what benefits it offers over other OS choices.

8.2 What is Linux?

Linux is a fast and stable open source operating system for personal computers (PCs) and workstations that features professional-level Internet services, extensive development tools, fully functional graphical user interfaces (GUIs), and a massive number of applications ranging from office suites to multimedia applications.

Linux is a PC version of the Unix operating system that has been used for decades on mainframes and minicomputers and is currently the system of choice for network servers and workstations. Linux brings the speed, efficiency, scalability, and flexibility of Unix to your PC, taking advantage of all the capabilities that PCs can now provide.

Linux can be generally divided into three major components: the kernel, the environment, and the file structure. The kernel is the core program that runs programs and manages hardware devices, such as disks and printers. The environment provides an interface for the user. It receives commands from the user and sends those commands to the kernel for execution. The file structure organizes the way files are stored on a storage device, such as a disk.

Files are organized into directories. Each directory may contain any number of subdirectories, each holding files. Together, the kernel, the environment, and the file structure form the basic operating system structure. With these three, you can run programs, manage files, and interact with the system.

8.3 Linux Commands structure

The basic command structure in Linux follows the following format:

```
command [options] [arguments]
```

Here's a breakdown of each component:

- **command:** The command is the name of the Linux program you want to run, such as "ls" or "cd".
- **options:** Options are optional parameters that can modify how the command behaves. They usually start with a hyphen (-) and can be combined together.

For example, the "ls" command has several options, such as "-l" to display the files in long format and "-a" to display hidden files.

- **arguments:** Arguments are additional pieces of information that the command needs to run. They typically follow the command and options and can be things like file names, directory names, or other parameters.

Here's an example of how this command structure might be used in practice:

```
ls -la /home/user/Documents
```

In this example,

"ls" is the command,

"-la" are the options, and

"/home/user/Documents" is the argument.

This command would display a detailed list of all files (including hidden files) in the /home/user/Documents directory.

8.4 File Commands:

In Linux, file commands are used to manage files and directories, including creating, modifying, moving, copying, deleting, and searching for files.

The most common file commands in Linux are:

- **ls** - list directory contents

The **ls** command in Linux is used to list the files and directories in a given directory.

Here is the syntax for the **ls** command:

```
ls [OPTIONS] [FILE or DIRECTORY]
```

Examples:

1. List all files and directories in the current directory:

```
ls
```

2. List file details in a long for

```
ls -l
```

- **cp** - copy files or directories

The cp command in Linux is used to copy files or directories from one location to another.

Here is the syntax for the **cp** command:

```
cp [OPTIONS] SOURCE DESTINATION
```

Here are some examples of how to use the **cp** command:

1. Copy a file to a new location:

```
cp /path/to/file.txt /path/to/destination/
```

2. Copy a directory and its contents to a new location:

```
cp -r /path/to/directory/ /path/to/destination/
```

- **mv** - move/rename files or directories

The mv command in a Linux is used to move or rename files and directories.

The syntax for the **mv** command is as follows:

```
mv [options] source destination
```

Here, **source** represents the file or directory to be moved, and **destination** represents the new name or location of the file or directory.

Examples:

1. Move a file to a new location:

```
mv myfile.txt /home/user/newfolder/myfile.txt
```

This will move the file **myfile.txt** to the directory **/home/user/newfolder/**.

2. Rename a file:

```
mv oldname.txt newname.txt
```

This will rename the file **oldname.txt** to **newname.txt**.

- **rm** - remove files or directories

The **rm** command in a Linux is used to remove or delete files and directories.

The syntax for the **rm** command is as follows:

```
rm [options] file1 file2 ... fileN
```

Here, **file1**, **file2**, ..., **fileN** represent the files and directories to be removed.

Examples:

1. Remove a single file:

```
rm myfile.txt
```

This will remove the file **myfile.txt**.

2. Remove multiple files:

```
rm file1.txt file2.txt file3.txt
```

This will remove the files **file1.txt**, **file2.txt**, and **file3.txt**.

3. Remove a directory and its contents:

```
rm -r mydirectory
```

This will remove the directory **mydirectory** and all its contents recursively.

- **mkdir** - create a new directory

The **mkdir** command is used to create a new directory (also known as a folder) in a specified location.

The syntax for the **mkdir** command is as follows:

```
mkdir [OPTION]... DIRECTORY...
```

Where **OPTION** is an optional flag that modifies the behavior of the command, and **DIRECTORY** is the name of the directory to be created.

Examples:

1. To create a directory called "example" in the current working directory:

```
mkdir example
```

2. To create a directory called "parent" and a subdirectory called "child" inside it:

```
mkdir parent/child
```

- **rmdir** - remove an empty directory

The **rmdir** command is used to remove an empty directory (folder) in a specified location.

The syntax for the **rmdir** command is as follows:

```
rmdir [OPTION]... DIRECTORY...
```

Where **OPTION** is an optional flag that modifies the behavior of the command, and **DIRECTORY** is the name of the directory to be removed.

Examples:

1. To remove a directory called "example" in the current working directory:

```
rmdir example
```

2. To remove a directory called "parent" and its empty subdirectory called "child":

```
rmdir parent/child
```

- **cat** - concatenate files and print on the standard output

The **cat** command is used to concatenate and display the contents of one or more files in the terminal.

The syntax for the **cat** command is as follows:

```
cat [OPTION]... [FILE]...
```

Where **OPTION** is an optional flag that modifies the behavior of the command, and **FILE** is the name of the file to be displayed.

Examples:

1. To display the contents of a file called "example.txt":

```
cat example.txt
```

2. To create a new file called "example.txt":

```
cat > example.txt  
This is an example text file.  
It has multiple lines of text.  
Press Ctrl+D to save and exit.
```

3. To appending text to a file named "example.txt":

```
cat >> example.txt
```

Your input will be appended to the end of the "example.txt" file.

8.5 Symbolic link commands:

- **ln -s** - create a symbolic link to a file or directory

The **ln -s** command is used to create a symbolic link between files. It creates a shortcut or alias to the target file or directory, which can be accessed with a different name or path.

The syntax for creating a symbolic link using **ln -s** is as follows:

```
ln -s [target_file] [symbolic_link_name]
```

Examples:

Suppose you have a file named **file.txt** in your home directory (**/home/username/**), and you want to create a symbolic link to this file in a different directory, say **/var/www/html/**.

To do this, you can use the following command:

```
ln -s /home/username/file.txt /var/www/html/file-link
```

8.6 Process management commands:

- **ps** - report a snapshot of the current processes

The **ps** command is used to display information about the currently running processes on a Linux system. It can be used to view detailed information about a particular process or to get an overview of all running processes.

The syntax for using the **ps** command is as follows:

```
ps [options]
```

Examples:

Here's an example of how to use the **ps** command to view detailed information about all running processes:

```
ps aux
```

- **top** - display system processes in real-time

The **top** command is a system monitoring tool that displays information about running processes on a Linux system. It provides real-time information about the CPU usage, memory usage, and other system resources.

Here is the basic syntax for the **top** command:

```
top [options]
```

Examples:

To display only the process with a specific process ID, use the following command:

```
top -p pid
```

- **kill** - terminate a running process

The **kill** command is a Linux command that allows you to terminate running processes by sending a signal to them.

Here is the basic syntax for the **kill** command:

```
kill [options] <PID>
```

Examples:

To terminate a process with a specific process ID using the default signal, use the following command:

```
kill 1234
```

8.7 File permission commands:

- **chmod** - change file permissions

The **chmod** command is a Unix/Linux command that is used to change the permissions of files and directories. The command stands for "change mode".

Here is the basic syntax for the **chmod** command:

```
chmod [options] mode file(s)
```

Examples:

To give the owner of a file read, write, and execute permission, use the following command:

```
chmod u+rw file.txt
```

- **chown** - change file ownership

The **chown** command is a Unix/Linux command that is used to change the owner or group of files and directories.

Here is the basic syntax for the **chown** command:

```
chown [options] owner[:group] file(s)
```

The **owner** argument specifies the new owner of the file(s), and the **group** argument specifies the new group of the file(s).

Examples:

To change the owner of a file to a specific user, use the following command:

```
chown user file.txt
```


8.8 "Find" command:

- **find** - search for files in a directory hierarchy based on specific criteria

The **find** command is a Unix/Linux command that is used to search for files and directories in a specified location based on various criteria such as name, size, and modified time.

Here is the basic syntax for the **find** command:

```
find [path] [expression]
```

The **path** argument specifies the starting location for the search, and the **expression** argument specifies the criteria for the search.

Examples:

To find all files with the name "example.txt" in the current directory and its subdirectories, use the following command:

```
find . -name example.txt
```

8.9 Word count commands:

- **wc** - print the number of lines, words, and bytes in a file

The **wc** command is a Unix/Linux command that is used to count the number of lines, words, and characters in a file or set of files.

Here is the basic syntax for the **wc** command:

```
wc [options] [file(s)]
```

Example:

To count the number of lines, words, and characters in a file called **example.txt**, use the following command:

```
wc example.txt
```

8.10 Compression commands:

- **tar** - create, list, and extract tar archives

The **tar** command is a utility in Unix-like operating systems used to create, manipulate, and extract files from archive files.

The basic syntax of the **tar** command is:

```
tar [options] [archive-file] [file or directory to be archived]
```

Some common options for the **tar** command are:

- c: Create a new archive file.
- x: Extract files from an archive.
- v: Verbose output. Prints the list of files being archived or extracted.
- z: Compress the archive file using gzip.

Example:

1. To create an archive file named **archive.tar** of a directory named **mydirectory**, use the following command:

```
tar -cvf archive.tar mydirectory/
```

2. To extract the contents of an archive file named **archive.tar**, use the following command:

```
tar -xvf archive.tar
```

- **gzip** - compress or decompress files using gzip format

The **gzip** command is a utility in Unix-like operating systems used to compress files. The basic syntax of the **gzip** command is:

```
gzip [options] [file]
```

Example:

1. To compress a file named **myfile.txt** and create a new file named **myfile.txt.gz**, use the following command:

```
gzip myfile.txt
```

2. To compress a file named **myfile.txt** and keep the original file unchanged, use the following command:

```
gzip -c myfile.txt > myfile.txt.gz
```

- **bzip2** - compress or decompress files using bzip2 format

The **bzip2** command is a utility in Unix-like operating systems used to compress files. The basic syntax of the **bzip2** command is:

```
bzip2 [options] [file]
```

Example:

1. To compress a file named **myfile.txt** and create a new file named **myfile.txt.bz2**, use the following command:

```
bzip2 myfile.txt
```

2. To compress a file named **myfile.txt** and keep the original file unchanged, use the following command:

```
bzip2 -c myfile.txt > myfile.txt.bz2
```

- **zip** - create, list, and extract zip archives

The **zip** command is a commonly used command-line utility in Linux. It allows you to create compressed archive files of one or more files and directories.

Here is the syntax and an example of how to use the **zip** command:

```
zip [options] archive_name.zip file1 file2 ... fileN
```

Example:

1. To create a compressed archive file named **myfiles.zip** containing all the files and directories in the current directory, you can use the following command:

```
zip myfiles.zip *
```

This will create a compressed archive of all the files and directories in the current directory, and save it as **myfiles.zip**.

2. To extract the files from a compressed archive, you can use the **unzip** command:

```
unzip myfiles.zip
```

UNIT-8

Linux Commands

Assignment Questions: (Short Answer Questions (SAQs))

- 1) What are the different types of operating systems, and how Linux differ from one another?
- 2) What is the importance of Linux commands in the Linux operating system?
- 3) What is the basic structure of a Linux command, and how is it used?
- 4) What is the different between an argument and an option?
- 5) How can you display the contents of a file in Linux?
- 6) How do you create a new empty file in Linux?
- 7) What is the command to rename a file in Linux?
- 8) How do you copy a file in Linux?
- 9) How can you move a file to a different location in Linux?
- 10) How do you delete a file in Linux?
- 11) How can you create a new directory in Linux?
- 12) What command can you use to list all the files and directories in the current working directory in Linux?
- 13) What is the difference between the cp command and the mv command in Linux?
- 14) What is a symbolic link in Linux, and how is it created?
- 15) How can you view all running processes in Linux?
- 16) How can you terminate a running process in Linux?
- 17) How can you check the CPU and memory usage of a process in Linux?
- 18) What are the three types of permissions that can be set on a file in Linux, and what do they control?
- 19) How can you view the permissions of a file in Linux?
- 20) How can you change the permissions of a file in Linux using chmod?
- 21) How can you add or remove a specific permission for a file in Linux using chmod?
- 22) How can you change the ownership of a file in Linux using chown?
- 23) What is the difference between the chmod and chown commands in Linux?
- 24) What is the find command in Linux, and what is it used for?
- 25) How can you search for a file by its name using the find command in Linux?
- 26) What is the wc command in Linux, and what is it used for?
- 27) How can you count the total number of lines, words, and characters in a file using the wc command in Linux?

- 28) What is a compression command in Linux, and what is it used for?
- 29) How can you create a tar file in Linux using the tar command?
- 30) How can you extract a tar file in Linux using the tar command?
- 31) How can you compress a tar file using the gzip command in Linux?
- 32) How can you compress a tar file using the bzip2 command in Linux?

Multiple Choice Question (MCQs):

1. Which command is used to display the contents of a file?

a) ls b) mv c) cp d) cat

Answer: d) cat

2. Which command is used to create a new directory?

a) rm b) mkdir c) touch d) pwd

Answer: b) mkdir

3. Which command is used to copy a file in Linux?

a) mv b) cp c) rm d) cat

Answer: b) cp

4. Which command is used to delete a file in Linux?

a) cp b) rm c) mv d) cat

Answer: b) rm

5. Which command is used to rename a file in Linux?

a) cp b) mv c) rm d) cat

Answer: b) mv

7. Which command is used to display the current directory?

a) pwd b) cd c) ls d) touch

Answer: a) pwd

8. Which command is used to display the permissions of a file?

a) chmod b) chown c) chgrp d) ls -l

Answer: d) ls -l

9. Which command is used to change the permissions of a file?

a) chmod b) chown c) chgrp d) ls -l

Answer: a) chmod

10. Which command is used to search for a file in Linux?

a) locate b) find c) search d) grep

Answer: b) find

11) What command is used to display the contents of a file in Linux?

a. ls b. cat c. find d. cp

Answer: b. cat

12) What command is used to copy a file in Linux?

a. copy b. mv c. cp d. duplicate

Answer: c. cp

13) What command is used to move a file in Linux?

a. mv b. move c. cp d. shift

Answer: a. mv

14) What command is used to rename a file in Linux?

a. rename b. mv c. copy d. rn

Answer: b. mv

15) What command is used to delete a file in Linux?

a. remove b. erase c. rm d. delete

Answer: c. rm

16) What command is used to change the permissions of a file in Linux?

a. chmod b. chperm c. perm d. chown

Answer: a. chmod

17) What is the syntax for the copy command in Linux?

a. cp file1 file2 b. copy file1 file2 c. move file1 file2 d. cp -r file1 file2

Answer: a. cp file1 file2

18) What is a symbolic link in Linux?

a. A shortcut to a file or directory b. A compressed file
c. A hidden file d. A temporary file

Answer: a. A shortcut to a file or directory

19) Which command is used to create a symbolic link in Linux?

a. link b. ln c. slink d. shortcut

Answer: b. ln

20) What is the syntax for creating a symbolic link in Linux?

- a. ln -s /path/to/source /path/to/link
- b. ln -p /path/to/source /path/to/link
- c. ln -r /path/to/source /path/to/link
- d. ln -t /path/to/source /path/to/link

Answer: a. ln -s /path/to/source /path/to/link

21) Which command is used to display the current running processes in Linux?

- a. ps
- b. top
- c. pstree
- d. kill

Answer: a. ps

22) What is the syntax for the ps command in Linux?

- a. ps -l
- b. ps -a
- c. ps -u
- d. All of the above

Answer: d. All of the above

23) Which command is used to kill a running process in Linux?

- a. stop
- b. kill
- c. end
- d. terminate

Answer: b. kill

24) What is the syntax for killing a process using the kill command in Linux?

- a. kill process_name
- b. kill -p process_id
- c. kill -9 process_id
- d. kill -u username

Answer: c. kill -9 process_id

25) Which command is used to display the system load average and the running processes in real-time in Linux?

- a. ps
- b. top
- c. pstree
- d. killall

Answer: b. top

26) Which command is used to display the file permissions in Linux?

- a. chmod
- b. chown
- c. chgrp
- d. ls

Answer: d. ls

27) What is the syntax for displaying the file permissions using the ls command in Linux?

- a. ls -l
- b. ls -p
- c. ls -a
- d. ls -h

Answer: a. ls -l

28) Which character represents a directory in the output of the ls command in Linux?

- a. d
- b. f
- c. l
- d. -

Answer: a. d

29) Which command is used to change the file permissions in Linux?

- a. chmod
- b. chown
- c. chgrp
- d. ls

Answer: a. chmod

30) What is the syntax for changing the file permissions using the chmod command in Linux?

- a. chmod permissions filename
- b. chmod filename permissions
- c. chmod user permissions filename
- d. chmod group permissions filename

Answer: a. chmod permissions filename

31) Which symbol is used to represent the read permission for a file in Linux?

- a. r
- b. w
- c. x
- d. -

Answer: a. r

32) Which symbol is used to represent the execute permission for a file in Linux?

- a. r
- b. w
- c. x
- d. -

Answer: c. x

33) Which command is used to change the ownership of a file in Linux?

- a. chown
- b. chmod
- c. chgrp
- d. ls

Answer: a. chown

34) Which command is used to count the number of lines, words, and characters in a file in Linux?

- a. wc
- b. lc
- c. wc-lc
- d. lw

Answer: a. wc

35) What is the syntax for using the wc command to count the number of lines, words, and characters in a file in Linux?

- a. wc -lwc filename
- b. wc -lw filename
- c. wc -cwl filename
- d. wc filename

Answer: b. wc -lw filename

36) Which compression command can compress and archive multiple files/folders at once?

- a) tar
- b) gzip
- c) bzip2
- d) zip

Answer: a) tar

37) Which compression command can compress files/folders but not archive them?

- a) tar
- b) gzip
- c) bzip2
- d) zip

Answer: b) gzip

38) Which compression command uses the .tar.gz file extension?

- a) tar
- b) gzip
- c) bzip2
- d) zip

Answer: a) tar

39) Which compression command is known for its high compression ratio?

- a) tar
- b) gzip
- c) bzip2
- d) zip

Answer: c) bzip2

40) Which compression command is commonly used in Windows operating systems?

- a) tar b) gzip c) bzip2 d) zip

Answer: d) zip

41) Which compression command uses the .tar.bz2 file extension?

- a) tar b) gzip c) bzip2 d) zip

Answer: c) bzip2

42) Which compression command is commonly used for compressing large files?

- a) tar b) gzip c) bzip2 d) zip

Answer: d) zip

43) Which compression command can be used to extract files from a compressed archive?

- a) tar b) gzip c) bzip2 d) unzip

Answer: d) unzip

