Basic Linux commands

Listing commands

ls option\_flag arguments --> list the sub directories and files available in the present directory

**Examples:**

ls -l--> list the files and directories in long list format with extra information

ls -a --> list all including hidden files and directory

ls \*.sh --> list all the files having .sh extension.

ls -i --> list the files and directories with index numbers inodes

ls -d \*/ --> list only directories.(we can also specify a pattern)

**Directory commands**

pwd --> print work directory. Gives the present working directory.

cd path\_to\_directory --> change directory to the provided path

cd ~ or just cd --> change directory to the home directory

cd - --> Go to the last working directory.

cd .. --> change directory to one step back.

cd ../.. --> Change directory to 2 levels back.

mkdir directoryName --> to make a directory in a specific location

**Examples:**

mkdir newFolder # make a new folder 'newFolder'

mkdir .NewFolder # make a hidden directory (also . before a file to make it hidden)

mkdir A B C D #make multiple directories at the same time

mkdir /home/user/Mydirectory # make a new folder in a specific location

mkdir -p A/B/C/D # make a nested directory

**1. File and Directory Operations Commands**

File and directory operations are fundamental in working with the Linux operating system. Here are some commonly used File and Directory Operations commands:

| **Command** | **Description** | **Options** | **Examples** |
| --- | --- | --- | --- |
| [ls](https://www.geeksforgeeks.org/ls-command-in-linux/) | List files and directories. | * **-l**: Long format listing. * **-a**: Include hidden files hidden ones * **-h**: Human-readable file sizes. | * **ls -l** displays files and directories with detailed information. * **ls -a** shows all files and directories, including * **ls -lh**  displays file sizes in a human-readable format. |
| [cd](https://www.geeksforgeeks.org/cd-command-in-linux-with-examples/) | Change directory. |  | * **cd /path/to/directory** changes the current directory to the specified path. |
| [pwd](https://www.geeksforgeeks.org/pwd-command-in-linux-with-examples/) | Print current working directory. |  | * **pwd** displays the current working directory. |
| [mkdir](https://www.geeksforgeeks.org/mkdir-command-in-linux-with-examples/) | Create a new directory. |  | * **mkdir my\_directory** creates a new directory named “my\_directory”. |
| [rm](https://www.geeksforgeeks.org/rm-command-linux-examples/) | Remove files and directories. | * **-r**: Remove directories recursively. * **-f**: Force removal without confirmation. | * **rm file.txt** deletes the file named “file.txt”. * **rm -r my\_directory** deletes the directory “my\_directory” and its contents. * **rm -f file.txt** forcefully deletes the file “file.txt” without confirmation. |
| [cp](https://www.geeksforgeeks.org/cp-command-linux-examples/) | Copy files and directories. | * **-r**: Copy directories recursively. | * **cp -r directory destination** copies the directory “directory” and its contents to the specified destination. * **cp file.txt destination** copies the file “file.txt” to the specified destination. |
| [**mv**](https://www.geeksforgeeks.org/mv-command-linux-examples/) | Move/rename files and directories. |  | * **mv file.txt new\_name.txt**  renames the file “file.txt” to “new\_name.txt”. * **mv file.txt directory** moves the file “file.txt” to the specified directory. |
| [touch](https://www.geeksforgeeks.org/touch-command-in-linux-with-examples/) | Create an empty file or update file timestamps. |  | * **touch file.txt**  creates an empty file named “file.txt”. |
| [cat](https://www.geeksforgeeks.org/cat-command-in-linux-with-examples/) | View the contents of a file. |  | * **cat file.txt**  displays the contents of the file “file.txt”. |
| [head](https://www.geeksforgeeks.org/head-command-linux-examples/) | Display the first few lines of a file. | * **-n**: Specify the number of lines to display. | * **head file.txt**  shows the first 10 lines of the file “file.txt”. * **head -n 5 file.txt**  displays the first 5 lines of the file “file.txt”. |
| [tail](https://www.geeksforgeeks.org/tail-command-linux-examples/) | Display the last few lines of a file. | * **-n**: Specify the number of lines to display. | * **tail file.txt**  shows the last 10 lines of the file “file.txt”. * **tail -n 5 file.txt** displays the last 5 lines of the file “file.txt”. |
| [ln](https://www.geeksforgeeks.org/ln-command-in-linux-with-examples/) | Create links between files. | * **-s**: Create symbolic (soft) links. | * **ln -s source\_file link\_name**  creates a symbolic link named “link\_name” pointing to “source\_file”. |
| [find](https://www.geeksforgeeks.org/find-command-in-linux-with-examples/) | Search for files and directories. | * **-name**: Search by filename. * **-type**: Search by file type. | * **find /path/to/search -name “\*.txt”**  searches for all files with the extension “.txt” in the specified directory. |

**2. File Permission Commands**

File permissions on Linux and Unix systems control access to files and directories. There are three basic permissions: read, write, and execute. Each permission can be granted or denied to three different categories of users: the owner of the file, the members of the file’s group, and everyone else.

Here are some file permission commands:

| **Command** | **Description** | **Options** | **Examples** |
| --- | --- | --- | --- |
| [chmod](https://www.geeksforgeeks.org/chmod-command-linux/) | Change file permissions. | * **u**: User/owner permissions. * **g**: Group permissions. * **o**: Other permissions. * **+**: Add permissions. * **–**: Remove permissions. * **=**: Set permissions explicitly. | * **chmod u+rwx file.txt**  grants read, write, and execute permissions to the owner of the file. |
| [chown](https://www.geeksforgeeks.org/chown-command-in-linux-with-examples/) | Change file ownership. |  | * **chown user file.txt** changes the owner of “file.txt” to the specified user. |
| [chgrp](https://www.geeksforgeeks.org/chgrp-command-in-linux-with-examples/) | Change group ownership. |  | * **chgrp group file.txt** changes the group ownership of “file.txt” to the specified group. |
| [umask](https://www.geeksforgeeks.org/umask-command-in-linux-with-examples/) | Set default file permissions. |  | * **umask 022**  sets the default file permissions to read and write for the owner, and read-only for group and others. |

## 3. File Compression and Archiving Commands

Here are some file compression and archiving commands in Linux:

| **Commands** | **Description** | **Options** | **Examples** |
| --- | --- | --- | --- |
| [tar](https://www.geeksforgeeks.org/tar-command-linux-examples/) | Create or extract archive files. | * **-c**: Create a new archive. * **-x**: Extract files from an archive. * **-f**: Specify the archive file name. * **-v**: Verbose mode. * **-z**: Compress the archive with gzip. * **-j**: Compress the archive with bzip2. | * **tar -czvf archive.tar.gz files/**  creates a compressed tar archive named “archive.tar.gz” containing the files in the “files/” directory. |
| [gzip](https://www.geeksforgeeks.org/gzip-command-linux/) | Compress files. | * **-d**: Decompress files. | * **gzip file.txt** compresses the file “file.txt” and renames it as “file.txt.gz”. |
| [zip](https://www.geeksforgeeks.org/zip-command-in-linux-with-examples/) | Create compressed zip archives. | * **-r**: Recursively include directories. | * **zip archive.zip file1.txt file2.txt**  creates a zip archive named “archive.zip” containing “file1.txt” and “file2.txt”. |

## 4. Process Management Commands

In Linux, process management commands allow you to monitor and control running processes on the system. Here are some commonly used process management commands:

| **Commands** | **Description** | **Options** | **Examples** |
| --- | --- | --- | --- |
| [ps](https://www.geeksforgeeks.org/ps-command-in-linux-with-examples/) | Display running processes. | * **-aux**: Show all processes. | * **ps aux**  shows all running processes with detailed information. |
| [top](https://www.geeksforgeeks.org/top-command-in-linux-with-examples/) | Monitor system processes in real-time. |  | * **top** displays a dynamic view of system processes and their resource usage. |
| [kill](https://www.geeksforgeeks.org/kill-command-in-linux-with-examples/) | Terminate a process. | * **-9**: Forcefully kill a process. | * **kill PID**  terminates the process with the specified process ID. |
| [pkill](https://www.geeksforgeeks.org/kill-command-in-linux-with-examples/) | Terminate processes based on their name. |  | * **pkill process\_name**  terminates all processes with the specified name. |
| **pgrep** | List processes based on their name. |  | * **pgrep process\_name**  lists all processes with the specified name. |
| [grep](https://www.geeksforgeeks.org/grep-command-in-unixlinux/) | used to search for specific patterns or regular expressions in text files or streams and display matching lines. | * **-i**: Ignore case distinctions while searching. * **-v**: Invert the match, displaying non-matching lines. * **-r or -R**: Recursively search directories for matching patterns. * **-l**: Print only the names of files containing matches. * **-n**: Display line numbers alongside matching lines. * **-w**: Match whole words only, rather than partial matches. * **-c**: Count the number of matching lines instead of displaying them. * **-e**: Specify multiple patterns to search for. * **-A**: Display lines after the matching line. * **-B**: Display lines before the matching line. * **-C**: Display lines both before and after the matching line. | * **grep -i “hello” file.txt** * **grep -v “error” file.txt** * **grep -r “pattern” directory/** * **grep -l “keyword” file.txt** * **grep -n “pattern” file.txt** In these examples we are extracting our desirec output from filename (file.txt) |

## 5. System Information Commands

In Linux, there are several commands available to gather system information. Here are some commonly used system information commands:

| **sudCommand** | **Description** | **Options** | **Examples** |
| --- | --- | --- | --- |
| [uname](https://www.geeksforgeeks.org/uname-command-in-linux-with-examples/) | Print system information. | * **-a**: All system information. | * **uname -a** displays all system information. |
| [whoami](https://www.geeksforgeeks.org/whoami-command-linux-example/) | Display current username. |  | * **whoami**  shows the current username. |
| [df](https://www.geeksforgeeks.org/df-command-in-linux-with-examples/) | Show disk space usage. | * **-h**: Human-readable sizes. | * **df -h**  displays disk space usage in a human-readable format. |
| [du](https://www.geeksforgeeks.org/du-command-linux-examples/) | Estimate file and directory sizes. | * **-h**: Human-readable sizes. * **-s**: Display total size only. | * **du -sh directory/**  provides the total size of the specified directory. |
| [free](https://www.geeksforgeeks.org/free-command-linux-examples/) | Display memory usage information. | * **-h**: Human-readable sizes. | * **free -h**  displays memory usage in a human-readable format. |
| [uptime](https://www.geeksforgeeks.org/linux-uptime-command-with-examples/) | Show system uptime. |  | * **uptime** shows the current system uptime. |
| **lscpu** | Display CPU information. |  | * **lscpu**  provides detailed CPU information. |
| **lspci** | List PCI devices. |  | * **lspci** List PCI devices. |
| [lsusb](https://www.geeksforgeeks.org/lsusb-command-in-linux-with-examples/) | List USB devices. |  | * **lsusb**  lists all connected USB devices. |

## 6. Networking Commands

In Linux, there are several networking commands available to manage and troubleshoot network connections. Here are some commonly used networking commands:

| **Command** | **Description** | **Examples** |
| --- | --- | --- |
| [ifconfig](https://www.geeksforgeeks.org/ifconfig-command-in-linux-with-examples/) | Display network interface information. | * **ifconfig**  shows the details of all network interfaces. |
| [ping](https://www.geeksforgeeks.org/ping-command-in-linux-with-examples/) | Send ICMP echo requests to a host. | * **ping google.com**  sends ICMP echo requests to “google.com” to check connectivity. |
| [netstat](https://www.geeksforgeeks.org/netstat-command-linux/) | Display network connections and statistics. | * **netstat -tuln**  shows all listening TCP and UDP connections. |
| **ss** | Display network socket information. | * **ss -tuln**  shows all listening TCP and UDP connections. |
| [ssh](https://www.geeksforgeeks.org/ssh-command-in-linux-with-examples/) | Securely connect to a remote server. | * **ssh user@hostname**  initiates an SSH connection to the specified hostname. |
| [scp](https://www.geeksforgeeks.org/scp-command-in-linux-with-examples/) | Securely copy files between hosts. | * **scp file.txt user@hostname:/path/to/destination**  securely copies “file.txt” to the specified remote host. |
| [wget](https://www.geeksforgeeks.org/wget-command-in-linux-unix/) | Download files from the web. | * **wget http://example.com/file.txt**  downloads “file.txt” from the specified URL. |
| [curl](https://www.geeksforgeeks.org/curl-command-in-linux-with-examples/) | Transfer data to or from a server. | * **curl http://example.com**  retrieves the content of a webpage from the specified URL. |

## 7. IO Redirection Commands

In Linux, IO (Input/Output) redirection commands are used to redirect the standard input, output, and error streams of commands and processes. Here are some commonly used IO redirection commands:

| **Command** | **Description** |
| --- | --- |
| cmd < file | Input of cmd is taken from file. |
| cmd > file | Standard output (stdout) of cmd is redirected to file. |
| cmd 2> file | Error output (stderr) of cmd is redirected to file. |
| cmd 2>&1 | stderr is redirected to the same place as stdout. |
| cmd1 <(cmd2) | Output of cmd2 is used as the input file for cmd1. |
| cmd > /dev/null | Discards the stdout of cmd by sending it to the null device. |
| cmd &> file | Every output of cmd is redirected to file. |
| cmd 1>&2 | stdout is redirected to the same place as stderr. |
| cmd >> file | Appends the stdout of cmd to file. |

## 8. Environment Variable Commands

In Linux, environment variables are used to store configuration settings, system information, and other variables that can be accessed by processes and shell scripts. Here are some commonly used environment variable commands:

| **Command** | **Description** |
| --- | --- |
| **export VARIABLE\_NAME=value** | Sets the value of an environment variable. |
| **echo $VARIABLE\_NAME** | Displays the value of a specific environment variable. |
| **env** | Lists all environment variables currently set in the system. |
| **unset VARIABLE\_NAME** | Unsets or removes an environment variable. |
| **export -p** | Shows a list of all currently exported environment variables. |
| **env VAR1=value COMMAND** | Sets the value of an environment variable for a specific command. |
| **printenv** | Displays the values of all environment variables. |

## 9. User Management Commands

In Linux, user management commands allow you to create, modify, and manage user accounts on the system. Here are some commonly used user management commands:

| **Command** | **Description** |
| --- | --- |
| **who** | Show who is currently logged in. |
| **sudo adduser username** | Create a new user account on the system with the specified username. |
| **finger** | Display information about all the users currently logged into the system, including their usernames, login time, and terminal. |
| **sudo deluser USER GROUPNAME** | Remove the specified user from the specified group. |
| **last** | Show the recent login history of users. |
| **finger username** | Provide information about the specified user, including their username, real name, terminal, idle time, and login time. |
| **sudo userdel -r username** | Delete the specified user account from the system, including their home directory and associated files. The -r option ensures the removal of the user’s files. |
| **sudo passwd -l username** | Lock the password of the specified user account, preventing the user from logging in. |
| **su – username** | Switch to another user account with the user’s environment. |
| **sudo usermod -a -G GROUPNAME USERNAME** | Add an existing user to the specified group. The user is added to the group without removing them from their current groups. |

## 10. Shortcuts Commands

There are many shortcuts commands in Linux that can help you be more productive. Here are a few of the most common ones:

### 10.1: Bash Shortcuts Commands:

| **Navigation** | **Description** | **Editing** |  | **Description** | **History** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| **Ctrl + A** | Move to the beginning of the line. | **Ctrl + U** |  | Cut/delete from the cursor position to the beginning of the line. | **Ctrl + R** | Search command history (reverse search). |
| **Ctrl + E** | Move to the end of the line. | **Ctrl + K** |  | Cut/delete from the cursor position to the end of the line. | **Ctrl + G** | Escape from history search mode. |
| **Ctrl + B** | Move back one character. | **Ctrl + W** |  | Cut/delete the word before the cursor. | **Ctrl + P** | Go to the previous command in history. |
| **Ctrl + F** | Move forward one character. | **Ctrl + Y** |  | Paste the last cut text. | **Ctrl + N** | Go to the next command in history. |
| **Alt + B** | Move back one word | **Ctrl + L** |  | Clear the screen. | **Ctrl + C** | Terminate the current command. |
| **Alt + F** | Move forward one word. |  |  |  |  |  |

# Linux Directory Structure

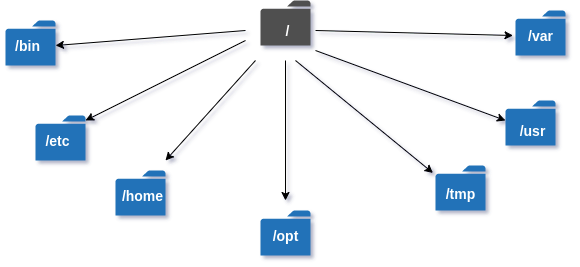
**Prerequisite:**[**Linux File Hierarchy Structure**](https://www.geeksforgeeks.org/linux-file-hierarchy-structure/)

In Linux/Unix operating system everything is a file even directories are files, files are files, and devices like mouse, keyboard, printer, etc are also files. Here we are going to see the Directory Structure in Linux.

## ****Types**** of files in the Linux system.

1. **General Files** – It is also called ordinary files. It may be an image, video, program, or simple text file. These types of files can be in ASCII or Binary format. It is the most commonly used file in the Linux system.
2. **Directory Files**– These types of files are a warehouse for other file types. It may be a directory file within a directory (subdirectory).
3. **Device Files –** In a Windows-like operating system, devices like CD-ROM, and hard drives are represented as drive letters like F: G: H whereas in the Linux system devices are represented as files. As for example, /dev/sda1, /dev/sda2, and so on.

We know that in a Windows-like operating system, files are stored in different folders on different data drives like C: D: E: whereas in the Linux/Unix operating system files are stored in a tree-like structure starting with the root directory as shown in the below diagram.



*data storage in Linux/Unix operating systems*

The Linux/Unix file system hierarchy base begins at the root and everything starts with the root directory.

### **These are the common top-level directories associated with the root directory:**

| **Directories** | **Description** |
| --- | --- |
| **/bin** | binary or executable programs. |
| **/etc** | system configuration files. |
| **/home** | home directory. It is the default current directory. |
| **/opt** | optional or third-party software. |
| **/tmp** | temporary space, typically cleared on reboot. |
| **/usr** | User related programs. |
| **/var** | log files. |

### **Some other directories in the Linux system:**

| **Directories** | **Description** |
| --- | --- |
| **/boot** | It contains all the boot-related information files and folders such as conf, grub, etc. |
| **/dev** | It is the location of the device files such as dev/sda1, dev/sda2, etc. |
| **/lib** | It contains kernel modules and a shared library. |
| **/lost+found** | It is used to find recovered bits of corrupted files. |
| **/media** | It contains subdirectories where removal media devices are inserted. |
| **/mnt** | It contains temporary mount directories for mounting the file system. |
| **/proc** | It is a virtual and pseudo-file system to contains info about the running processes with a specific process ID or PID. |
| **/run** | It stores volatile runtime data. |
| **/sbin** | binary executable programs for an administrator. |
| **/srv** | It contains server-specific and server-related files. |
| **/sys** | It is a virtual file system for modern Linux distributions to store and allows modification of the devices connected to the system. |

### **Exploring directories and their usability**:

We know that Linux is a very complex system that requires an efficient way to start, stop, maintain and reboot a system, unlike Windows operating system. In the Linux system some well-defined configuration files, binaries, main pages information files are available for every process.

#### **Linux Kernel File:**

* **/boot/vmlinux –**The Linux kernel file.

#### **Device Files:**

* **/dev/hda –** Device file for the first IDE HDD.
* **/dev/hdc –**A pseudo-device that output garbage output is redirected to /dev/null.

#### **System Configuration Files:**

| **Configuration Files** | **Description** |
| --- | --- |
| **/etc/bashrc** | It is used by bash shell that contains system defaults and aliases. |
| **/etc/crontab** | A shell script to run specified commands on a predefined time interval. |
| **/etc/exports** | It contains information on the file system available on the network. |
| **/etc/fstab** | Information of the Disk Drive and their mount point. |
| **/etc/group** | It is a text file to define Information of Security Group. |
| **/etc/grub.conf** | It is the grub bootloader configuration file. |
| **/etc/init.d** | Service startup Script. |
| **/etc/lilo.conf** | It contains lilo bootloader configuration file. |
| **/etc/hosts** | Information of IP and corresponding hostnames |
| **/etc/hosts.allow** | It contains a list of hosts allowed accessing services on the local machine. |
| **/etc/host.deny** | List of hosts denied accessing services on the local machine. |
| **/etc/inittab** | INIT process and their interaction at the various run levels. |
| **/etc/issue** | Allows editing the pre-login message. |
| **/etc/modules.conf** | It contains the configuration files for the system modules. |
| **/etc/motd** | It contains the message of the day. |
| **/etc/mtab** | Currently mounted blocks information. |
| **/etc/passwd** | It contains username, password of the system, users in a shadow file. |
| **/etc/printcap** | It contains printer Information. |
| **/etc/profile** | Bash shell defaults. |
| **/etc/profile.d** | It contains other scripts like application scripts, executed after login. |
| **/etc/rc.d** | It avoids script duplication. |
| **/etc/rc.d/init.d** | Run Level Initialisation Script. |
| **/etc/resolv.conf** | DNS being used by System. |
| **/etc/security** | It contains the name of terminals where root login is possible. |
| **/etc/skel** | Script that initiates new user home directory. |
| **/etc/termcap** | An ASCII file that defines the behavior of different types of the terminal. |
| **/etc/X11** | Directory tree contains all the conf files for the X-window System. |

#### **User Related Files:**

| **User Related Files** | **Descriptions** |
| --- | --- |
| **/usr/bin** | It contains most of the executable files. |
| **/usr/bin/X11** | Symbolic link of /usr/bin. |
| **/usr/include** | It contains standard files used by C program. |
| **/usr/share** | It contains architecture independent shareable text files. |
| **/usr/lib** | It contains object files and libraries. |
| **/usr/sbin** | It contains commands for Super User, for System Administration. |

#### Virtual and Pseudo Process Related Files:

| **Virtual and Pseudo Process Related Files** | **Descriptions** |
| --- | --- |
| **/proc/cpuinfo** | CPU Information |
| **/proc/filesystems** | It keeps useful info about the processes that are currently running. |
| **/proc/interrupts** | it keeps the information about the number of interrupts per IRQ. |
| **/proc/ioports** | Contains all the Input and Output addresses used by devices on the server |
| **/proc/meminfo** | It reports the memory usage information. |
| **/proc/modules** | Currently using kernel module. |
| **/proc/mount** | Mounted File-system Information. |
| **/proc/stat** | It displays the detailed statistics of the current system. |
| **/proc/swaps** | It contains swap file information. |

#### Version Information File:

* **/version –** It displays the Linux version information.

#### Log Files:

| **Log Files** | **Descriptions** |
| --- | --- |
| **/var/log/lastlog** | It stores user’s last login info. |
| **/var/log/messages** | It has all the global system messages |
| **/var/log/wtmp** | It keeps a history of login and logout information. |