

Linux is one of the most popular operating systems with an extensive user-base around the world, most commonly programmers and developers. The open-source Unix-like operating system (OS) is rooted from the Linux kernel - an operating system developed by Linus Torvalds.

In simple words, Linux is a Unix clone, and its open-source feature allows anyone to change anything in Linux and redistribute it with your branding. These iterations of Linux are referred to as 'distros.'

Below are some of the most common Linux distributions such as:

- Debian
- Fedora
- Mandriva Linux
- openSUSE
- Arch Linux
- Gentoo
- Slackware
- Ubuntu

Its global use proves the fact that Linux is the most used operating system. Over 80% of the internet runs on Linux servers. More so, about 70-80% of smartphones operate on the Linux operating system.

According to the latest statistics, Android has a market share of 76% in the mobile OS. [Android](#) is developed from the Linux kernel, and the reason for its wide acceptance is that it is free and open-source. Apart from that, the Linux operating system is safe as Windows is more prone to viruses.

1. The Linux Directory Structure,
 - a. If you're coming from Windows, the Linux file system structure can seem particularly alien. The C:\ drive and drive letters are gone, replaced by a / and cryptic-sounding directories, most of which have three letter names.
 - b. The Filesystem Hierarchy Standard (FHS) defines the structure of file systems on Linux and other UNIX-like operating systems. However, Linux file systems also contain some directories that aren't yet defined by the standard.
2. / — The Root Directory
 - a. Everything on your Linux system is located under the / directory, known as the root directory. You can think of the / directory as being similar to the C:\ directory on Windows — but this isn't strictly true, as Linux doesn't have drive letters. While another partition would be located at D:\ on Windows, this other partition would appear in another folder under / on Linux.

```

ubuntu@ubuntu:~/Desktop$ ls -l /
total 2
lrwxrwxrwx   1 root root    7 Feb 23 03:57 bin -> usr/bin
drwxr-xr-x   3 root root 245 Feb 23 04:02 boot
dr-xr-xr-x   1 root root 2048 Feb 23 04:13 cdrom
drwxr-xr-x  18 root root 4040 Apr  7 07:04 dev
drwxr-xr-x   1 root root  560 Apr  7 07:03 etc
drwxr-xr-x   1 root root   60 Apr  7 07:03 home
lrwxrwxrwx   1 root root    7 Feb 23 03:57 lib -> usr/lib
lrwxrwxrwx   1 root root    9 Feb 23 03:57 lib32 -> usr/lib32
lrwxrwxrwx   1 root root    9 Feb 23 03:57 lib64 -> usr/lib64
lrwxrwxrwx   1 root root   10 Feb 23 03:57 libx32 -> usr/libx32
drwxr-xr-x   1 root root   60 Apr  7 07:03 media
drwxr-xr-x   2 root root    3 Feb 23 03:57 mnt
drwxr-xr-x   2 root root    3 Feb 23 03:57 opt
dr-xr-xr-x 247 root root    0 Apr  7 07:03 proc
drwxr-xr-x  18 root root  308 Feb 23 04:02 rofs
drwx-----   1 root root  100 Apr  7 07:08 root
drwxr-xr-x  34 root root 1020 Apr  7 07:12 run
lrwxrwxrwx   1 root root    8 Feb 23 03:57/sbin -> usr/sbin
drwxr-xr-x  11 root root  193 Feb 23 04:03 snap
drwxr-xr-x   2 root root    3 Feb 23 03:57 srv
dr-xr-xr-x  13 root root    0 Apr  7 07:03 sys
drwxrwxrwt  18 root root  380 Apr  7 10:02 tmp
drwxr-xr-x   1 root root  100 Feb 23 03:57 usr
drwxr-xr-x   1 root root  180 Feb 23 04:02 var

```

- b. /bin — Essential User Binaries: The /bin directory contains the essential user binaries (programs) that must be present when the system is mounted in single-user mode. Applications such as Firefox are stored in /usr/bin, while important system programs and utilities such as the bash shell are located in /bin. The /usr directory may be stored on another partition — placing these files in the /bin directory ensures the system will have these important utilities even if no other file systems are mounted. The /sbin directory is similar — it contains essential system administration binaries.

```

ubuntu@ubuntu:~/Desktop$ ls -l /bin
lrwxrwxrwx 1 root root 7 Feb 23 03:57 /bin -> usr/bin
ubuntu@ubuntu:~/Desktop$

```

- c. /boot — Static Boot Files: The /boot directory contains the files needed to boot the system — for example, the GRUB boot loader's files and your Linux kernels are stored here. The boot loader's configuration files aren't located here, though — they're in /etc with the other configuration files.
- d. /cdrom — Historical Mount Point for CD-ROMs: The /cdrom directory isn't part of the FHS standard, but you'll still find it on Ubuntu and other operating systems. It's a

temporary location for CD-ROMs inserted in the system. However, the standard location for temporary media is inside the `/media` directory.

- e. `/dev` — Device Files: Linux exposes devices as files, and the `/dev` directory contains a number of special files that represent devices. These are not actual files as we know them, but they appear as files — for example, `/dev/sda` represents the first SATA drive in the system. If you wanted to partition it, you could start a partition editor and tell it to edit `/dev/sda`.

This directory also contains pseudo-devices, which are virtual devices that don't actually correspond to hardware. For example, `/dev/random` produces random numbers. `/dev/null` is a special device that produces no output and automatically discards all input — when you pipe the output of a command to `/dev/null`, you discard it.

```
total 0
crw-r--r--  1 root  root    10, 235 Apr  7 07:04 autofs
drwxr-xr-x  2 root  root   340 Apr  7 07:04 block
drwxr-xr-x  2 root  root    80 Apr  7 07:03 bsg
crw-----  1 root  root   10, 234 Apr  7 07:03 btrfs-control
drwxr-xr-x  3 root  root    60 Apr  7 07:03 bus
lrwxrwxrwx  1 root  root     3 Apr  7 07:04 cdrom -> sr0
drwxr-xr-x  2 root  root  3640 Apr  7 07:04 char
crw-----  1 root  root     5,  1 Apr  7 07:04 console
lrwxrwxrwx  1 root  root    11 Apr  7 07:03 core -> /proc/kcore
drwxr-xr-x  3 root  root    60 Apr  7 07:03 cpu
crw-----  1 root  root   10, 124 Apr  7 07:04 cpu_dma_latency
crw-----  1 root  root   10, 203 Apr  7 07:03 cuse
drwxr-xr-x  8 root  root   160 Apr  7 07:03 disk
drwxr-xr-x  2 root  root    60 Apr  7 07:03 dma_heap
crw-----  1 root  root   10, 126 Apr  7 07:04 ecryptfs
lrwxrwxrwx  1 root  root    13 Apr  7 07:03 fd -> /proc/self/fd
crw-rw-rw-  1 root  root     1,  7 Apr  7 07:04 full
crw-rw-rw-  1 root  root   10, 229 Apr  7 07:04 fuse
crw-----  1 root  root  239,  0 Apr  7 07:04 hidraw0
crw-----  1 root  root   10, 228 Apr  7 07:04 hpet
drwxr-xr-x  2 root  root     0 Apr  7 07:03 hugepages
crw-----  1 root  root   10, 183 Apr  7 07:04 hwrng
crw-----  1 root  root    89,  0 Apr  7 07:04 i2c-0
lrwxrwxrwx  1 root  root    12 Apr  7 07:03 initctl -> /run/initctl
drwxr-xr-x  4 root  root   340 Apr  7 07:03 input
crw-r--r--  1 root  root     1, 11 Apr  7 07:04 kmsg
lrwxrwxrwx  1 root  root    28 Apr  7 07:03 log -> /run/systemd/journal
```

- f. `/etc` — Configuration Files: The `/etc` directory contains configuration files, which can generally be edited by hand in a text editor. Note that the `/etc/` directory contains system-wide configuration files — user-specific configuration files are located in each user's home directory.
- g. `/home` — Home Folders: The `/home` directory contains a home folder for each user. For example, if your user name is bob, you have a home folder located at `/home/bob`. This home folder contains the user's data files and user-specific configuration files. Each user only has write access to their own home folder and

must obtain elevated permissions (become the root user) to modify other files on the system.

- h. `/lib` — Essential Shared Libraries: The `/lib` directory contains libraries needed by the essential binaries in the `/bin` and `/sbin` folder. Libraries needed by the binaries in the `/usr/bin` folder are located in `/usr/lib`.
- i. `/lost+found` — Recovered Files: Each Linux file system has a `lost+found` directory. If the file system crashes, a file system check will be performed at next boot. Any corrupted files found will be placed in the `lost+found` directory, so you can attempt to recover as much data as possible.
- j. `/media` — Removable Media: The `/media` directory contains subdirectories where removable media devices inserted into the computer are mounted. For example, when you insert a CD into your Linux system, a directory will automatically be created inside the `/media` directory. You can access the contents of the CD inside this directory.
- k. `/mnt` — Temporary Mount Points: Historically speaking, the `/mnt` directory is where system administrators mounted temporary file systems while using them. For example, if you're mounting a Windows partition to perform some file recovery operations, you might mount it at `/mnt/windows`. However, you can mount other file systems anywhere on the system.
- l. `/opt` — Optional Packages: The `/opt` directory contains subdirectories for optional software packages. It's commonly used by proprietary software that doesn't obey the standard file system hierarchy — for example, a proprietary program might dump its files in `/opt/application` when you install it.
- m. `/proc` — Kernel & Process Files: The `/proc` directory similar to the `/dev` directory because it doesn't contain standard files. It contains special files that represent system and process information.
- n. `/root` — Root Home Directory: The `/root` directory is the home directory of the root user. Instead of being located at `/home/root`, it's located at `/root`. This is distinct from `/`, which is the system root directory.

- o. /run — Application State Files: The /run directory is fairly new, and gives applications a standard place to store transient files they require like sockets and process IDs. These files can't be stored in /tmp because files in /tmp may be deleted.
- p. /sbin — System Administration Binaries: The /sbin directory is similar to the /bin directory. It contains essential binaries that are generally intended to be run by the root user for system administration.
- q. /selinux — SELinux Virtual File System: If your Linux distribution uses SELinux for security (Fedora and Red Hat, for example), the /selinux directory contains special files used by SELinux. It's similar to /proc. Ubuntu doesn't use SELinux, so the presence of this folder on Ubuntu appears to be a bug.
- r. /srv — Service Data: The /srv directory contains "data for services provided by the system." If you were using the Apache HTTP server to serve a website, you'd likely store your website's files in a directory inside the /srv directory.
- s. /tmp — Temporary Files: Applications store temporary files in the /tmp directory. These files are generally deleted whenever your system is restarted and may be deleted at any time by utilities such as tmpwatch.
- t. /usr — User Binaries & Read-Only Data: The /usr directory contains applications and files used by users, as opposed to applications and files used by the system. For example, non-essential applications are located inside the /usr/bin directory instead of the /bin directory and non-essential system administration binaries are located in the /usr/sbin directory instead of the /sbin directory. Libraries for each are located inside the /usr/lib directory. The /usr directory also contains other directories — for example, architecture-independent files like graphics are located in /usr/share.
- u. The /usr/local directory is where locally compiled applications install to by default — this prevents them from mucking up the rest of the system.
- v. /var — Variable Data Files: The /var directory is the writable counterpart to the /usr directory, which must be read-only in normal operation. Log files and everything else that would normally be written to /usr during normal operation are written to the /var directory. For example, you'll find log files in /var/log.

3. Start with following commands

```
ubuntu@ip-172-31-32-24:~/Desktop$ sudo su -
root@ip-172-31-32-24:~# pwd
/root
root@ip-172-31-32-24:~# who
ubuntu pts/1 2023-04-05 09:54
root@ip-172-31-32-24:~# whoami
root
root@ip-172-31-32-24:~#
```

4. Now work with Basic commands

Command	Description
Basic Linux commands	
Ls	Lists all files and directories in the present working directory
ls -R	Lists files in sub-directories as well
ls -a	Lists hidden files as well
ls -al	Lists files and directories with detailed information like permissions,size, owner, etc.
cd or cd ~	Navigate to HOME directory
cd ..	Move one level up
cd	To change to a particular directory
cd /	Move to the root directory
cat > filename	Creates a new file
cat filename	Displays the file content
cat file1 file2 > file3	Joins two files (file1, file2) and stores the output in a new file (file3)
mv file "new file path"	Moves the files to the new location
mv filename new_file_name	Renames the file to a new filename
sudo	Allows regular users to run programs with the security privileges of the superuser or root
rm filename	Deletes a file
man	Gives help information on a command
history	Gives a list of all past commands typed in the current terminal session

clear	Clears the terminal
mkdir directoryname	Creates a new directory in the present working directory or a at the specified path
rmdir	Deletes a directory
mv	Renames a directory
File Permission commands	
ls -l	to show file type and access permission
r	read permission
w	write permission
x	execute permission
-=	no permission
Chown user	For changing the ownership of a file/directory
Chown user:group filename	change the user as well as group for a file or directory
Environment Variables command	
echo \$VARIABLE	To display value of a variable
env	Displays all environment variables
VARIABLE_NAME=variable_value	Create a new variable
Unset	Remove a variable
export Variable=value	To set value of an environment variable
User management commands of linux	
sudo adduser username	To add a new user
sudo passwd -l 'username'	To change the password of a user
sudo userdel -r 'username'	To remove a newly created user
sudo usermod -a -G GROUPNAME USERNAME	To add a user to a group
sudo deluser USER GROUPNAME	To remove a user from a group
finger	Shows information of all the users logged in
finger username	Gives information of a particular user
Networking command	
SSH username@ip-address or hostname	login into a remote Linux machine using SSH
Ping hostname="" or =""	To ping and Analyzing network and host connections
dir	Display files in the current directory of a remote computer
cd "dirname"	change directory to "dirname" on a remote computer
put file	upload 'file' from local to remote computer
get file	Download 'file' from remote to local computer
quit	Logout
Process command	

bg	To send a process to the background
fg	To run a stopped process in the foreground
top	Details on all Active Processes
ps	Give the status of processes running for a user
ps PID	Gives the status of a particular process
pidof	Gives the Process ID (PID) of a process
kill PID	Kills a process
nice	Starts a process with a given priority
renice	Changes priority of an already running process
df	Gives free hard disk space on your system
free	Gives free RAM on your system
VI Editing Commands	
i	Insert at cursor (goes into insert mode)
a	Write after cursor (goes into insert mode)
A	Write at the end of line (goes into insert mode)
ESC	Terminate insert mode
u	Undo last change
U	Undo all changes to the entire line
o	Open a new line (goes into insert mode)
dd	Delete line
3dd	Delete 3 lines
D	Delete contents of line after the cursor
C	Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.
dw	Delete word
4dw	Delete 4 words
cw	Change word
x	Delete character at the cursor
r	Replace character
R	Overwrite characters from cursor onward
S	Substitute one character under cursor continue to insert
S	Substitute entire line and begin to insert at the beginning of the line
~	Change case of individual character

5. What is a Pipe in Linux?

The Pipe is a command in Linux that lets you use two or more commands such that output of one command serves as input to the next. In short, the output of each process directly as input to the next one like a pipeline. The symbol '|' denotes a pipe.

Pipes help you mash-up two or more commands at the same time and run them consecutively. You can use powerful commands which can perform complex tasks in a jiffy.

Let us understand this with an example.

When you use 'cat' command to view a file which spans multiple pages, the prompt quickly jumps to the last page of the file, and you do not see the content in the middle.

To avoid this, you can pipe the output of the 'cat' command to 'less' which will show you only one scroll length of content at a time.

```
ubuntu@ubuntu:~/Desktop$ grep 'A' month.txt
April
August
ubuntu@ubuntu:~/Desktop$ grep 'A[p]' month.txt
April
ubuntu@ubuntu:~/Desktop$ grep 'A[a-z]' month.txt
April
August
ubuntu@ubuntu:~/Desktop$ grep 'A[^a-z]' month.txt
ubuntu@ubuntu:~/Desktop$
```

```

ubuntu@ubuntu:~/Desktop$ grep "A[a-z]" month.txt|cut -c3
r
g
ubuntu@ubuntu:~/Desktop$ grep "A[a-z]" month.txt|cut -c3|wc
      2      2      4
ubuntu@ubuntu:~/Desktop$ grep "A[a-z]" month.txt|cut -c3|wc -w
2
ubuntu@ubuntu:~/Desktop$ grep "A[a-z]" month.txt|cut -c3|wc -d
wc: invalid option -- 'd'
Try 'wc --help' for more information.
ubuntu@ubuntu:~/Desktop$ grep "A[a-z]" month.txt|cut -c3|wc -c
4
ubuntu@ubuntu:~/Desktop$ wc
^C
ubuntu@ubuntu:~/Desktop$ man wc
ubuntu@ubuntu:~/Desktop$ grep "A[a-z]" month.txt|cut -c3|wc -m
4
ubuntu@ubuntu:~/Desktop$ ls
month.txt  ubiquity.desktop
ubuntu@ubuntu:~/Desktop$ ls -
ls: cannot access '-': No such file or directory
ubuntu@ubuntu:~/Desktop$ ls -l
total 16
-rw-rw-r-- 1 ubuntu ubuntu  87 Apr  7 07:16 month.txt
-rwxr-xr-x 1 ubuntu ubuntu 8273 Apr  7 07:03 ubiquity.desktop
ubuntu@ubuntu:~/Desktop$ chmod u+r u+w u+x month.txt
chmod: cannot access 'u+w': No such file or directory

```

```
chmod: cannot access 'g-rwx': No such file or directory
ubuntu@ubuntu:~/Desktop$ chmod g-rwx month.txt
ubuntu@ubuntu:~/Desktop$ ls -l
total 16
-rw-r--r-- 1 ubuntu ubuntu 87 Apr 7 07:16 month.txt
-rwxr-xr-x 1 ubuntu ubuntu 8273 Apr 7 07:03 ubiquity.desktop
ubuntu@ubuntu:~/Desktop$ chmod go-rwx month.txt
ubuntu@ubuntu:~/Desktop$ ls -l
total 16
-rwx----- 1 ubuntu ubuntu 87 Apr 7 07:16 month.txt
-rwxr-xr-x 1 ubuntu ubuntu 8273 Apr 7 07:03 ubiquity.desktop
ubuntu@ubuntu:~/Desktop$ chmod ugo+rwx g-rwx month.txt
chmod: cannot access 'g-rwx': No such file or directory
ubuntu@ubuntu:~/Desktop$ chmod ugo+rwx month.txt
ubuntu@ubuntu:~/Desktop$ ls -l
total 16
-rwxrwxrwx 1 ubuntu ubuntu 87 Apr 7 07:16 month.txt
-rwxr-xr-x 1 ubuntu ubuntu 8273 Apr 7 07:03 ubiquity.desktop
ubuntu@ubuntu:~/Desktop$ chmod a-rwx month.txt
ubuntu@ubuntu:~/Desktop$ ls -l
total 16
----- 1 ubuntu ubuntu 87 Apr 7 07:16 month.txt
-rwxr-xr-x 1 ubuntu ubuntu 8273 Apr 7 07:03 ubiquity.desktop
ubuntu@ubuntu:~/Desktop$ chmod a+rwx month.txt
ubuntu@ubuntu:~/Desktop$ ls -l
total 16
-rwxrwxrwx 1 ubuntu ubuntu 87 Apr 7 07:16 month.txt
-rwxr-xr-x 1 ubuntu ubuntu 8273 Apr 7 07:03 ubiquity.desktop
ubuntu@ubuntu:~/Desktop$
```

```

total 16
-rwxrwxrwx 1 ubuntu ubuntu 87 Apr 7 07:16 month.txt
-rwxr-xr-x 1 ubuntu ubuntu 8273 Apr 7 07:03 ubiquity.desktop
ubuntu@ubuntu:~/Desktop$ ls -l|grep "M[a-z]"
ubuntu@ubuntu:~/Desktop$ ls -l|grep "m[a-z]"
-rwxrwxrwx 1 ubuntu ubuntu 87 Apr 7 07:16 month.txt
ubuntu@ubuntu:~/Desktop$ ls -l|grep "m[a-z]"|cut -d" " -f2
1
ubuntu@ubuntu:~/Desktop$ ls -l|grep "m[a-z]"|cut -d" " -f3
ubuntu
ubuntu@ubuntu:~/Desktop$ ls -l|grep "m[a-z]"|cut -d" " -f4
ubuntu
ubuntu@ubuntu:~/Desktop$ ls -l|grep "m[a-z]"|cut -d" " -f4|wc
      1      1      7
ubuntu@ubuntu:~/Desktop$ ls -l|grep "m[a-z]"|cut -d" " -f4|wc -m2
wc: invalid option -- '2'
Try 'wc --help' for more information.
ubuntu@ubuntu:~/Desktop$ ls -l|grep "m[a-z]"|cut -d" " -f4|wc -m
7
ubuntu@ubuntu:~/Desktop$ ls -l|grep "m[a-z]"|cut -d" " -f4|wc -c
7
ubuntu@ubuntu:~/Desktop$ ls -l|grep "m[a-z]"|cut -d" " -f4|wc -c1
wc: invalid option -- '1'
Try 'wc --help' for more information.
ubuntu@ubuntu:~/Desktop$ ls -l|grep "m[a-z]"|cut -d" " -f4|wc
      1      1      7
ubuntu@ubuntu:~/Desktop$

```



```

ubuntu@ubuntu:~/Desktop$ grep '[a-f]'|cut -c3|wc -c month.txt
87 month.txt
^[A
^C
ubuntu@ubuntu:~/Desktop$ grep '[a-f]'|cut -c3|wc month.txt
13 12 87 month.txt
^[A^C
ubuntu@ubuntu:~/Desktop$ grep '[a-f]'|cut -c2|wc month.txt
13 12 87 month.txt
^[A^C
ubuntu@ubuntu:~/Desktop$ grep '[a-f]'|cut -c2|wc month.txt
13 12 87 month.txt
^C
ubuntu@ubuntu:~/Desktop$ grep "A[a-z]"|cut -c2|wc month.txt
13 12 87 month.txt
^C
ubuntu@ubuntu:~/Desktop$ grep "A[a-z]"|wc month.txt
13 12 87 month.txt
^[A^C
ubuntu@ubuntu:~/Desktop$ grep "A[a-z]" month.txt
April
August
ubuntu@ubuntu:~/Desktop$ grep "A[a-z]" month.txt|cut -c3
r
g
ubuntu@ubuntu:~/Desktop$ grep "A[a-z]" month.txt|cut -c3|wc
      2      2      4

```

```

ubuntu@ubuntu:~/Desktop$ ls -l
total 16
-rwxrwxrwx 1 ubuntu ubuntu 87 Apr 7 07:16 month.txt
-rwxr-xr-x 1 ubuntu ubuntu 8273 Apr 7 07:03 ubiquity.desktop
ubuntu@ubuntu:~/Desktop$ sudo chown root:root month.txt
ubuntu@ubuntu:~/Desktop$ ls -l
total 16
-rwxrwxrwx 1 root root 87 Apr 7 07:16 month.txt
-rwxr-xr-x 1 ubuntu ubuntu 8273 Apr 7 07:03 ubiquity.desktop
ubuntu@ubuntu:~/Desktop$ sudo chown ubuntu:ubuntu month.txt
ubuntu@ubuntu:~/Desktop$

```