$ cd subdirectory

$ mv filepath /path/to/destination/

LINUX CHEAT SHEET

SCALER Topics

$ cat file1 file2 > c.txt

$ ls /path/to/directory

File and Directory Management

cd command

The cd command is used to change the current working directory in the terminal.

Let's say your root directory contains the following files and folders.

```
.  ──────────────────────────────────────────→ This . represents the home or root directory of the linux machine.
├── Desktop
│   ├── important_report.txt
│   └── meeting_minutes.docx
├── Documents
│   ├── project_plan.pdf
│   └── research_paper.doc
├── Downloads
│   ├── software_installer.exe
│   └── vacation_photo_album.zip
├── Music                                  ←──────  Tree diagram of home directory.
│   ├── favorite_song.mp3
│   └── relaxing_playlist.m3u
├── Videos
│   ├── family_vacation.mp4
│   └── tutorial_series.mkv
└── Animals
    ├── animals.txt
    ├── Cats
    │   ├── cat1.png
    │   └── cat2.png
    └── Dogs
        ├── dog1.png
        └── dog2.png
```

Change to a specific directory:

Command

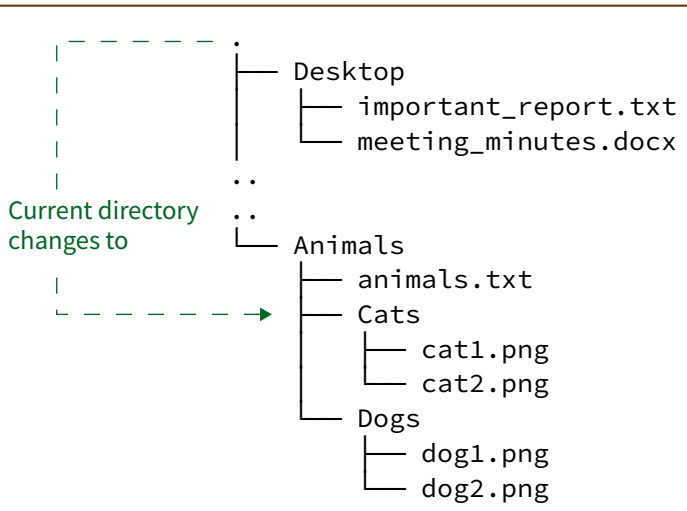
```
$ cd /path/to/directory
```

Description

This command changes the current directory to the one specified by the absolute path

Example

```
$ cd /home/Animals/Cats
```



Change to the home directory:

Command

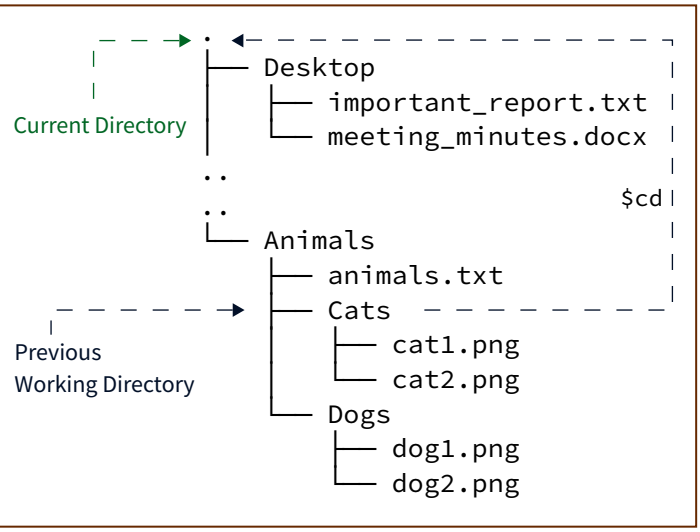
```
$ cd
```

Description

Running cd without any arguments takes you to your home directory from wherever you are.

Example

```
$ cd
```



Move up one directory (parent directory):

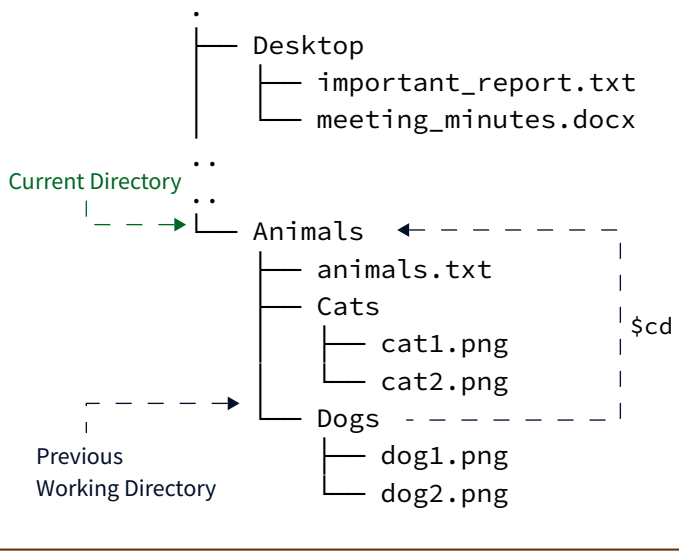
Command

```
$ cd ..
```

Description

This command moves you up one level in the directory hierarchy to the parent directory.

Example



Change to a subdirectory within the current directory:

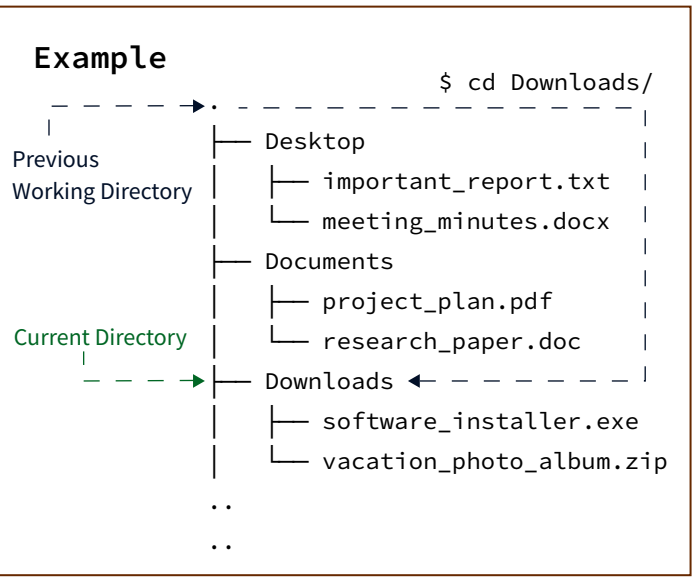
Command

```
$ cd subdirectory
```

Description

Here, subdirectory represents the name of the directory you want to change into.

Example



Use relative paths:

Command

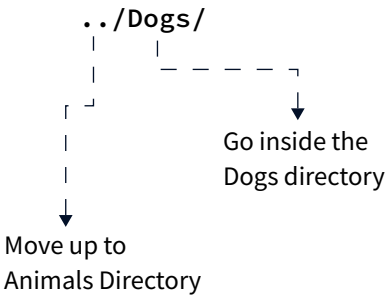
```
$ cd ../relative/path
```

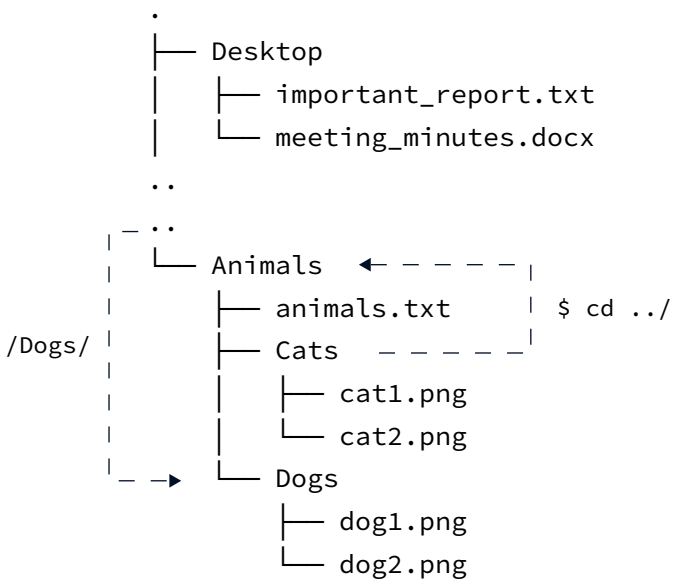
Description

You can use relative paths to navigate to a directory relative to your current location. .. represents the parent directory, and you can specify additional directories to traverse from there.

Example

Assume, your current directory is Cats directory.





ls (list all directories) command

The ls command in Linux is used to list the contents of a directory.

List files and directories in the current directory:

Command

```
$ ls
```

Description

Running ls without any arguments lists the files and directories in the current working directory.

Example

Let's say the current working directory is Animals.

```
$ ls  ----> animals.txt  Cats  Dogs
```

List files and directories in a specific directory:

Command

```
$ ls /path/to/directory
```

Description

This command lists the files and directories in the specified directory (/path/to/directory).

Example

Assuming the current working directory is home:

```
$ ls Documents/ --
                  |
                  |
                  v
project_plan.pdf  research_paper.doc
```

List files and directories in a long format:

Command

```
$ ls -l
```

Description

The -l option provides detailed information like file/folder permissions, owner, size, and modification timestamps.

Example

Assuming the current working directory is Animals:

```
                                                  --> Timestamps of file/folder creation
$ ls -l ----> total 12                            |
              -rw-rw-r-- 1 scaler scaler   23 May 29 14:57 animals.txt
              drwxrwxr-x 2 scaler scaler 4096 May 29 14:54 Cats
      ------- drwxrwxr-x 2 scaler scaler 4096 May 29 14:55 Dogs
     |                                      |
     v                                      ------> Linux username
Permissions of the user on these files/folders.
```

List files and directories including hidden files:

Command

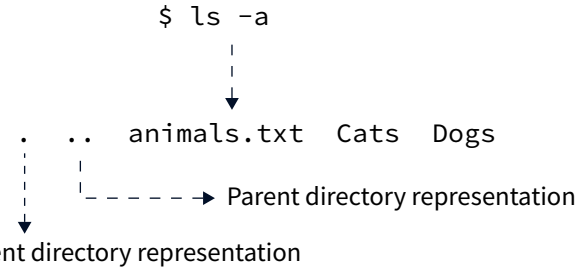
```
$ ls -a
```

Description

The -a option shows all files and directories, including those with names starting with a dot (hidden files).

Example

Assuming the current working directory is Animals:



List files and directories including hidden files:

Command

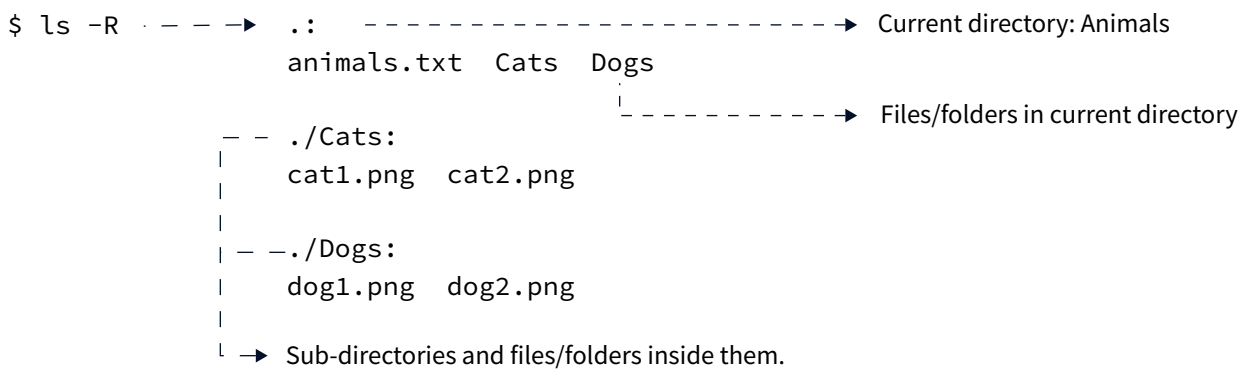
```
$ ls -R
```

Description

Lists all the files and directories inside the current directory as well as all the files and directories of all sub-directories.

Example

Assuming the current working directory is Animals:



pwd command

Command

```
$ pwd
```

Description

Shows the current directory path.

Example

Assuming the current working directory is Animals:

```
$ pwd --------> /home/scaler/Animals
```

mkdir command

Command

```
$ mkdir
```

Description

This command creates a directory.

Example

Let's say we are in the root directory.

```
$ mkdir Pictures ------> .
                        ├── Desktop
                        │   ├── important_report.txt
                        │   └── meeting_minutes.docx
                        ├── Documents
                        │   ├── project_plan.pdf
                        │   └── research_paper.doc
                        ..
                        ..
                        ├── Animals
                        │   ├── animals.txt
                        │   ├── Cats
                        │   │   ├── cat1.png
                        │   │   └── cat2.png
                        │   └── Dogs
                        │       ├── dog1.png
                        │       └── dog2.png
                        └── Pictures  ---------> New directory created!!
```

rmdir command

The rmdir command in Linux is used to remove directories (folders) that are empty. It can be used to delete directories that do not contain any files or subdirectories.

Command

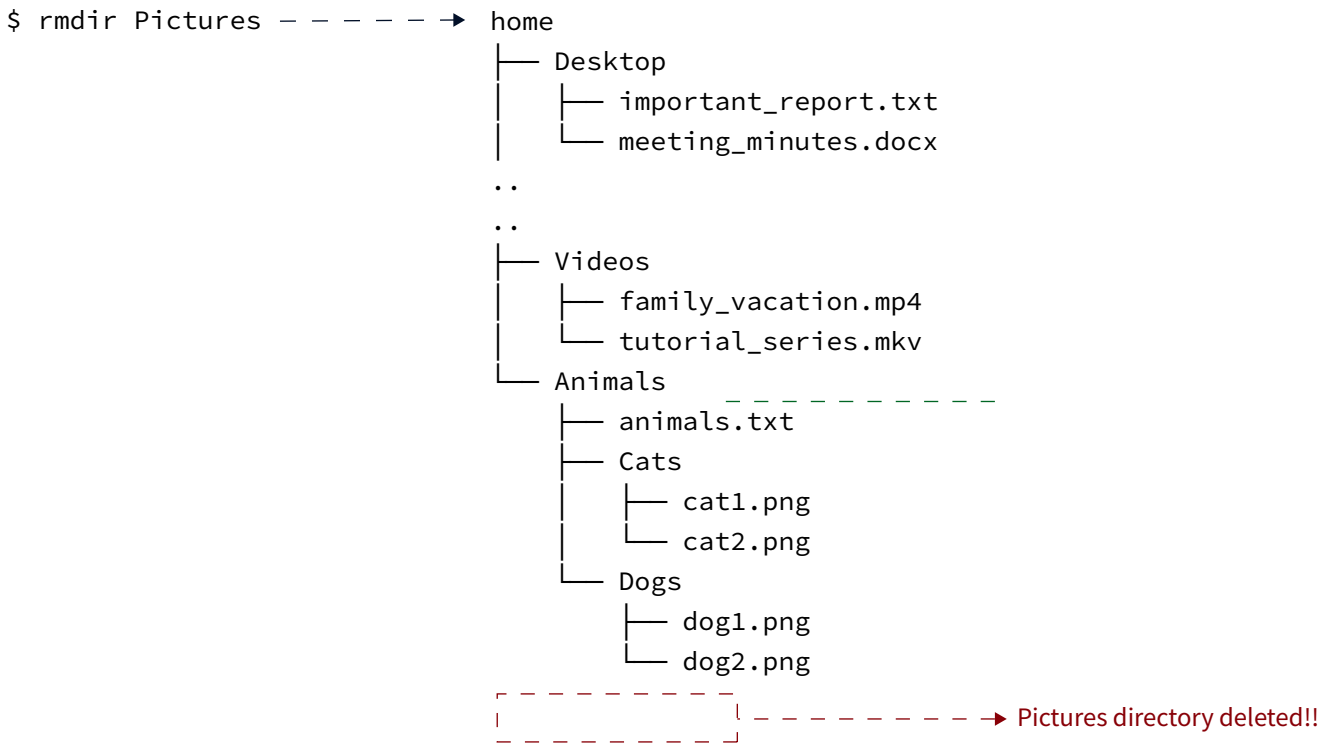
```
$ rmdir directory_name
```

Description

The rmdir command in Linux is used to remove directories (folders) that are empty.

Example

Let's delete the Pictures directory while we are in the root directory.



mv command

The mv command in Linux is used to move or rename files and directories.
It allows you to change the location of a file or directory within the file system or give it a new name.

Move a file to a different directory:

Command

```
$ mv filepath /path/to/destination/
```

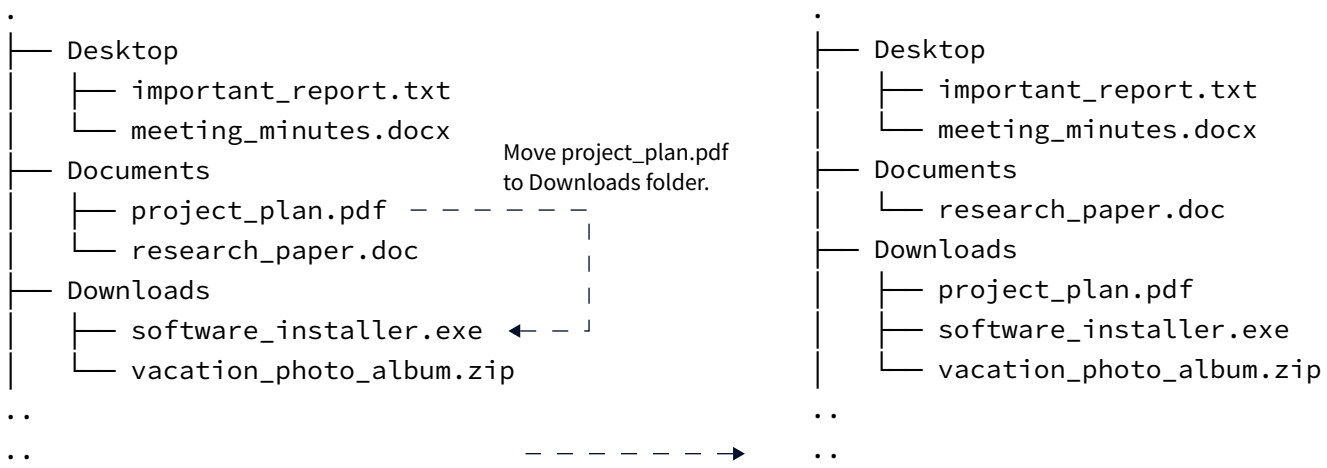
Description

This command moves the file with the given location to the directory specified by /path/to/destination/.

Example

Let's move the project_plan.pdf file to Downloads folder. Assume we are in the root directory.

```
$ mv Documents/project_plan.pdf Downloads/
```



Rename a file:

Command

```
$ mv old_file.txt new_file.txt
```

Description

By specifying a different name as the destination, you can rename the file old_file.txt to new_file.txt within the same directory.

Example

Let's rename the project_plan.pdf which is in Documents folder. Assume currently we are in Documents folder.

```
$ mv project_plan.pdf project_design.pdf
```

```
.
├── Desktop
│   ├── important_report.txt
│   └── meeting_minutes.docx
├── Documents
│   ├── project_design.pdf
│   └── research_paper.doc
├── Downloads
│   ├── software_installer.exe
│   └── vacation_photo_album.zip
..
..
```

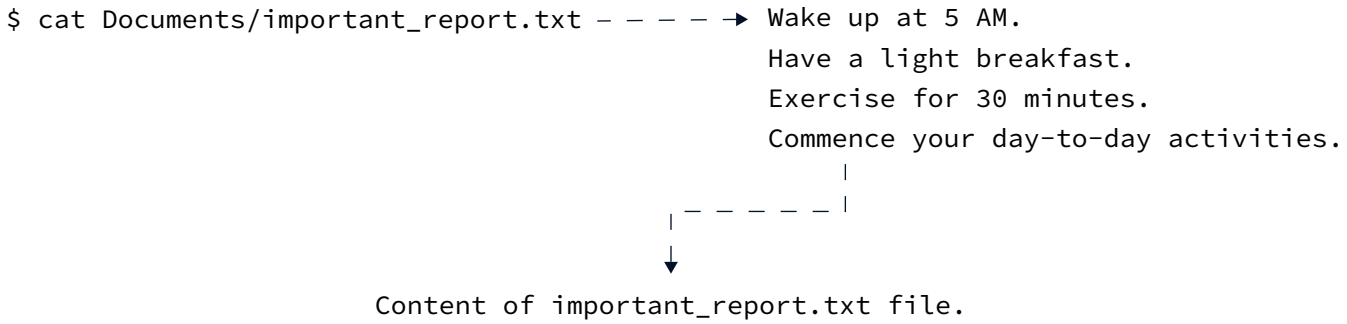
cat command

The cat command in Linux is used to concatenate and display the contents of files.
It can also be used to create new files or append to existing ones.

Display the contents of a file:	
Command `$ cat filename`	**Description** This command displays the contents of filename in the terminal.

Example

Let's see the content of important_report.txt file.



Concatenate multiple files:

Command

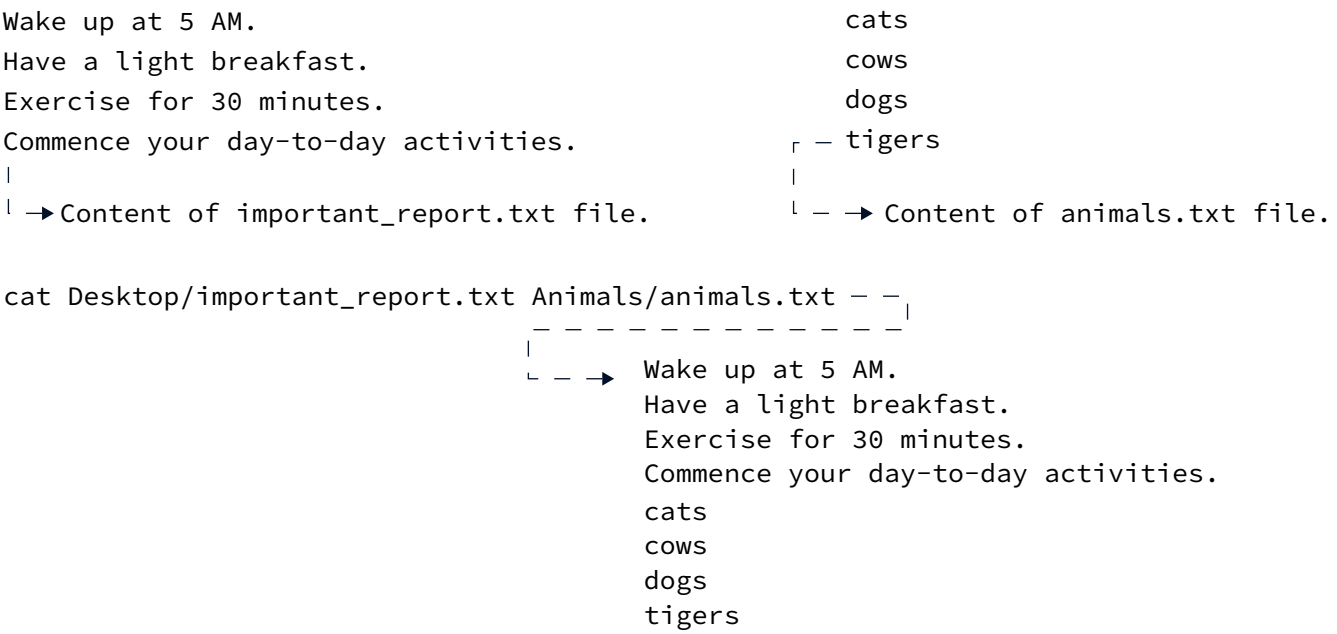
```
$ cat file1 file2
```

Description

You can concatenate multiple files by specifying their names consecutively. The output will be the combined contents of the files.

Example

Let's concatenate animals.txt and important_report.txt files.




```
cat Desktop/important_report.txt Animals/animals.txt
```

```
Wake up at 5 AM.
Have a light breakfast.
Exercise for 30 minutes.
Commence your day-to-day activities.
cats
cows
dogs
tigers
```

Output to a new file:

Command

```
$ cat file1 file2 > combined.txt
```

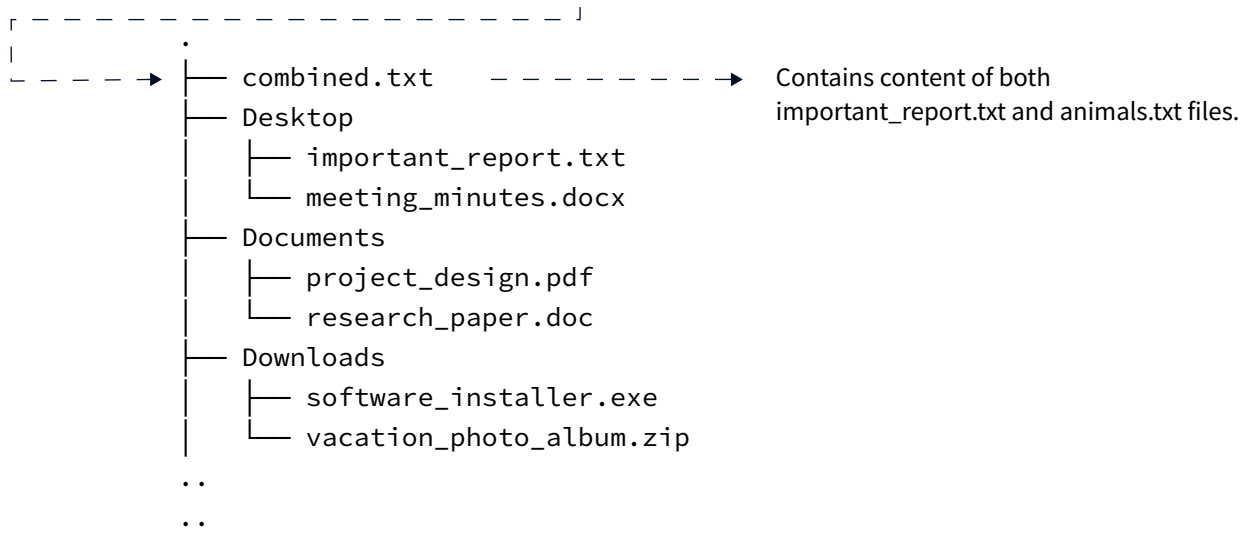
Description

The > symbol redirects the concatenated output of file1.txt and file2.txt to a new file named combined.txt.

Example

Let's concatenate animals.txt and important_report.txt and output it to combined.txt which is created in the current directory.

```
cat Desktop/important_report.txt Animals/animals.txt > combined.txt
```



Append to an existing file:

Command

```
$ cat >> existingfile.txt
```

Description

Anything you type will be appended to existingfile.txt.
Press Ctrl + D to save and exit.

Example

Let's add "Go to sleep by 10 PM" in the important_report.txt file.

```
cat >> Desktop/important_report.txt
Go to sleep by 10 PM.
```

```
Wake up at 5 AM.
Have a light breakfast.
Exercise for 30 minutes.
Commence your day-to-day activities.
Go to sleep by 10 PM.
```

→ Content of important_report.txt file.

grep command

The grep command in Linux is used to search for specific patterns or lines of text within files. It allows you to filter and extract information based on specified criteria.

Search for a pattern in a single file:

Command

```
$ grep pattern file
```

Description

This command searches for the given pattern within the given file and displays all lines that contain the pattern.

Example

Let's search for the line which contains "breakfast" in the file important_report.txt

```
grep "breakfast" Desktop/important_report.txt
```

Output → Have a light breakfast.

Search for lines starting with "Wake":

```
grep "^Wake" ~/Desktop/important_report.txt
```

Output → Wake up at 5 AM.

Display line numbers:

Command

```
$ grep -n pattern file
```

Description

The -n option adds line numbers to the output, indicating the line number of each matched line.

Example

Search for lines with line numbers that contain "a":

```
grep -n "a" Desktop/important_report.txt
```

Output

```
1:Wake up at 5 AM.
2:Have a light breakfast.
4:Commence your day-to-day activities.
```

Case-insensitive search:

Command

```
$ grep -i pattern file
```

Description

The -i option performs a case-insensitive search, ignoring the distinction between uppercase and lowercase letters.

Example

Search for lines with case-insensitive match for "am":

```
grep -i "am" Desktop/important_report.txt
```

Output

```
Wake up at 5 AM.
```

Invert match (exclude a pattern):

Command

```
$ grep -v pattern file
```

Description

The -v option displays lines that do not match the specified pattern, effectively excluding those lines from the output.

Example

Search for lines that do not contain the word "Exercise":

```
grep -v "Exercise" Desktop/important_report.txt
```

Output

```
Wake up at 5 AM.
Have a light breakfast.
Commence your day-to-day activities.
Go to sleep by 10 PM.
```

Match the pattern against all files in a directory

Command

```
$ grep -r pattern /path/to/directory/
```

Description

The -r option searches the pattern against all the files in the specified directory.

Example

Search for lines that do not contain the word "Exercise" in all the files inside root directory:
Assume there are only two files in the root directory: important_report.txt and animals.txt.

```
grep -r -v "Exercise" .
```

Output

```
./Desktop/important_report.txt:Wake up at 5 AM.
./Desktop/important_report.txt:Have a light breakfast.
./Desktop/important_report.txt:Commence your day-to-day activities.
./Desktop/important_report.txt:Go to sleep by 10 PM.
./Animals/animals.txt:cats
./Animals/animals.txt:cows
./Animals/animals.txt:dogs
./Animals/animals.txt:tigers
```

File and Directory Compression Commands

gzip command

Command	Description
`$ gzip file1`	This command is used to compress a file with gzip compression.

Example

Lets compress the project_plan.pdf file in Documents directory.

```
$ gzip project_plan.pdf  --- ls command --->  project_plan.pdf.gz   research_paper.doc
```

Basically, it has compressed project_plan.pdf and the hence, the new name is project_plan.pdf.gz

gunzip command

Command

```
$ gunzip fileDemo.gz
```

Description

This command is used to unzip a file that has gzip compression.

Example

Lets decompress the project_plan.pdf.gz file which we created in the last step.

```
$ gunzip project_plan.pdf.gz
        |
        | ls command
        ↓
project_plan.pdf   research_paper.doc
```

tar cf command

Command	Description
`$ tar cf myDir.tar myDir`	This command is used to create an uncompressed tar archive.

Example

Lets compress the Documents directory using the command above.

```
$ tar cf docs.tar Documents
        --- ls command --->
Animals        Desktop    Documents   Music      Videos
combined.txt   docs.tar   Downloads   Pictures
```

tar cfz command

Command

```
$ tar cfz demoDir.tar demoDir
```

Description

This command is used to create a tar archive with gzip compression.

Example

Lets compress the Documents directory with gzip compression using the command above.

```
$ tar cfz docs.tar Documents
```

ls command

```
Animals       Desktop   Documents  Music     Videos
combined.txt  docs.tar  Downloads  Pictures
```

tar xf file command

Command

```
$ tar xf demoFile
```

Description

This command is used to extract the contents of any type of tar archive.

Example

Let's decompress the Documents directory in the Downloads directory.
Assume current directory: Downloads

```
$ tar xf ../home/docs.tar
```

ls

```
Documents  software_installer.exe  vacation_photo_album.zip
```

Environment Variable Commands

env command

Command

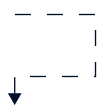
```
$ env
```

Description

This command displays all the environment variables.

Example

```
$ env
```



```
SHELL=/bin/bash
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
...
...
USER=scaler
GNOME_TERMINAL_SERVICE=:1.101
DISPLAY=:0
OLDPWD=/home/scaler/
_=/usr/bin/env
```

echo command

Command

```
$ echo ENV_VARIABLE NAME
```

Description

This command displays the environment variable.

Example

```
$ echo $SHELL  - - - - - - → /bin/bash
```

unset command

Command

```
$ unset
```

Description

This command removes a variable.

Network Configuration

ifconfig command

Command

```
$ ifconfig [interface] [options]
```

Description

This command is used to display information for all active network interfaces.

Options

`up or down`: Enables or disables the specified interface.

`address [address]`: Sets the IP address for the interface.

`netmask [netmask]`: Sets the network mask for the interface.

`broadcast [broadcast]`: Sets the broadcast address for the interface.

`hw ether [MAC address]`: Sets the MAC (Media Access Control) address for the interface.

`mtu [size]`: Sets the Maximum Transmission Unit (MTU) for the interface.

Example

```
$ ifconfig ---> docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
                        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
                        ether 02:42:a4:7a:5f:60  txqueuelen 0  (Ethernet)
                        RX packets 0  bytes 0 (0.0 B)
                        RX errors 0  dropped 0  overruns 0  frame 0
                        TX packets 0  bytes 0 (0.0 B)
                        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

                lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
                        inet 127.0.0.1  netmask 255.0.0.0
                        inet6 ::1  prefixlen 128  scopeid 0x10<host>
                ...

                virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
                        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
                        ether 52:54:00:71:49:3f  txqueuelen 1000  (Ethernet)
                        RX packets 0  bytes 0 (0.0 B)
                        RX errors 0  dropped 0  overruns 0  frame 0
                        TX packets 0  bytes 0 (0.0 B)
                        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Enable/Disable a network interface: `--------> sudo ifconfig eth0 up/down`

Set an IP address and netmask for a network interface: `---> sudo ifconfig eth0 192.168.0.10 netmask 255.255.255.0`

Set the Maximum Transmission Unit (MTU) for a network interface (e.g., eth0): `---> sudo ifconfig eth0 mtu 1500`

ping command

Command

```
$ ping [options] destination
```

Description

The ping command in Linux is used to test network connectivity by sending ICMP echo request packets to a specific destination.

Options

```
-c count: Specifies the number of ICMP echo request packets to send before stopping.
-i interval: Sets the interval (in seconds) between sending each packet.
-w deadline: Specifies the time (in seconds) to wait for a reply before stopping.
-s packetsize: Sets the size of the ICMP echo request packets to send.
-q: Quiet output, only displays summary statistics.
-v: Verbose output, displays more detailed information.
```

Example

Send 5 ICMP echo request packets with a packet size of 64 bytes to the IP address 8.8.8.8

```
ping -c 5 -s 64 8.8.8.8 ------> PING 8.8.8.8 (8.8.8.8) 64(92) bytes of data.
                                72 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=70.7 ms
                                72 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=69.0 ms
                                72 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=67.7 ms
                                72 bytes from 8.8.8.8: icmp_seq=4 ttl=113 time=65.3 ms
                                72 bytes from 8.8.8.8: icmp_seq=5 ttl=113 time=54.6 ms

                                --- 8.8.8.8 ping statistics ---
                                5 packets transmitted, 5 received, 0% packet loss, time 4006ms
                                rtt min/avg/max/mdev = 54.590/65.457/70.681/5.712 ms
```

Ping a domain name with a deadline of 3 seconds and display verbose output

```
ping -w 3 -v www.google.com --┐
┌---------------------------┘
└---> PING www.google.com(bom12s04-in-x04.1e100.net (2404:6800:4009:804::2004)) 56 data bytes
      64 bytes from bom12s04-in-x04.1e100.net (2404:6800:4009:804::2004): icmp_seq=1 ttl=58 time=64.3 ms
      64 bytes from bom12s04-in-x04.1e100.net (2404:6800:4009:804::2004): icmp_seq=2 ttl=58 time=72.8 ms
      64 bytes from bom12s04-in-x04.1e100.net (2404:6800:4009:804::2004): icmp_seq=3 ttl=58 time=61.5 ms

      --- www.google.com ping statistics ---
      3 packets transmitted, 3 received, 0% packet loss, time 2004ms
      rtt min/avg/max/mdev = 61.545/66.186/72.760/4.778 ms
```

ssh command

Command

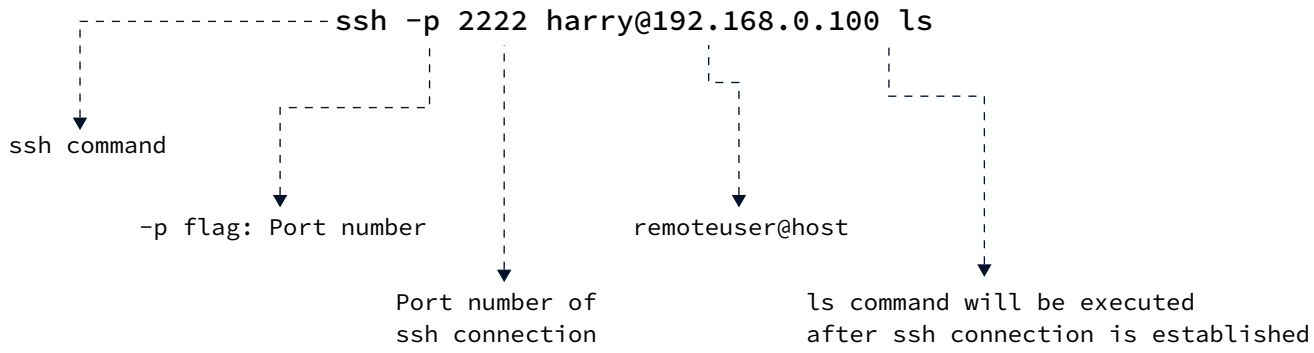
```
$ ssh [options] [user@]host [command]
```

Description

The ssh command is used in Linux to establish a secure shell (SSH) connection to a remote server or device.
It allows you to securely log in to a remote system and execute commands or transfer files over an encrypted connection.

`[options]`	Command line options to configure SSH connection
`[user@]`	Username of the remote machine. Defaults to host machine username.
`host`	IP address of the remote machine.
`[command]`	Command to run just after connection is established.

Example



Users and Permissions

adduser command

Command	Description
`$ sudo adduser username`	This command is used to add a user.

Example

This command is used to add a user.

```
$ sudo adduser scalertopics ---┐
                               |
                               |
                               |
                               └---> Adding user `scalertopics' ...
                                     Adding new group `scalertopics' (1002) ...
                                     Adding new user `scalertopics' (1002) with group `scalertopics' ...
                                     Creating home directory `/home/scalertopics' ...
                                     Copying files from `/etc/skel' ...
                                     New password:
                                     Retype new password:
                                     passwd: password updated successfully
                                     Changing the user information for scalertopics
                                     Enter the new value, or press ENTER for the default
                                             Full Name []: Scaler Academy
                                             Room Number []: BLR
                                             Work Phone []: NA
                                             Home Phone []: NA
                                             Other []: hello@scaler.com
                                     Is the information correct? [Y/n] Y
```

passwd command

Command	Description
`$ sudo passwd -l 'username'`	This command is used to change the password of a user.

Example

```
$ sudo passwd -l scalertopics --------> passwd: password expiry information changed.
```

userdel command

Command	Description
`$ sudo userdel -r 'username'`	This command is used to remove a newly created user.

Example

```
sudo userdel -r 'scalertopics'
```

usermod command

Command	Description
`$ sudo usermod -a -G GROUPNAME USERNAME`	This command is used to add a user to a particular group.

Example

The command to add scalertopics to root is shown.

```
$ sudo usermod -a -G root scalertopics ----------> scalertopics is the username
                                  |
                                  ---------------------------> root is the groupname
```

deluser command

Command	Description
`$ sudo deluser USER GROUPNAME`	This command is used to remove a user from a group.

Example

The command to remove scalertopics from the root group is shown.

```
$ sudo deluser scalertopics root ------------> Removing user `scalertopics' from group `root' ...
                                                Done.
scalertopics is the username <--  |
                                  ---------------------> root is the groupname
```

finger command

Command

```
$ finger
```

Description

This command shows the information of all the users logged in.

Example

```
$ finger ------>  Login          Name             Tty      Idle  Login Time     Office     Office Phone
                  scalertopics   Scaler Academy   *:0             May 29 13:12   BLR, NA
```

finger username command

Command

```
$ finger username
```

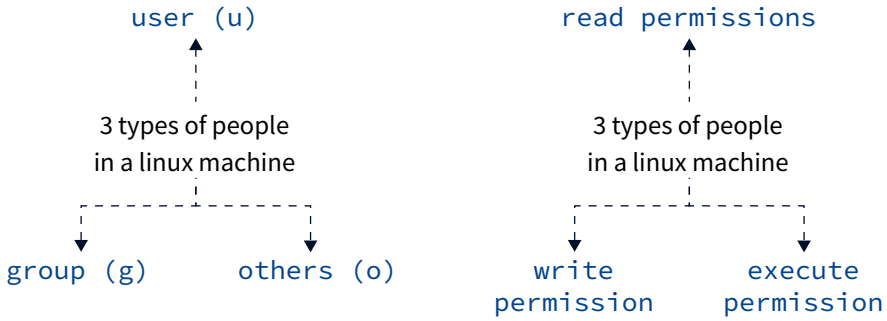
Description

This command gives information about a particular user.

Example

```
$ finger scalertopics ---------> Login: scalertopics                Name: Scaler Academy
                                 Directory: /home/scalertopics      Shell: /bin/bash
                                 Office: BLR, NA                    Home Phone: NA
                                 Never logged in.
                                 No mail.
                                 No Plan.
```

File Permissions



ls -l filename command

Command to view file permissions, owner, creation, modification timestamps.

Command

```
$ ls -l filename
```

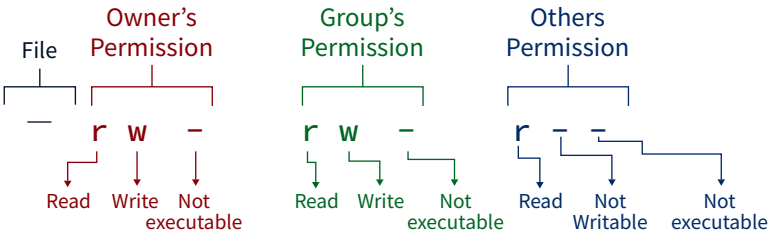
Description

This command is used to show the file permissions along with the owner and other details of the specified file.

Example

```
$ ls -l important_report.txt
-rw-rw-r-- 1 captain captain 125 May 29 20:43 important_report.txt
```

File Permissions for different users



chmod command

The chmod command stands for “change-mode” which means that using this command, we can change the mode in which some user is able to access the file.

Command

```
$ chmod [ugoa…][-+=]perms…[,....] FILE….
```

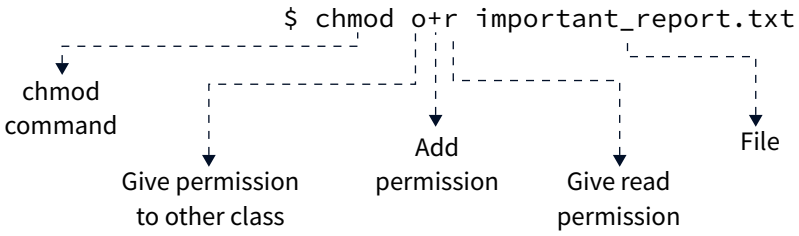
Description

Notations

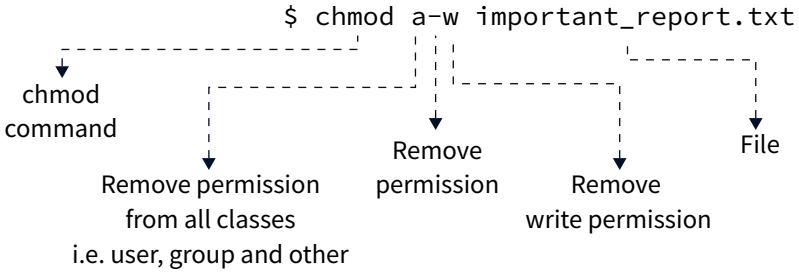
User Notation	Assignment Notation	Permission Notation
u -> user g -> group o -> other a -> All of the above, equivalent to ugo	+ -> Add specified permission - -> Remove specified permission = -> Change the current permission to specified permissions	r -> read w -> write x -> execute

Example

Let's add read permission to the other class for the file: important_report.txt
Current directory: Desktop



Let's add read permission to the other class for the file: important_report.txt
Current directory: Desktop



Numerical Method

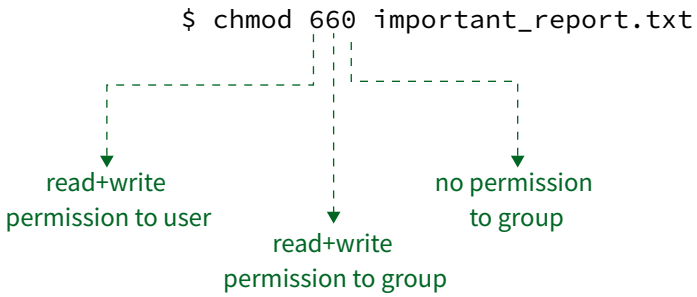
In numeric method, there are numeric codes for each permission.

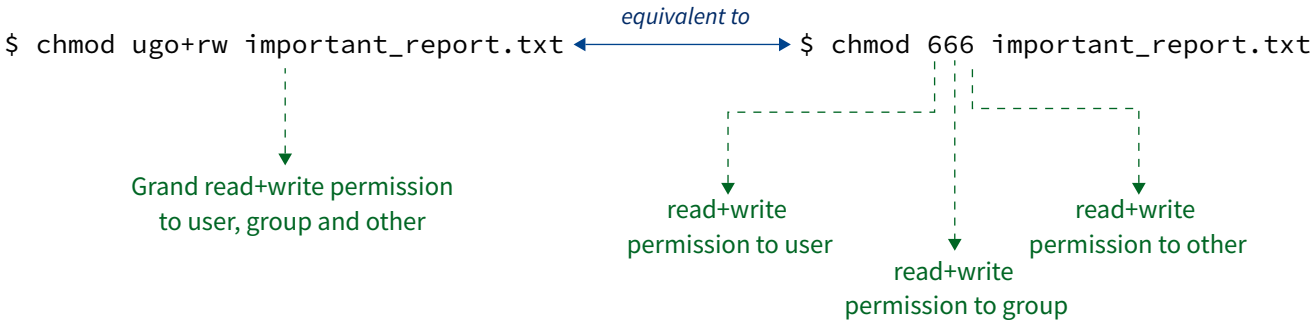
```
r (read) = 4
w (write) = 2
x (execute) = 1
No permissions  = 0
```

Example

If a user has read and write permission, then

Permission Number: 6 (read + write = 4 + 2 = 6)





System Information Commands

history command

Command

```
$ history
```

Description

This command displays the list of all the typed commands in the current terminal session.

Example

```
$ history ------------------>  1970  cd ../Animals/Dogs/
                               1971  cd ../Dogs/
                               1972  clear
                               1973  cd ..
                               1974  clear
                               1975  ls
                               1976  ls /home/captain/home/Documents
                               ...
                               ...
                               ...
                               2018  clear
                               2019  cd Desktop/
                               2020 history
```

clear command

Command

```
$ clear
```

Description

Clears the terminal i.e. no previous command will be visible on the screen now.

Example

```
$ history
...
...
1970  cd ../Animals/Dogs/
1971  cd ../Dogs/
1972  clear
1973  cd ..
1974  clear
1975  ls
1976  ls /home/captain/home/Documents
...
...
...
2018  clear
2019  cd Desktop/
2020 history
```

clear command ---------->

```
$ |
```

hostname command

Command

```
$ hostname
```

Description

Shows the name of the system host.

Example

```
$ hostname ------------->    scaler
```

hostid command

Command

```
$ hostid
```

Description

Shows the name of the system host.

Example

```
$ hostid    ------------->    520d3428
```

sudo command

Command

```
$ sudo
```

Description

Allows a regular user to run the programs with the security privileges of a superuser or root.

Example

You can not run the command $ apt-get update without administrator-level access.

```
$ apt-get update
    ----------------
    ---> Reading package lists... Done
         E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
```

However, using the sudo command, you can perform the required operation.

```
$ sudo apt-get update          ---------------------->    ...
[sudo] password for scaler:                               ...
                                                          ...
                                                          Metadata [940 B]
                                                          Fetched 12.7 MB in 37s (346 kB/s)
                                                          Reading package lists... Done
```

apt-get command

Command

```
$ apt-get
```

Description

This command is used to:

* Update package lists
* Upgrade installed packages
* Install a package
* Remove a package
* Purge a package (remove package and its configuration files)
* Search for a package
* Show information about a package
* List installed packages
* Upgrade the distribution to a new release
* Fix broken dependencies
* Clean the local repository cache

Example

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install package_name
$ sudo apt-get purge package_name
$ sudo apt-get remove package_name
```

date command

Command

```
$ date
```

Description

This command is used to show the current date and time.

Example

```
Monday 29 May 2023 10:47:58 PM IST
```

cal command

Command

```
$ cal
```

Description

Shows the calendar of the current month.

Example

```
      May 2023
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

whoami command

Command

```
$ whoami
```

Example

```
$ whoami  ----------------->  scaler
```

Description

This command displays the name with which you are logged in.

whereis command

Command

```
$ whereis [options] command
```

Description

The "whereis" command in Linux is used to locate the binary, source code, and manual page files for a given command or program.

Options

```
"-b": Searches only for executable files.
"-m": Searches only for manual page files.
"-s": Searches only for source code files.
"-u": Reports files that are unknown to the system's database
(located outside thestandard directories).
```

Example

```
$ whereis ls    ----------------------->   ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
```

Searching for the binary and manual page of the "gcc" command

```
$ whereis -b -m gcc ---------------------┐
       ┌--------------------------------┘
       ┆
       └----> gcc: /usr/bin/gcc /usr/lib/gcc /usr/share/gcc /usr/share/man/man1/gcc.1.gz
```

Process Commands

bg command

Command

```
$ bg %ID
```

Description

This command is used to send a process to the background.

Example

Move a specific job with job ID 2 to the background:

```
$ bg %2
```

fg command

Command

```
$ fg %ID
```

Description

This command is used to run a stopped process in the background.

Example

Move a specific job with job ID 2 to the background:

```
$ fg %1
```

top command

Command

```
$ top
```

Description

This command is used to get the details of all active processes.

Example

```
$ top ---------> PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
                2392 scaler    20   0 4652336 292348 106788 S   9.3   3.7   8:11.99 gnome-shell
                2252 scaler    20   0  526384  84936  48076 S   8.9   1.1   6:06.34 Xorg
                6206 scaler    20   0  816444  52048  39088 S   8.3   0.7   0:33.52 gnome-terminal-
                 237 root     -51   0       0      0      0 S   3.0   0.0   1:38.87 irq/128-ELAN066
                1097 mysql     20   0 2380348 397260  37608 S   1.0   5.0   1:34.41 mysqld
               97073 scaler    20   0   11992   3968   3188 R   0.7   0.1   0:00.25 top
                  40 root      20   0       0      0      0 S   0.3   0.0   0:03.56 ksoftirqd/4
                 877 root      20   0  273896  10636   9620 S   0.3   0.1   0:07.36 thermald
                 882 root      20   0 1473848  41016  14796 S   0.3   0.5   0:46.55 warp-svc
                1088 postgres  20   0  218288  29064  27028 S   0.3   0.4   0:00.25 postgres
                1454 root      20   0 1374764  12380  10756 S   0.3   0.2   0:16.54 teamviewerd
                2443 scaler    20   0  162836   7544   6756 S   0.3   0.1   0:04.84 at-spi2-registr
                2622 scaler    20   0  617944   4200   3864 S   0.3   0.1   0:10.68 warp-taskbar
```

ps command

Command

```
$ ps
```

Description

The "ps" command in Linux is used to provide information about the currently running processes on the system.

Example

```
$ ps ------> PID   TTY          TIME CMD
             6214  pts/0    00:00:00 bash
             97206 pts/0    00:00:00 ps
```

ps PID command

Command

```
$ ps PID
```

Description

This command gives the status of a particular process.

Example

```
$ ps 1 --------
   ------------
   |
   v
PID TTY      STAT   TIME COMMAND
  1 ?        Ss     0:02 /sbin/init splash
```

pidof command

Command

```
$ pidof PROCESSNAME
```

Description

This command is used to give the process ID of a particular process.

Example

```
$ pidof bash ----------------> 6214
```

Hardware Configuration Commands

cpu-info command

Command

```
$ cpu-info
```

Description

This command is used to display the information about your CPU. It can be used after installation of the necessary package using sudo apt install cpuinfo.

Example

```
$ cpu-info
```

```
Packages:
        0: Intel Core i5-8265U
Cores:
        0: 2 processors (0-1), Intel Kaby Lake
        1: 2 processors (2-3), Intel Kaby Lake
        2: 2 processors (4-5), Intel Kaby Lake
        3: 2 processors (6-7), Intel Kaby Lake
Logical processors:
        0: APIC ID 0x00000000
        1: APIC ID 0x00000001
        2: APIC ID 0x00000002
        3: APIC ID 0x00000003
        4: APIC ID 0x00000004
        5: APIC ID 0x00000005
        6: APIC ID 0x00000006
        7: APIC ID 0x00000007
```

free -h command

Command

```
$ free -h
```

Description

This command is used to display the free and used memory. The -h flag is used for converting the information (to be displayed) to human-readable form.

Example

```
$ cpu-info
```

```
              total        used        free      shared  buff/cache   available
Mem:          7.5Gi       2.0Gi       2.7Gi       420Mi       2.9Gi       4.9Gi
Swap:         2.0Gi          0B       2.0Gi
```

lsusb -tv command

Command

```
$ lsusb -tv
```

Description

List all the USB connected devices.

Example

```
$ lsusb -tv -----▸ /:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/6p, 10000M
                       ID 1d6b:0003 Linux Foundation 3.0 root hub
                   /:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/12p, 480M
                       ID 1d6b:0002 Linux Foundation 2.0 root hub
                       |__ Port 2: Dev 2, If 0, Class=Vendor Specific Class, Driver=, 480M
                           ID 27c6:55b4 Shenzhen Goodix Technology Co.,Ltd. Fingerprint Reader
                       |__ Port 4: Dev 3, If 1, Class=Video, Driver=uvcvideo, 480M
                           ID 04f2:b61e Chicony Electronics Co., Ltd
                       |__ Port 4: Dev 3, If 0, Class=Video, Driver=uvcvideo, 480M
                           ID 04f2:b61e Chicony Electronics Co., Ltd
                       |__ Port 10: Dev 4, If 0, Class=Wireless, Driver=btusb, 12M
                           ID 8087:0aaa Intel Corp.
                       |__ Port 10: Dev 4, If 1, Class=Wireless, Driver=btusb, 12M
                           ID 8087:0aaa Intel Corp.
```

cat /proc/meminfo command

Command

```
$ cat /proc/meminfo
```

Description

Gives the information about memory like total and occupied and so on.

Example

```
$ cat /proc/meminfo  ---------------------------▸ MemTotal:        7886836 kB
                                                  MemFree:         2807932 kB
                                                  MemAvailable:    5085552 kB
                                                  Buffers:          192036 kB
                                                  Cached:          2681972 kB
                                                  ...
                                                  ...
                                                  ...
                                                  Hugetlb:               0 kB
                                                  DirectMap4k:      365180 kB
                                                  DirectMap2M:     7770112 kB
                                                  DirectMap1G:     1048576 kB
```

du command

Command

```
$ du -h file/folder
```

Description

This command stands for disk usage and is used to estimate the space usage for a file or directory.

Example

```
$ du -h Downloads  ---------------------------------->  2.1M    Downloads/Wallpapers
                                                        200K    Downloads/Tunes
                                                        1.5M    Downloads/CVs
                                                        ...
                                                        ...
                                                        549M    Downloads
```

SCALER TOPICS

Unlock your potential in software development with
FREE COURSES from **SCALER TOPICS!**

Register now and take the first step towards your future Success!

C++ for Beginners

5.9k enrolled Free

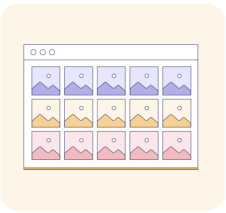
Java for Beginners

6.8k enrolled Free

That's not it. Explore 20+ Courses by clicking below

Explore Other Courses

Practice **CHALLENGES**
and become 1% better everyday



CIFAR-10 Image Classification Using PyTorch

Article

No. Of Questions : 3

Go to Challenge >

How to Build a Snake Game in JavaScript?

Article

No. Of Questions : 3

Go to Challenge >

Explore Other Challenges