Operating Systems Lab 01 (Workbook)

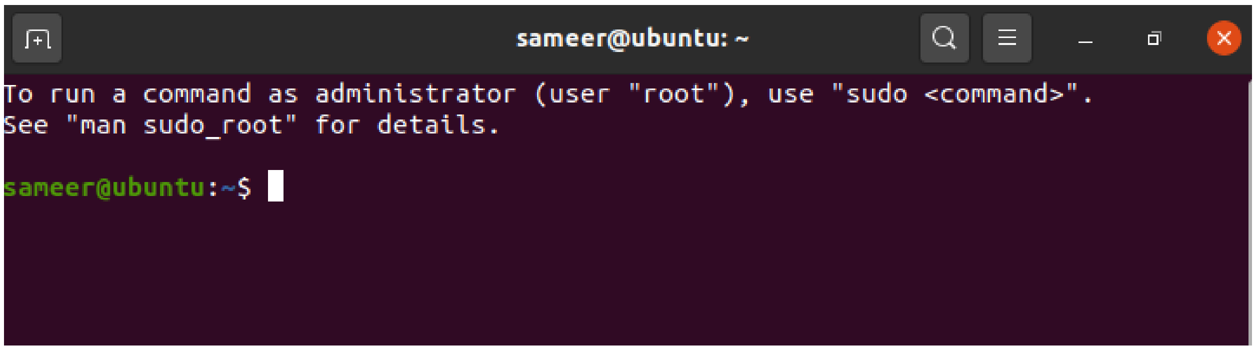
Course: Operating Systems (CL2006)
Instructor: Sameer Faisal

Semester: Fall 2025
T.A: N/A

Note:
- Maintain discipline during the lab.
- Listen and follow the instructions as they are given.
- Just raise hand if you have any problem.
- Completing all tasks of each lab is compulsory.
- Get your lab checked at the end of the session.

INTRODUCTION
Linux is an open source operating system (OS). An operating system is the software that directly manages a system's hardware and resources, like CPU, memory, and storage. The OS sits between applications and hardware and makes the connections between all of your software and the physical resources that do the work.

LINUX SHELL
Shell is a program that receives commands from the user and gives it to the OS to process, and it shows the output. Shell is command line interface (CLI), a console that interacts with the system via texts and processes. It's similar to the Command Prompt application in Windows.



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sameer@ubuntu:~$
```

LINUX COMMAND
A Linux command is a program or utility that runs on shell. Linux commands are executed on Terminal by pressing 'Enter' at the end of the line. User can run commands to perform various tasks, from package installation to user management and file manipulation.
Linux command's general syntax looks like:

CommandName [option(s)] [parameter(s)]

A command may contain an option or a parameter. In some cases, it can still run without them. These are the three most common parts of a command:

- **CommandName:** is the rule that you want to perform**.**
- **Option or flag:** modifies a command's operation. To invoke it, use hyphens (–) or double hyphens (—).
- **Parameter or argument:** specifies any necessary information for the command.

BASIC LINUX COMMANDS

From here the reader is exposed to the basic Linux commands. All the commands have to be tried in the terminal. Throughout the lab manuals Ubuntu will be used for explaining the concepts.

NOTE: All Linux commands are case sensitive i.e. 'cp' is not equivalent to 'CP'. Also, all the files and directories in linux are case sensitive so for example '/etc/hosts' is not equivalent to '/etc/Hosts' and so hosts and Hosts are two different files in the same directory. When executing multiple commands in a single line, use ; to separate them.

1. TERMINAL RELATED COMMANDS

Following are the terminal related commands:

- man
- clear
- exit
- history
- echo
- alias, unalias
- | **(**known as pipe operator)
- **&&**

a. man command

The man command provides a user manual of any commands or utilities you can run in Terminal, including the name, description, and options.
It consists of nine sections:

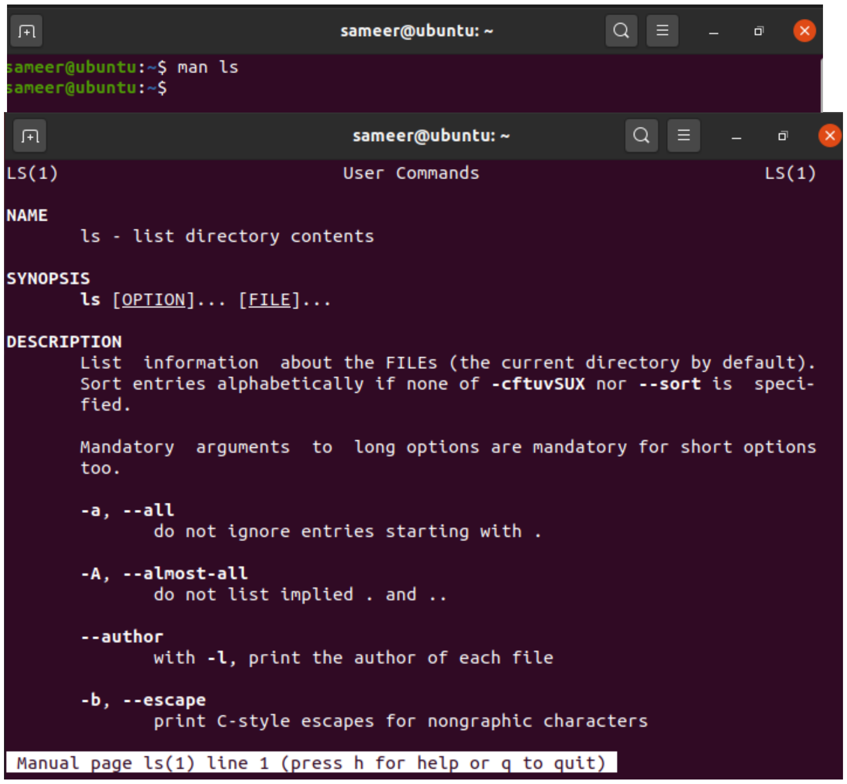
I. Executable programs or shell commands.
II. System Calls.
III. Library Calls.
IV. Games.
V. Special Files.
VI. File formats and conventions.
VII. System administration commands.
VIII. Kernel routines.
IX. Miscellaneous.

To display the complete manual, enter:

man man[command_name]

For example, you want to access the manual for the ls command:

man ls



```
sameer@ubuntu:~$ man ls
sameer@ubuntu:~$
```

```
LS(1)                          User Commands                          LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort is  speci-
       fied.

       Mandatory  arguments  to  long options are mandatory for short options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

 Manual page ls(1) line 1 (press h for help or q to quit)
```

Enter this command if you want to specify the displayed section: **man**

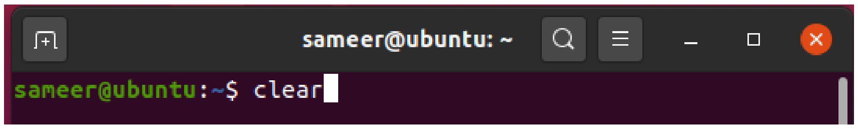
[option] [section_number] [command_name]

For instance, you want to see section 2 of the ls command manual:

man man 2 ls

b. clear command

Enter the clear command to clean the Terminal screen**.**



```
sameer@ubuntu:~$ clear
```

c. exit command

Enter the exit command to exit the terminal.

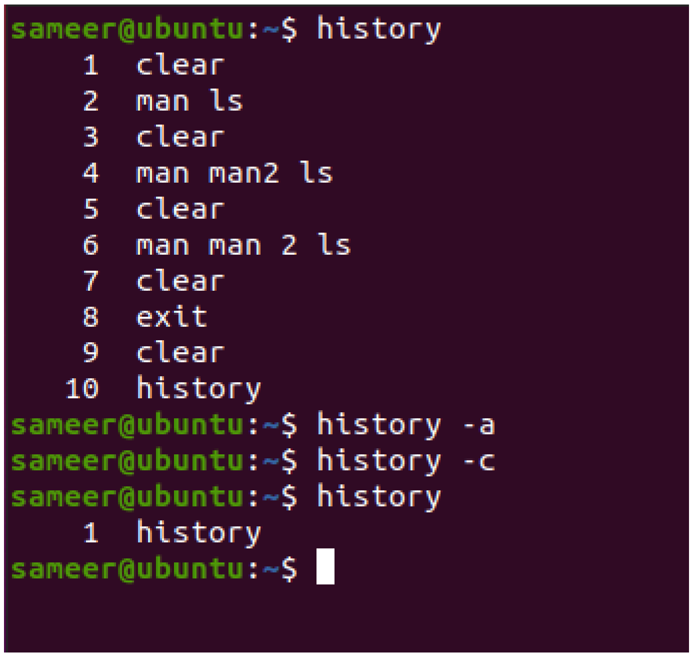
d. history command

With history, the system will list up to 500 previously executed commands, allowing you to reuse them without re-entering. Keep in mind that only users with sudo privileges can execute this command. How this utility runs also depends on which Linux shell you use. To run it, enter the command below:

history [option]

This command supports many options, such as:

- **-c** clears the complete history list.
- **-d** offset deletes the history entry at the OFFSET position.
- **-a** appends history lines.



```
sameer@ubuntu:~$ history
    1  clear
    2  man ls
    3  clear
    4  man man2 ls
    5  clear
    6  man man 2 ls
    7  clear
    8  exit
    9  clear
   10  history
sameer@ubuntu:~$ history -a
sameer@ubuntu:~$ history -c
sameer@ubuntu:~$ history
    1  history
sameer@ubuntu:~$ 
```

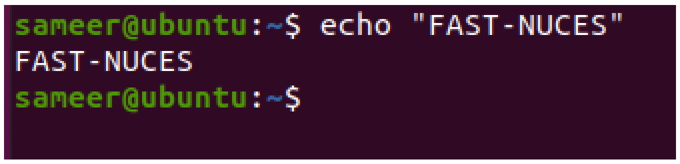
e. echo command

The echo command is a built-in utility that displays a line of text or string using the standard output. Here’s the basic syntax:

echo [option] [string]

For example, you can display the text FAST-NUCES by entering:

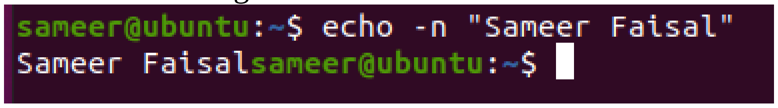
echo “FAST-NUCES”



```
sameer@ubuntu:~$ echo "FAST-NUCES"
FAST-NUCES
sameer@ubuntu:~$
```

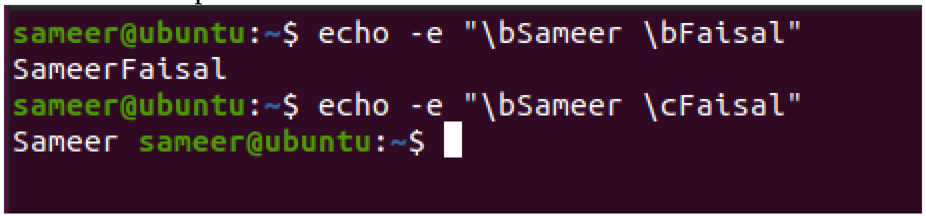
This command supports many options, such as:

- **-n** displays the output without the trailing newline.



```
sameer@ubuntu:~$ echo -n "Sameer Faisal"
Sameer Faisalsameer@ubuntu:~$ 
```

- **-e** enables the interpretation of the following backslash escapes:
    - **\a** plays sound alert.
    - **\b** removes spaces in between a text.
    - **\c** produces no further output.



```
sameer@ubuntu:~$ echo -e "\bSameer \bFaisal"
SameerFaisal
sameer@ubuntu:~$ echo -e "\bSameer \cFaisal"
Sameer sameer@ubuntu:~$ 
```

- **-E** displays the default option and disables the interpretation of backslash escapes.

f. alias, unalias command

alias allows you to create a shortcut with the same functionality as a command, file name, or text. When executed, it instructs the shell to replace one string with another.
To use the alias command, enter this syntax:

alias Name=String

For example, you want to make e the alias for the exit command:

alias e=exit

On the other hand, the unalias command deletes an existing alias.
Here’s what the general syntax looks like:

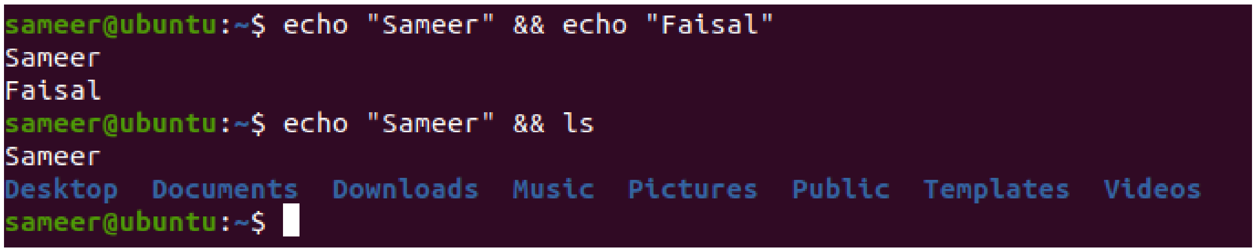
unalias [alias_name]

g. | (known as pipe operator) command

If a user in Linux likes to combine two or more commands, pipes is the option. Pipe is represented by the symbol ‘|’

h. && command

And command && is used to only allow the next command to run if the previous one is successful.



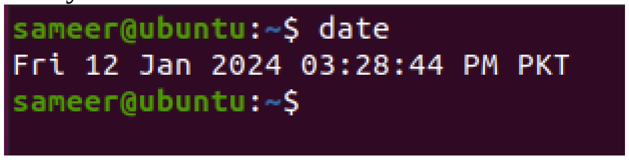
```
sameer@ubuntu:~$ echo "Sameer" && echo "Faisal"
Sameer
Faisal
sameer@ubuntu:~$ echo "Sameer" && ls
Sameer
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
sameer@ubuntu:~$ 
```

2. SYSTEM RELATED COMMANDS

Commands related to system includes some basic commands and commands for managing users and groups in Linux that require root access.
Following are the basic system related commands:

a. date command

The date command will display the system date.



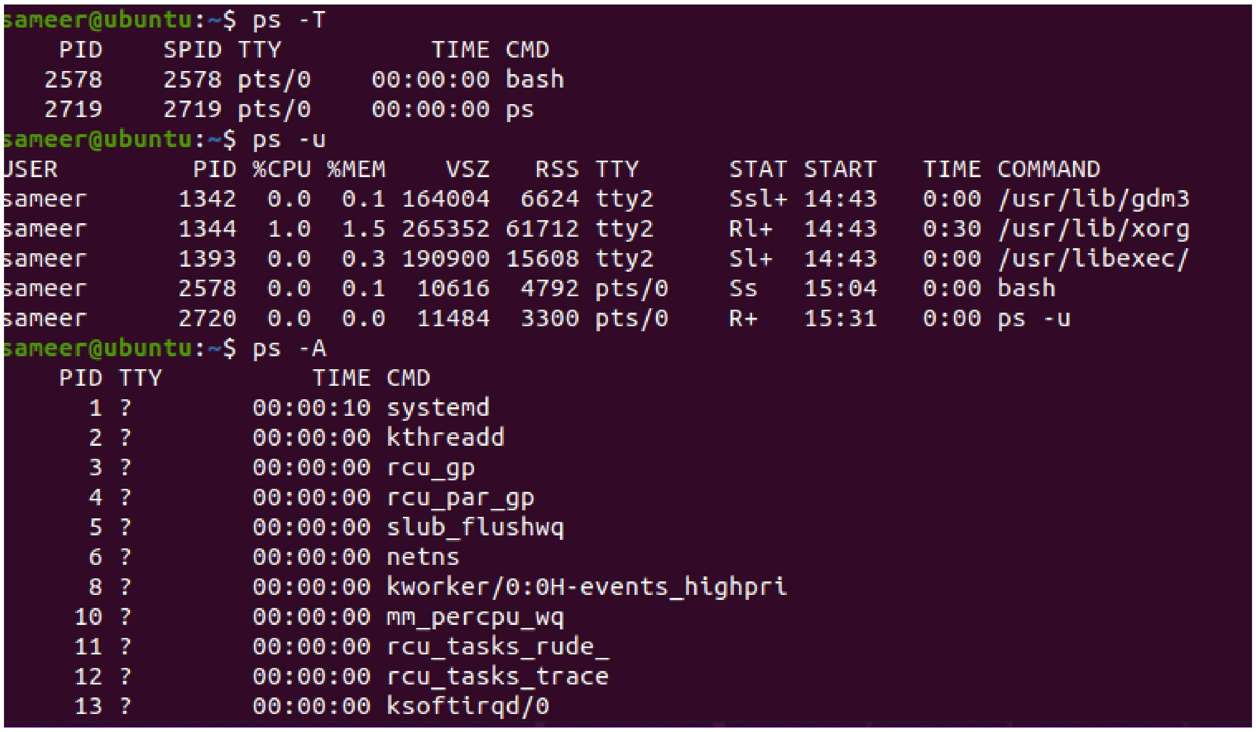
```
sameer@ubuntu:~$ date
Fri 12 Jan 2024 03:28:44 PM PKT
sameer@ubuntu:~$
```

b. ps command

The process status or ps command produces a snapshot of all running processes in your system. The static results are taken from the virtual files in the /proc file system.
Executing the ps command without an option or argument will list the running processes in the shell along with:

- The unique process ID (PID)
- The type of the terminal (TTY)
- The running time (TIME)
- The command that launches the process (CMD) Here are some acceptable options you can use:

  - **-T** displays all processes associated with the current shell session.
  - **-u** username lists processes associated with a specific user.
  - **-A** or **-e** shows all the running processes.



```
sameer@ubuntu:~$ ps -T
    PID    SPID TTY          TIME CMD
   2578    2578 pts/0    00:00:00 bash
   2719    2719 pts/0    00:00:00 ps
sameer@ubuntu:~$ ps -u
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
sameer      1342  0.0  0.1 164004  6624 tty2     Ssl+ 14:43   0:00 /usr/lib/gdm3
sameer      1344  1.0  1.5 265352 61712 tty2     Rl+  14:43   0:30 /usr/lib/xorg
sameer      1393  0.0  0.3 190900 15608 tty2     Sl+  14:43   0:00 /usr/libexec/
sameer      2578  0.0  0.1  10616  4792 pts/0    Ss   15:04   0:00 bash
sameer      2720  0.0  0.0  11484  3300 pts/0    R+   15:31   0:00 ps -u
sameer@ubuntu:~$ ps -A
    PID TTY          TIME CMD
      1 ?        00:00:10 systemd
      2 ?        00:00:00 kthreadd
      3 ?        00:00:00 rcu_gp
      4 ?        00:00:00 rcu_par_gp
      5 ?        00:00:00 slub_flushwq
      6 ?        00:00:00 netns
      8 ?        00:00:00 kworker/0:0H-events_highpri
     10 ?        00:00:00 mm_percpu_wq
     11 ?        00:00:00 rcu_tasks_rude_
     12 ?        00:00:00 rcu_tasks_trace
     13 ?        00:00:00 ksoftirqd/0
```

c. kill command

Use the kill command to terminate an unresponsive program manually. It will signal misbehaving applications and instruct them to close their processes. To kill a program, you must know its process identification number (PID). If you don't know the PID, run the following command:

ps ux

After knowing what signal to use and the program's PID, enter the following syntax:

kill [signal_option] pid

There are 64 signals that you can use, but these two are among the most commonly used:

SIGTERM requests a program to stop running and gives it some time to save all of its progress. The system will use this by default if you don't specify the signal when entering the kill command.

SIGKILL forces programs to stop, and you will lose unsaved progress. For example, the program's PID is 63773, and you want to force it to stop.

kill SIGKILL 63773 5.

d. df command

Use the df command to report the system's disk space usage, shown in percentage and kilobyte (KB). Here's the general syntax:

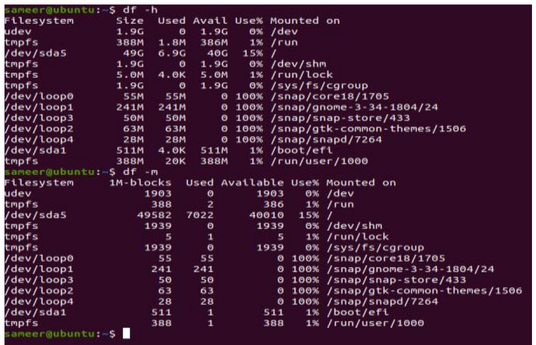
df [options] [file]

For example, enter the following command if you want to see the current directory's system disk space usage in a human-readable format:

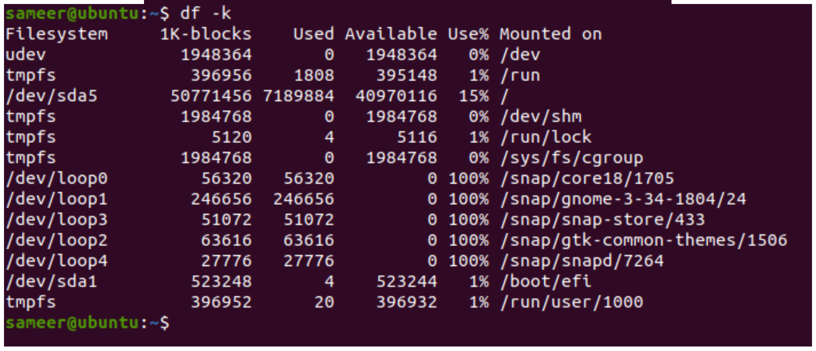
df –h

These are some acceptable options to use:

df -m displays information on the file system usage in MBs.
df -k displays file system usage in KBs. df -T shows the file system type in a new column..



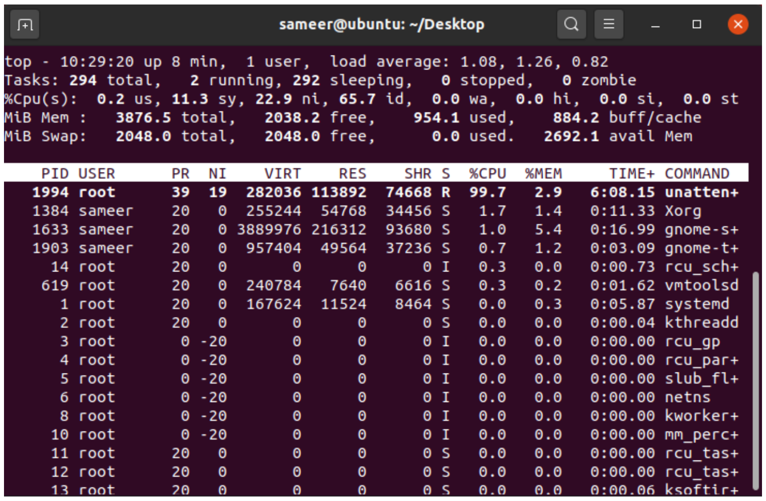
```
sameer@ubuntu:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           388M  1.8M  386M   1% /run
/dev/sda5        49G  6.9G   40G  15% /
tmpfs           1.9G     0  1.9G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/loop0       55M   55M     0 100% /snap/core18/1705
/dev/loop1      241M  241M     0 100% /snap/gnome-3-34-1804/24
/dev/loop3       50M   50M     0 100% /snap/snap-store/433
/dev/loop2       63M   63M     0 100% /snap/gtk-common-themes/1506
/dev/loop4       28M   28M     0 100% /snap/snapd/7264
/dev/sda1       511M  4.0K  511M   1% /boot/efi
tmpfs           388M   20K  388M   1% /run/user/1000
sameer@ubuntu:~$ df -m
Filesystem     1M-blocks  Used Available Use% Mounted on
udev                1903     0      1903   0% /dev
tmpfs                388     2       386   1% /run
/dev/sda5          49582  7022     40010  15% /
tmpfs               1939     0      1939   0% /dev/shm
tmpfs                  5     1         5   1% /run/lock
tmpfs               1939     0      1939   0% /sys/fs/cgroup
/dev/loop0            55    55         0 100% /snap/core18/1705
/dev/loop1           241   241         0 100% /snap/gnome-3-34-1804/24
/dev/loop3            50    50         0 100% /snap/snap-store/433
/dev/loop2            63    63         0 100% /snap/gtk-common-themes/1506
/dev/loop4            28    28         0 100% /snap/snapd/7264
/dev/sda1            511     1       511   1% /boot/efi
tmpfs                388     1       388   1% /run/user/1000
sameer@ubuntu:~$
```



```
sameer@ubuntu:~$ df -k
Filesystem     1K-blocks    Used Available Use% Mounted on
udev             1948364       0   1948364   0% /dev
tmpfs             396956    1808    395148   1% /run
/dev/sda5       50771456 7189884  40970116  15% /
tmpfs            1984768       0   1984768   0% /dev/shm
tmpfs               5120       4      5116   1% /run/lock
tmpfs            1984768       0   1984768   0% /sys/fs/cgroup
/dev/loop0         56320   56320         0 100% /snap/core18/1705
/dev/loop1        246656  246656         0 100% /snap/gnome-3-34-1804/24
/dev/loop3         51072   51072         0 100% /snap/snap-store/433
/dev/loop2         63616   63616         0 100% /snap/gtk-common-themes/1506
/dev/loop4         27776   27776         0 100% /snap/snapd/7264
/dev/sda1         523248       4    523244   1% /boot/efi
tmpfs             396952      20    396932   1% /run/user/1000
sameer@ubuntu:~$
```

e. top command

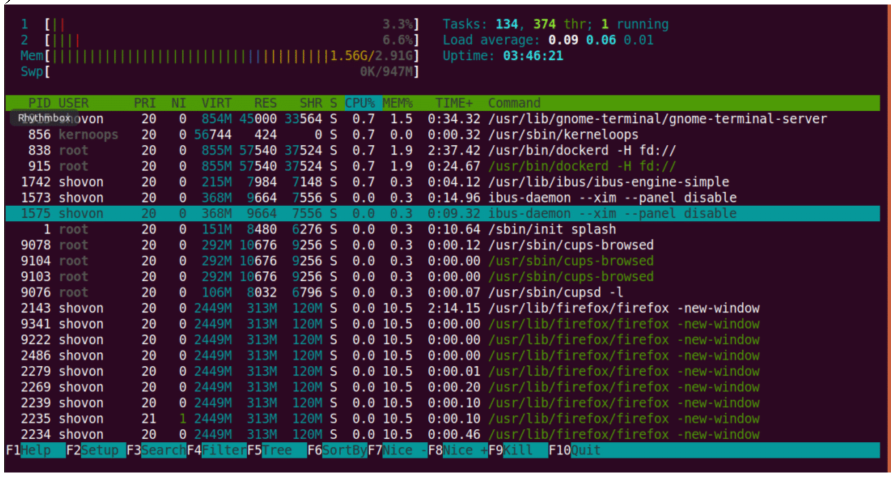
The top command in Linux Terminal will display all the running processes and a dynamic real-time view of the current system. It sums up the resource utilization, from CPU to memory usage.
The top command can also help you identify and terminate a process that may use too many system resources.


sameer@ubuntu: ~/Desktop

```
top - 10:29:20 up 8 min,  1 user,  load average: 1.08, 1.26, 0.82
Tasks: 294 total,   2 running, 292 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.2 us, 11.3 sy, 22.9 ni, 65.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3876.5 total,   2038.2 free,    954.1 used,    884.2 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   2692.1 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1994 root      39  19  282036 113892  74668 R  99.7   2.9   6:08.15 unatten+
   1384 sameer    20   0  255244  54768  34456 S   1.7   1.4   0:11.33 Xorg
   1633 sameer    20   0 3889976 216312  93680 S   1.0   5.4   0:16.99 gnome-s+
   1903 sameer    20   0  957404  49564  37236 S   0.7   1.2   0:03.09 gnome-t+
     14 root      20   0       0      0      0 I   0.3   0.0   0:00.73 rcu_sch+
    619 root      20   0  240784   7640   6616 S   0.3   0.2   0:01.62 vmtoolsd
      1 root      20   0  167624  11524   8464 S   0.0   0.3   0:05.87 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.04 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_fl+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.06 ksoftir+
```

f. htop command

The htop command is an interactive program that monitors system resources and server processes in real time. It is available on most Linux distributions, and you can install it using the default package manager (use “sudo apt install htop” to install).



```
  1  [||                                       3.3%]   Tasks: 134, 374 thr; 1 running
  2  [||||                                     6.6%]   Load average: 0.09 0.06 0.01
  Mem[|||||||||||||||||||||||||||||||||||||1.56G/2.91G]   Uptime: 03:46:21
  Swp[                                      0K/947M]

   PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 Rhythmbox ovon   20   0  854M 45000 33564 S  0.7  1.5  0:34.32 /usr/lib/gnome-terminal/gnome-terminal-server
   856 kernoops   20   0 56744   424     0 S  0.7  0.0  0:00.32 /usr/sbin/kerneloops
   838 root       20   0  855M 57540 37524 S  0.7  1.9  2:37.42 /usr/bin/dockerd -H fd://
   915 root       20   0  855M 57540 37524 S  0.7  1.9  0:24.67 /usr/bin/dockerd -H fd://
  1742 shovon     20   0  215M  7984  7148 S  0.7  0.3  0:04.12 /usr/lib/ibus/ibus-engine-simple
  1573 shovon     20   0  368M  9664  7556 S  0.0  0.3  0:14.96 ibus-daemon --xim --panel disable
  1575 shovon     20   0  368M  9664  7556 S  0.0  0.3  0:09.32 ibus-daemon --xim --panel disable
     1 root       20   0  151M  8480  6276 S  0.0  0.3  0:10.64 /sbin/init splash
  9078 root       20   0  292M 10676  9256 S  0.0  0.3  0:00.12 /usr/sbin/cups-browsed
  9104 root       20   0  292M 10676  9256 S  0.0  0.3  0:00.00 /usr/sbin/cups-browsed
  9103 root       20   0  292M 10676  9256 S  0.0  0.3  0:00.00 /usr/sbin/cups-browsed
  9076 root       20   0  106M  8032  6796 S  0.0  0.3  0:00.07 /usr/sbin/cupsd -l
  2143 shovon     20   0 2449M  313M  120M S  0.0 10.5  2:14.15 /usr/lib/firefox/firefox -new-window
  9341 shovon     20   0 2449M  313M  120M S  0.0 10.5  0:00.00 /usr/lib/firefox/firefox -new-window
  9222 shovon     20   0 2449M  313M  120M S  0.0 10.5  0:00.00 /usr/lib/firefox/firefox -new-window
  2486 shovon     20   0 2449M  313M  120M S  0.0 10.5  0:00.00 /usr/lib/firefox/firefox -new-window
  2279 shovon     20   0 2449M  313M  120M S  0.0 10.5  0:00.01 /usr/lib/firefox/firefox -new-window
  2269 shovon     20   0 2449M  313M  120M S  0.0 10.5  0:00.20 /usr/lib/firefox/firefox -new-window
  2239 shovon     20   0 2449M  313M  120M S  0.0 10.5  0:00.10 /usr/lib/firefox/firefox -new-window
  2235 shovon     21   1 2449M  313M  120M S  0.0 10.5  0:00.10 /usr/lib/firefox/firefox -new-window
  2234 shovon     20   0 2449M  313M  120M S  0.0 10.5  0:00.46 /usr/lib/firefox/firefox -new-window
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Compared to the top command, htop has many improvements and additional features, such as mouse operation and visual indicators. To use it, run the following command:

htop [options]

You can also add options, such as:

-d or –**delay** shows the delay between updates in tenths of seconds.

-C or –**no-color** enables the monochrome mode.

-h or –**help** displays the help message and exit.

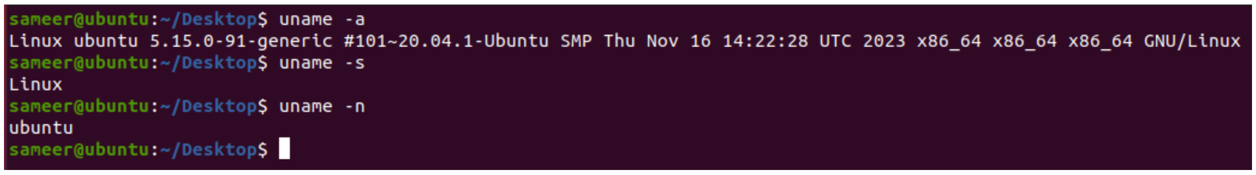
g. uname command

The uname or unix name command will print detailed information about your Linux system and hardware. This includes the machine name, operating system, and kernel. To run this command, simply enter uname into your CLI.

Here’s the basic syntax:

uname [option]

These are the acceptable options to use:
-a prints all the system information.
-s prints the kernel name.
-n prints the system's node hostname.



```
sameer@ubuntu:~/Desktop$ uname -a
Linux ubuntu 5.15.0-91-generic #101~20.04.1-Ubuntu SMP Thu Nov 16 14:22:28 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
sameer@ubuntu:~/Desktop$ uname -s
Linux
sameer@ubuntu:~/Desktop$ uname -n
ubuntu
sameer@ubuntu:~/Desktop$ 
```

h. shutdown, poweroff, reboot command

Command	Switch	Description	Example	Output
shutdown	None	Power off the computer	shutdown now	The system will shut down now for maintenance
	-r	Reboots after shutdown	shutdown –r now	None
poweroff	none	Shutdowns computer	sudo poweroff	None
reboot	none	Restarts computer	sudo reboot	None

3. MANAGING USERS AND GROUPS IN LINUX

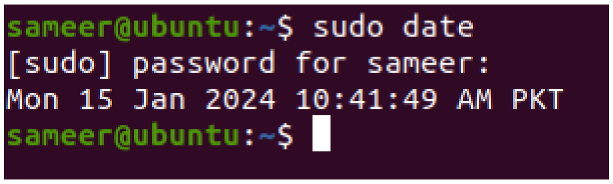
For root privilege, sudo command is used.

sudo command

Short for superuser do, sudo is one of the most popular basic Linux commands that lets you perform tasks that require administrative or root permissions.
When using sudo, the system will prompt users to authenticate themselves with a password. Then, the Linux system will log a timestamp as a tracker. By default, every root user can run sudo commands for 15 minutes/session.
If you try to run sudo in the command line without authenticating yourself, the system will log the activity as a security event.
Here's the general syntax:

sudo (command)

You can also add an option, such as:
-k or **–reset-timestamp** invalidates the timestamp file.
-g or **–group=group** runs commands as a specified group name or ID.
-h or **–host=host** runs commands on the host.



```
sameer@ubuntu:~$ sudo date
[sudo] password for sameer:
Mon 15 Jan 2024 10:41:49 AM PKT
sameer@ubuntu:~$
```

4. NETWORK RELATED COMMANDS

Linux terminal provides commands for interfacing with networks, some of the basic commands are:

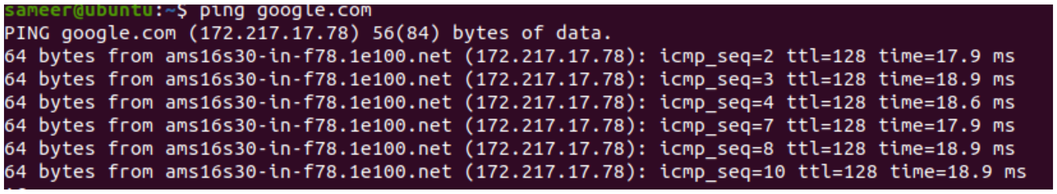
1. ping
2. wget
3. hostname
4. ip

a. ping command

The ping command is one of the most used basic Linux commands for checking whether a network or a server is reachable. In addition, it is used to troubleshoot various connectivity issues.
Here's the general format:

ping [option] [hostname_or_IP_address]

For example, you want to know whether you can connect to Google and measure its response time:



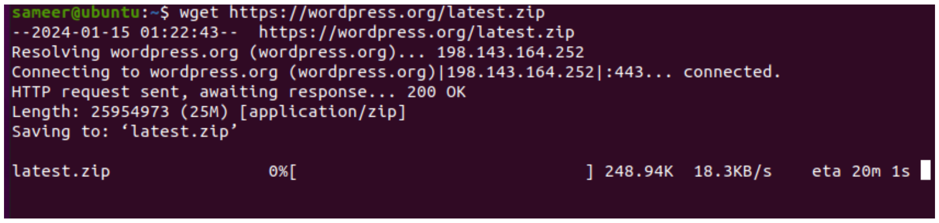
```
sameer@ubuntu:~$ ping google.com
PING google.com (172.217.17.78) 56(84) bytes of data.
64 bytes from ams16s30-in-f78.1e100.net (172.217.17.78): icmp_seq=2 ttl=128 time=17.9 ms
64 bytes from ams16s30-in-f78.1e100.net (172.217.17.78): icmp_seq=3 ttl=128 time=18.9 ms
64 bytes from ams16s30-in-f78.1e100.net (172.217.17.78): icmp_seq=4 ttl=128 time=18.6 ms
64 bytes from ams16s30-in-f78.1e100.net (172.217.17.78): icmp_seq=7 ttl=128 time=17.9 ms
64 bytes from ams16s30-in-f78.1e100.net (172.217.17.78): icmp_seq=8 ttl=128 time=18.9 ms
64 bytes from ams16s30-in-f78.1e100.net (172.217.17.78): icmp_seq=10 ttl=128 time=18.9 ms
```

b. wget command

The Linux command line lets you download files from the internet using the wget command. It works in the background without hindering other running processes.
The wget command retrieves files using HTTP, HTTPS, and FTP protocols. It can perform recursive downloads, which transfer website parts by following directory structures and links, creating local versions of the web pages.
To use it, enter the following command:

wget [option] [url]

For example, enter the following command to download the latest version of WordPress:

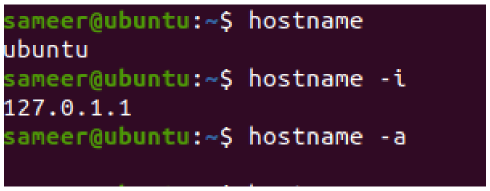

```
sameer@ubuntu:~$ wget https://wordpress.org/latest.zip
--2024-01-15 01:22:43--  https://wordpress.org/latest.zip
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 25954973 (25M) [application/zip]
Saving to: 'latest.zip'

latest.zip                0%[                             ] 248.94K  18.3KB/s    eta 20m 1s
```

c. hostname command

Run the hostname command to know the system's hostname. You can execute it with or without an option. Here's the general syntax:

hostname [option]

There are many optional flags to use, including:
-a or **–alias** displays the hostname's alias.
-A or **–all-fqdns** displays the machine's Fully Qualified Domain Name (FQDN).
-i or **–ip-address** displays the machine's IP address.
For example, enter the following command to know your computer's IP address:


```
sameer@ubuntu:~$ hostname
ubuntu
sameer@ubuntu:~$ hostname -i
127.0.1.1
sameer@ubuntu:~$ hostname -a
```

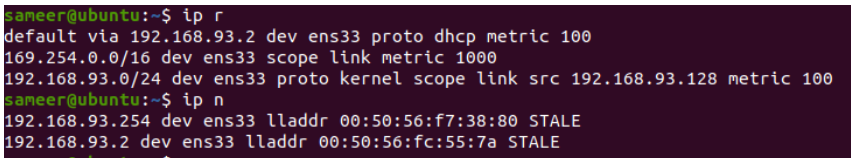
d. ip command

The ip command is a Linux net-tool for system and network administrators. IP stands for Internet Protocol and as the name suggests, the tool is used for configuring network interfaces.
Older Linux distributions used the ifconfig command, which operates similarly. However, ifconfig has a limited range of capabilities compared to the ip command.
To Use the ip command enter:

ip [OPTION] OBJECT {COMMAND | help}

OBJECTS (or subcommands) that you will use most often include:

1. link (l) – used to display and modify network interfaces.
2. address (addr/a) – used to display and modify protocol addresses (IP, IPv6).
3. route (r) – used to display and alter the routing table.
4. neigh (n) – used to display and manipulate neighbor objects (ARP table).


```
sameer@ubuntu:~$ ip r
default via 192.168.93.2 dev ens33 proto dhcp metric 100
169.254.0.0/16 dev ens33 scope link metric 1000
192.168.93.0/24 dev ens33 proto kernel scope link src 192.168.93.128 metric 100
sameer@ubuntu:~$ ip n
192.168.93.254 dev ens33 lladdr 00:50:56:f7:38:80 STALE
192.168.93.2 dev ens33 lladdr 00:50:56:fc:55:7a STALE
```

5. MANAGING FILES & DIRECTORIES

a. pwd command

Use the pwd command to find the path of your current working directory. Simply entering pwd will return the full current path – a path of all the directories that starts with a forward slash (/). For example, /home/username.

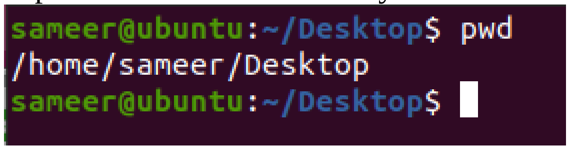
The pwd command uses the following syntax:

pwd [option]

It has two acceptable options:

-L or **–logical** prints environment variable content, including symbolic links.

-P or **–physical** prints the actual path of the current directory.

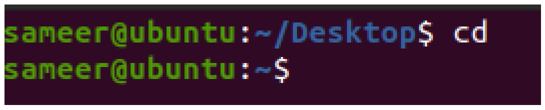


```
sameer@ubuntu:~/Desktop$ pwd
/home/sameer/Desktop
sameer@ubuntu:~/Desktop$
```

b. cd command

To navigate through the Linux files and directories, use the cd command. Depending on your current working directory, it requires either the full path or the directory name.

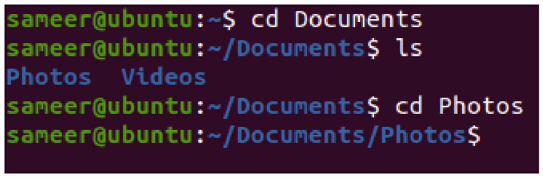
Running this command without an option will take you to the home folder. Keep in mind that only users with sudo privileges can execute it.



```
sameer@ubuntu:~/Desktop$ cd
sameer@ubuntu:~$
```

Let’s say you’re in /home/username/Documents and want to go to Photos, a subdirectory of Documents. To do so, enter the following command:

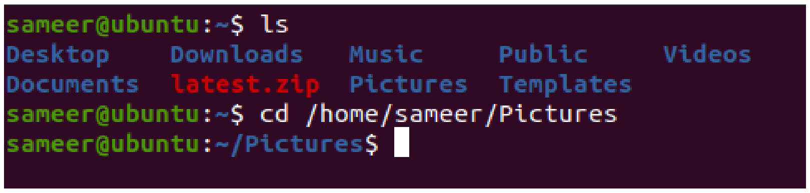
cd Photos



```
sameer@ubuntu:~$ cd Documents
sameer@ubuntu:~/Documents$ ls
Photos  Videos
sameer@ubuntu:~/Documents$ cd Photos
sameer@ubuntu:~/Documents/Photos$
```

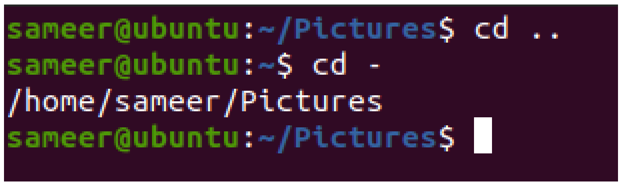
If you want to switch to a completely new directory, for example, **/home/username/Pictures**, you have to enter cd followed by the directory’s absolute path:

cd /home/username/Pictures



```
sameer@ubuntu:~$ ls
Desktop     Downloads   Music     Public     Videos
Documents   latest.zip  Pictures  Templates
sameer@ubuntu:~$ cd /home/sameer/Pictures
sameer@ubuntu:~/Pictures$ 
```

Here are some shortcuts to help you navigate:
cd ~[username] goes to another user's home directory.
cd .. moves one directory up**.**
cd- moves to your previous directory**.**

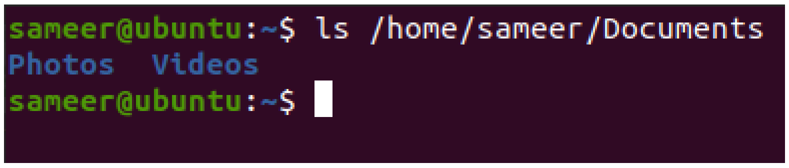


```
sameer@ubuntu:~/Pictures$ cd ..
sameer@ubuntu:~$ cd -
/home/sameer/Pictures
sameer@ubuntu:~/Pictures$ 
```

c. ls command

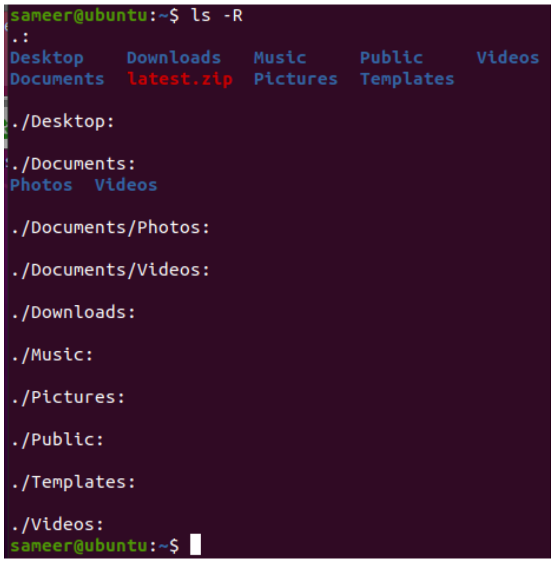
The ls command lists files and directories within a system. Running it without a flag or parameter will show the current working directory's content.
To see other directories' content, type ls followed by the desired path. For example, to view files in the Documents folder, enter:

ls /home/username/Documents

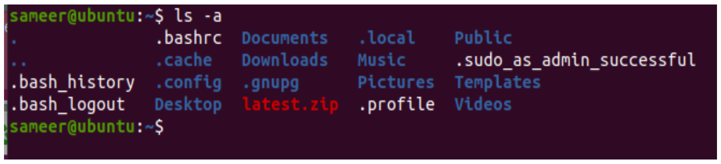


```
sameer@ubuntu:~$ ls /home/sameer/Documents
Photos  Videos
sameer@ubuntu:~$ 
```

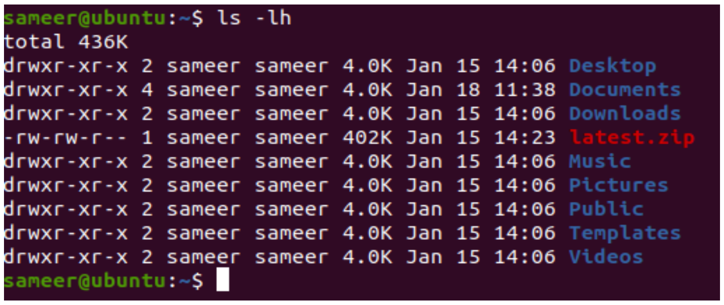
Here are some options you can use with the ls command:
ls -R lists all the files in the subdirectories.
ls -a shows hidden files in addition to the visible ones.
ls -lh shows the file sizes in easily readable formats, such as MB, GB, and TB.



```
sameer@ubuntu:~$ ls -R
.:
Desktop    Downloads   Music     Public     Videos
Documents  latest.zip  Pictures  Templates

./Desktop:

./Documents:
Photos  Videos

./Documents/Photos:

./Documents/Videos:

./Downloads:

./Music:

./Pictures:

./Public:

./Templates:

./Videos:
sameer@ubuntu:~$ 
```



```
sameer@ubuntu:~$ ls -a
.              .bashrc  Documents   .local    Public
..             .cache   Downloads   Music     .sudo_as_admin_successful
.bash_history  .config  .gnupg      Pictures  Templates
.bash_logout   Desktop  latest.zip  .profile  Videos
sameer@ubuntu:~$
```



```
sameer@ubuntu:~$ ls -lh
total 436K
drwxr-xr-x 2 sameer sameer 4.0K Jan 15 14:06 Desktop
drwxr-xr-x 4 sameer sameer 4.0K Jan 18 11:38 Documents
drwxr-xr-x 2 sameer sameer 4.0K Jan 15 14:06 Downloads
-rw-rw-r-- 1 sameer sameer 402K Jan 15 14:23 latest.zip
drwxr-xr-x 2 sameer sameer 4.0K Jan 15 14:06 Music
drwxr-xr-x 2 sameer sameer 4.0K Jan 15 14:06 Pictures
drwxr-xr-x 2 sameer sameer 4.0K Jan 15 14:06 Public
drwxr-xr-x 2 sameer sameer 4.0K Jan 15 14:06 Templates
drwxr-xr-x 2 sameer sameer 4.0K Jan 15 14:06 Videos
sameer@ubuntu:~$ 
```

d. mkdir command

Use the mkdir command to create one or multiple directories at once and set permissions for each of them. The user executing this command must have the privilege to make a new folder in the parent directory, or they may receive a permission denied error.
Here's the basic syntax:

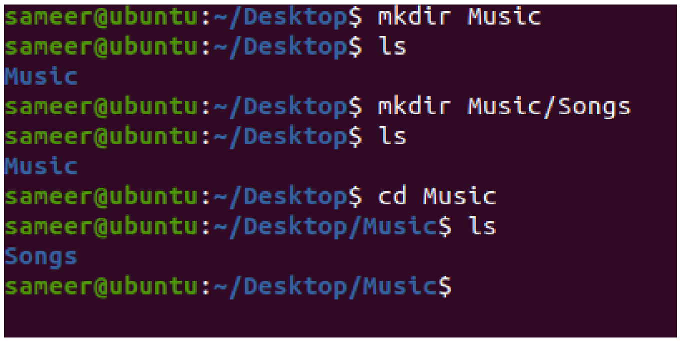
mkdir [option] directory_name

For example, you want to create a directory called Music:

mkdir Music

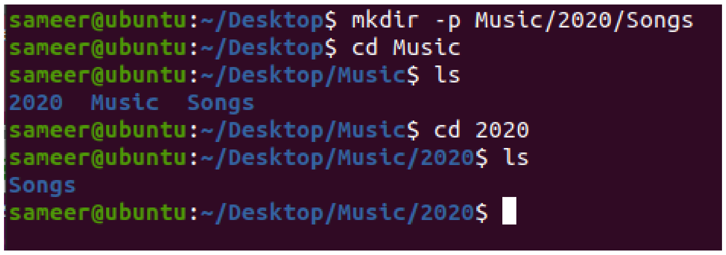
To make a new directory called Songs inside Music, use this command:

mkdir Music/Songs



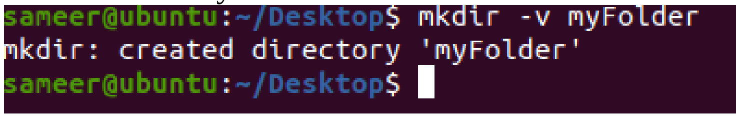
```
sameer@ubuntu:~/Desktop$ mkdir Music
sameer@ubuntu:~/Desktop$ ls
Music
sameer@ubuntu:~/Desktop$ mkdir Music/Songs
sameer@ubuntu:~/Desktop$ ls
Music
sameer@ubuntu:~/Desktop$ cd Music
sameer@ubuntu:~/Desktop/Music$ ls
Songs
sameer@ubuntu:~/Desktop/Music$
```

The mkdir command accepts many options, such as:
-p or **–parents** create a directory between two existing folders. For example, **mkdir -p Music/2020/Songs** will make the new "2020" directory.



```
sameer@ubuntu:~/Desktop$ mkdir -p Music/2020/Songs
sameer@ubuntu:~/Desktop$ cd Music
sameer@ubuntu:~/Desktop/Music$ ls
2020  Music  Songs
sameer@ubuntu:~/Desktop/Music$ cd 2020
sameer@ubuntu:~/Desktop/Music/2020$ ls
Songs
sameer@ubuntu:~/Desktop/Music/2020$
```

-m sets the file permissions. For instance, to create a directory with full read, write, and execute permissions for all users, enter mkdir -m777 directory_name.
-v prints a message for each created directory.

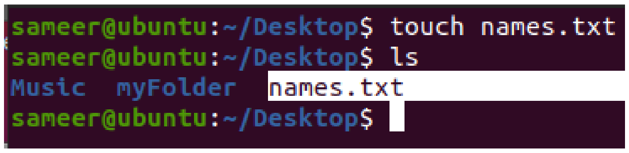


```
sameer@ubuntu:~/Desktop$ mkdir -v myFolder
mkdir: created directory 'myFolder'
sameer@ubuntu:~/Desktop$
```

e. touch command

It is used to create a file without any content. The file created using the touch command is empty. This command can be used when the user doesn't have data to store at the time of file creation.
Here is the basic syntax:

touch [filename.extension]

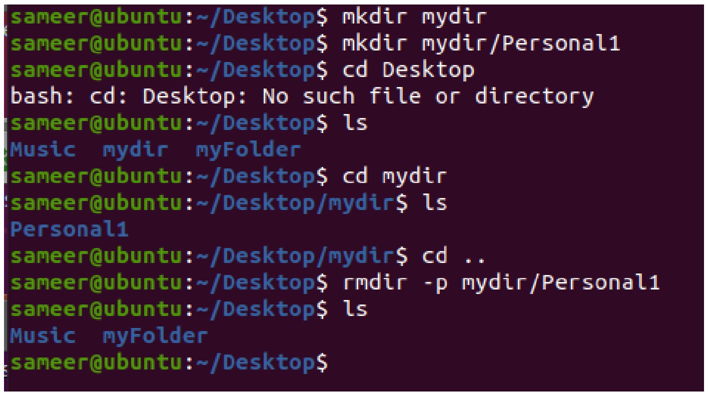


```
sameer@ubuntu:~/Desktop$ touch names.txt
sameer@ubuntu:~/Desktop$ ls
Music  myFolder  names.txt
sameer@ubuntu:~/Desktop$
```

f. rmdir command

To permanently delete an empty directory, use the rmdir command. Remember that the user running this command should have sudo privileges in the parent directory.
For example, you want to remove an empty subdirectory named personal1 and its main folder mydir:

rmdir -p mydir/personal1



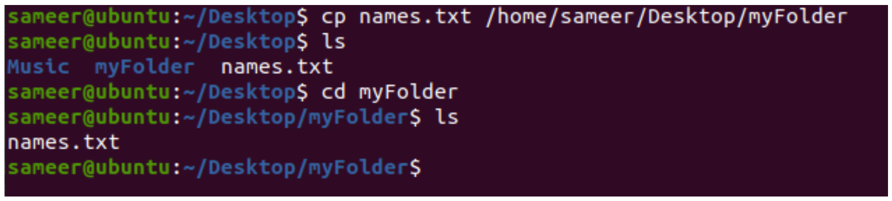
```
sameer@ubuntu:~/Desktop$ mkdir mydir
sameer@ubuntu:~/Desktop$ mkdir mydir/Personal1
sameer@ubuntu:~/Desktop$ cd Desktop
bash: cd: Desktop: No such file or directory
sameer@ubuntu:~/Desktop$ ls
Music  mydir  myFolder
sameer@ubuntu:~/Desktop$ cd mydir
sameer@ubuntu:~/Desktop/mydir$ ls
Personal1
sameer@ubuntu:~/Desktop/mydir$ cd ..
sameer@ubuntu:~/Desktop$ rmdir -p mydir/Personal1
sameer@ubuntu:~/Desktop$ ls
Music  myFolder
sameer@ubuntu:~/Desktop$
```

g. cp command

Use the cp command to copy files or directories and their content. Take a look at the following use cases.

To copy one file from the current directory to another, enter cp followed by the file name and the destination directory. For example:

cp filename.txt /home/username/Documents



```
sameer@ubuntu:~/Desktop$ cp names.txt /home/sameer/Desktop/myFolder
sameer@ubuntu:~/Desktop$ ls
Music  myFolder  names.txt
sameer@ubuntu:~/Desktop$ cd myFolder
sameer@ubuntu:~/Desktop/myFolder$ ls
names.txt
sameer@ubuntu:~/Desktop/myFolder$
```

To copy files to a directory, enter the file names followed by the destination directory:

cp filename1.txt filename2.txt filename3.txt /home/username/Documents

To copy the content of a file to a new file in the same directory, enter cp followed by the source file and the destination file:

cp filename1.txt filename2.txt

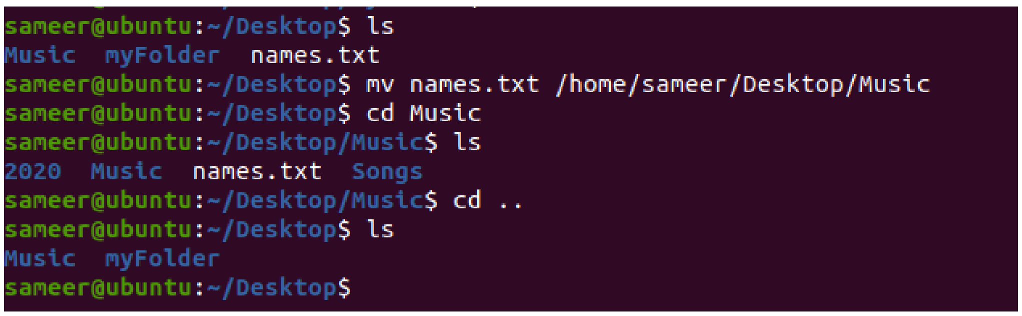
To copy an entire directory, pass the -R flag before typing the source directory, followed by the destination directory:

cp -R /home/username/Documents /home/username/Documents_backup

h. mv command

The primary use of the mv command is to move and rename files and directories. Additionally, it doesn't produce an output upon execution.
Simply type mv followed by the filename and the destination directory. For example, you want to move filename.txt to the /home/username/Documents directory:

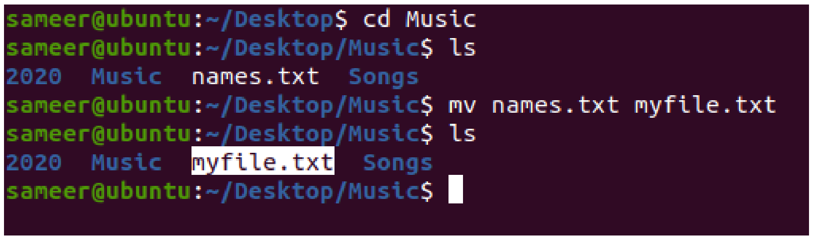
mv filename.txt /home/username/Documents



```
sameer@ubuntu:~/Desktop$ ls
Music  myFolder  names.txt
sameer@ubuntu:~/Desktop$ mv names.txt /home/sameer/Desktop/Music
sameer@ubuntu:~/Desktop$ cd Music
sameer@ubuntu:~/Desktop/Music$ ls
2020  Music  names.txt  Songs
sameer@ubuntu:~/Desktop/Music$ cd ..
sameer@ubuntu:~/Desktop$ ls
Music  myFolder
sameer@ubuntu:~/Desktop$
```

You can also use the mv command to rename a file:

mv old_filename.txt new_filename.txt



```
sameer@ubuntu:~/Desktop$ cd Music
sameer@ubuntu:~/Desktop/Music$ ls
2020  Music  names.txt  Songs
sameer@ubuntu:~/Desktop/Music$ mv names.txt myfile.txt
sameer@ubuntu:~/Desktop/Music$ ls
2020  Music  myfile.txt  Songs
sameer@ubuntu:~/Desktop/Music$
```

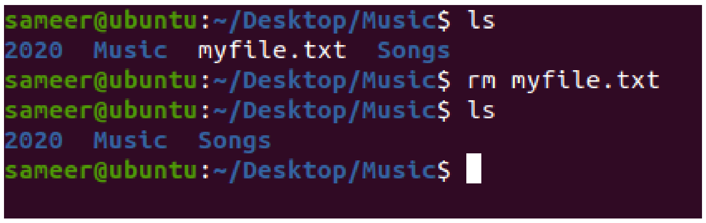
i. rm command

The rm command is used to delete files within a directory. Make sure that the user performing this command has write permissions.

Remember the directory's location as this will remove the file(s) and you can't undo it.

Here's the general syntax:

rm filename



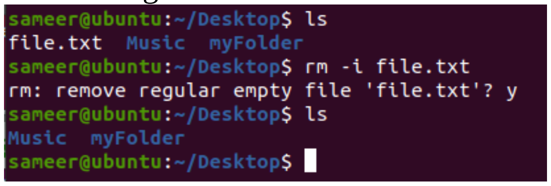
```
sameer@ubuntu:~/Desktop/Music$ ls
2020  Music  myfile.txt  Songs
sameer@ubuntu:~/Desktop/Music$ rm myfile.txt
sameer@ubuntu:~/Desktop/Music$ ls
2020  Music  Songs
sameer@ubuntu:~/Desktop/Music$
```

To remove multiple files, enter the following command:

rm filename1 filename2 filename3

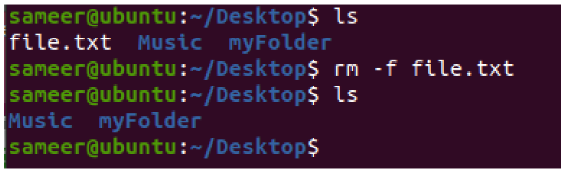
Here are some acceptable options you can add:

-i prompts system confirmation before deleting a file.



```
sameer@ubuntu:~/Desktop$ ls
file.txt  Music  myFolder
sameer@ubuntu:~/Desktop$ rm -i file.txt
rm: remove regular empty file 'file.txt'? y
sameer@ubuntu:~/Desktop$ ls
Music  myFolder
sameer@ubuntu:~/Desktop$
```

-f allows the system to remove without a confirmation.



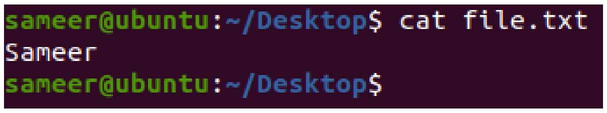
```
sameer@ubuntu:~/Desktop$ ls
file.txt  Music  myFolder
sameer@ubuntu:~/Desktop$ rm -f file.txt
sameer@ubuntu:~/Desktop$ ls
Music  myFolder
sameer@ubuntu:~/Desktop$
```

-r deletes files and directories recursively.

6. FILE EDITOR AND CREATION COMMANDS

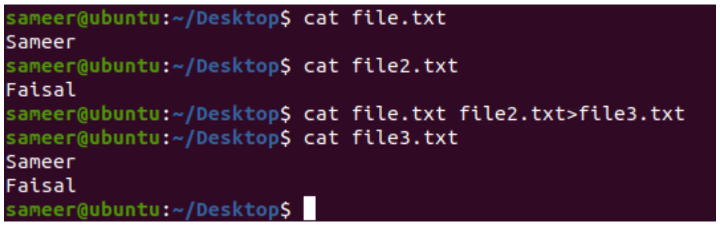
1. cat command

Concatenate, or cat, is one of the most frequently used Linux commands. It lists, combines, and writes file content to the standard output. To run the cat command, type cat followed by the file name and its extension. For instance: **cat filename.txt.**



```
sameer@ubuntu:~/Desktop$ cat file.txt
Sameer
sameer@ubuntu:~/Desktop$
```

Here are other ways to use the cat command:
cat filename1.txt filename2.txt > filename3.txt merges filename1.txt and filename2.txt and stores the output in filename3.txt.



```
sameer@ubuntu:~/Desktop$ cat file.txt
Sameer
sameer@ubuntu:~/Desktop$ cat file2.txt
Faisal
sameer@ubuntu:~/Desktop$ cat file.txt file2.txt>file3.txt
sameer@ubuntu:~/Desktop$ cat file3.txt
Sameer
Faisal
sameer@ubuntu:~/Desktop$
```

2. nano, vi, jed, gedit command

Linux allows users to edit and manage files via a text editor, such as nano, vi, or jed. nano and vi come with the operating system, while jed has to be installed.
The **nano** command denotes keywords and can work with most languages. To use it, enter the following command:

nano [filename]

vi uses two operating modes to work – insert and command. insert is used to edit and create a text file. On the other hand, the command performs operations, such as saving, opening, copying, and pasting a file.
To use **vi** on a file, enter:

vi [filename]

jed has a drop-down menu interface that allows users to perform actions without entering keyboard combinations or commands. Like vi, it has modes to load modules or plugins to write specific texts.
To open the program, simply enter **jed** to the command line.

gedit is an easy-to-use and general-purpose text editor just like notepad in windows.
To use **gedit**, enter:

gedit [filename]

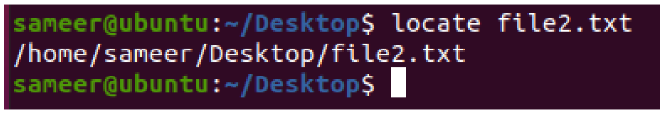
7. SEARCH COMMANDS

1. locate command

The locate command can find a file in the database system.
Moreover, adding the -i argument will turn off case sensitivity, so you can search for a file even if you don't remember its exact name.
To look for content that contains two or more words, use an asterisk (*). For example:

locate -i school*not

The command will search for files that contain the words school and note, whether they use uppercase or lowercase letters.



```
sameer@ubuntu:~/Desktop$ locate file2.txt
/home/sameer/Desktop/file2.txt
sameer@ubuntu:~/Desktop$ 
```

2. find command

Use the find command to search for files within a specific directory and perform subsequent operations.
Here's the general syntax:

find [option] [path] [expression]

For example, you want to look for a file called notes.txt within the home directory and its subfolders:

find /home -name notes.txt

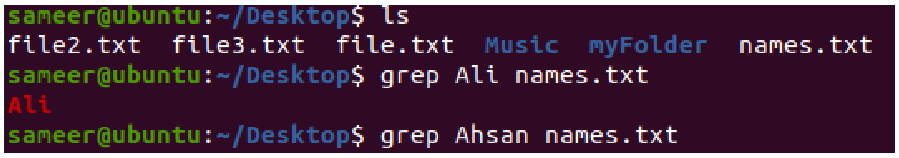
Here are other variations when using find:

find -name filename.txt to find files in the current directory.
./ -type d -name directoryname to look for directories.

3. grep command

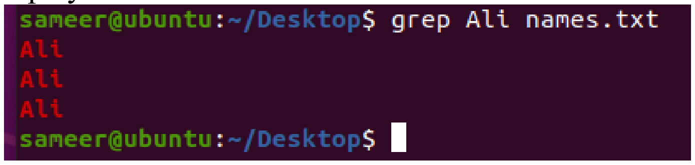
Another basic Linux command on the list is grep or global regular expression print. It lets you find a word by searching through all the texts in a specific file.
Once the grep command finds a match, it prints all lines that contain the specific pattern. This command helps filter through large log files.
For example, you want to search for the name "Ali" in the names.txt file:

grep Ali notepad.txt



```
sameer@ubuntu:~/Desktop$ ls
file2.txt  file3.txt  file.txt  Music  myFolder  names.txt
sameer@ubuntu:~/Desktop$ grep Ali names.txt
Ali
sameer@ubuntu:~/Desktop$ grep Ahsan names.txt
```

The command's output will display lines that contain Ali.



```
sameer@ubuntu:~/Desktop$ grep Ali names.txt
Ali
Ali
Ali
sameer@ubuntu:~/Desktop$
```

8. DISPLAYING OUTPUT COMMANDS

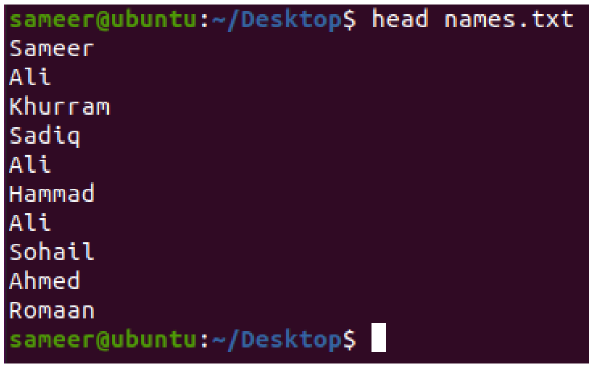
1. head command

The head command allows you to view the first ten lines of a text. Adding an option lets you change the number of lines shown. The head command is also used to output piped data to the CLI.
Here's the general syntax:

head [option] [file]

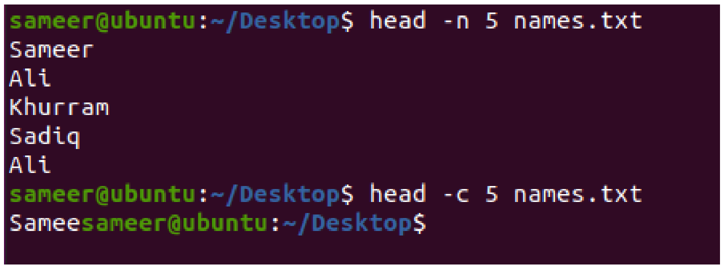
For instance, you want to view the first ten lines of names.txt, located in the current directory:

head names.txt



```
sameer@ubuntu:~/Desktop$ head names.txt
Sameer
Ali
Khurram
Sadiq
Ali
Hammad
Ali
Sohail
Ahmed
Romaan
sameer@ubuntu:~/Desktop$
```

Below are some options you can add:
-n or –lines prints the first customized number of lines. For example, enter **head -n 5 filename.txt** to show the first five lines of filename.txt.
-c or –bytes prints the first customized number of bytes of each file.
-q or –quiet will not print headers specifying the file name.

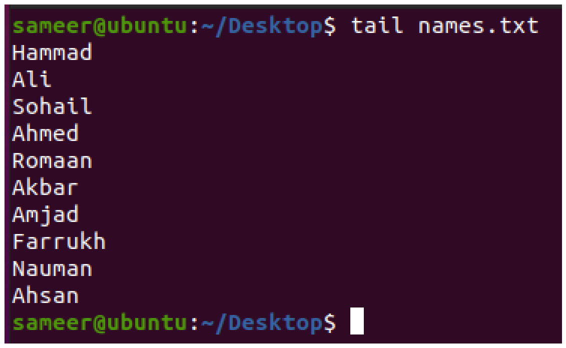


```
sameer@ubuntu:~/Desktop$ head -n 5 names.txt
Sameer
Ali
Khurram
Sadiq
Ali
sameer@ubuntu:~/Desktop$ head -c 5 names.txt
Sameesameer@ubuntu:~/Desktop$
```

2. tail command

The tail command displays the last ten lines of a file. It allows users to check whether a file has new data or to read error messages.
Here’s the general format:

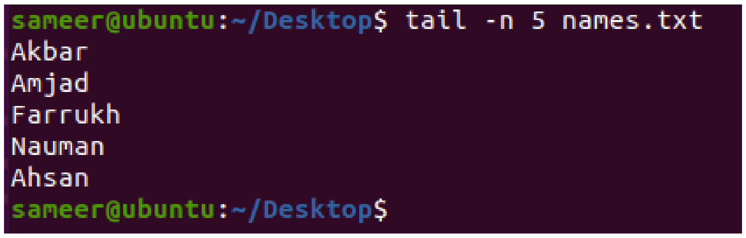
tail [option] [file]



```
sameer@ubuntu:~/Desktop$ tail names.txt
Hammad
Ali
Sohail
Ahmed
Romaan
Akbar
Amjad
Farrukh
Nauman
Ahsan
sameer@ubuntu:~/Desktop$ 
```

For example, you want to show the last 5 lines of the names.txt file:

tail –n 5 colors.txt



```
sameer@ubuntu:~/Desktop$ tail -n 5 names.txt
Akbar
Amjad
Farrukh
Nauman
Ahsan
sameer@ubuntu:~/Desktop$
```

9. FILE DIRECTORY PERMISSIONS & OWNERSHIP COMMANDS

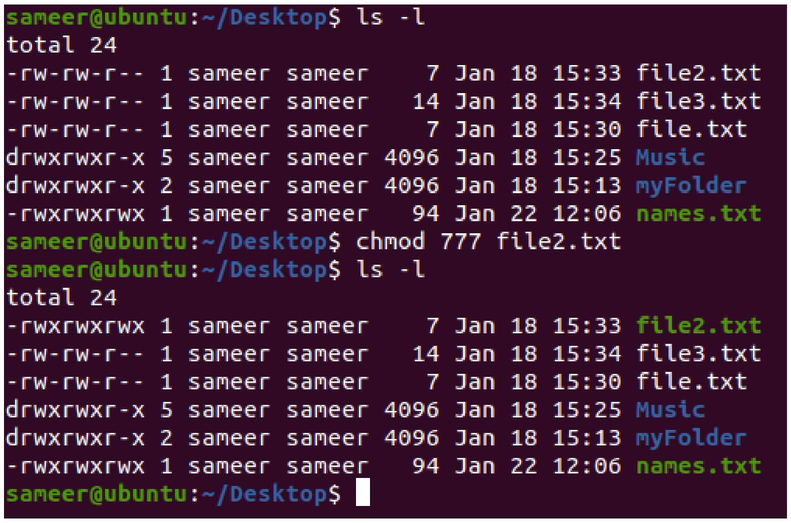
1. chmod command

chmod is a common command that modifies a file or directory’s read, write, and execute permissions.
In Linux, each file is associated with three user classes – owner, group member, and others.
Here’s the basic syntax:

chmod [option] [permission] [file_name]

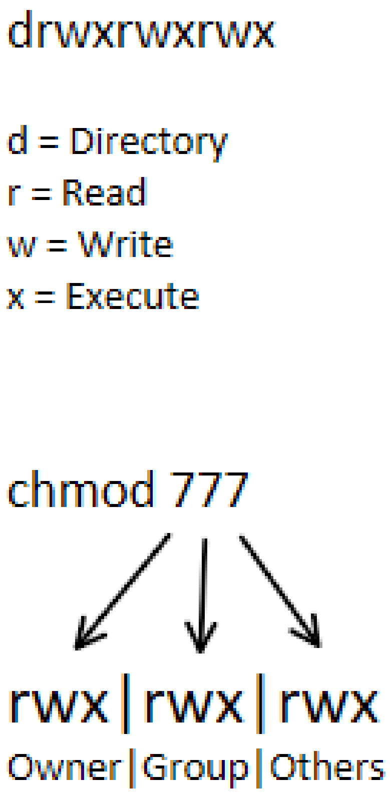
For example, the owner is currently the only one with full permissions to change names.txt. To allow group members and others to read, write, and execute the file, change it to the -rwxrwxrwx permission type, whose numeric value is 777:

chmod 777 names.txt



```
sameer@ubuntu:~/Desktop$ ls -l
total 24
-rw-rw-r-- 1 sameer sameer    7 Jan 18 15:33 file2.txt
-rw-rw-r-- 1 sameer sameer   14 Jan 18 15:34 file3.txt
-rw-rw-r-- 1 sameer sameer    7 Jan 18 15:30 file.txt
drwxrwxr-x 5 sameer sameer 4096 Jan 18 15:25 Music
drwxrwxr-x 2 sameer sameer 4096 Jan 18 15:13 myFolder
-rwxrwxrwx 1 sameer sameer   94 Jan 22 12:06 names.txt
sameer@ubuntu:~/Desktop$ chmod 777 file2.txt
sameer@ubuntu:~/Desktop$ ls -l
total 24
-rwxrwxrwx 1 sameer sameer    7 Jan 18 15:33 file2.txt
-rw-rw-r-- 1 sameer sameer   14 Jan 18 15:34 file3.txt
-rw-rw-r-- 1 sameer sameer    7 Jan 18 15:30 file.txt
drwxrwxr-x 5 sameer sameer 4096 Jan 18 15:25 Music
drwxrwxr-x 2 sameer sameer 4096 Jan 18 15:13 myFolder
-rwxrwxrwx 1 sameer sameer   94 Jan 22 12:06 names.txt
sameer@ubuntu:~/Desktop$ 
```

This command supports many options, including:
-c or –**changes** displays information when a change is made.
-f or –**silent** suppresses the error messages.
-v or –**verbose** displays a diagnostic for each processed file.



7	rwx	111
6	rw-	110
5	r-x	101
4	r--	100
3	-wx	011
2	-w-	010
1	--x	001
0	---	000

2. chown command

The chown command lets you change the ownership of a file, directory, or symbolic link to a specified username.
Here's the basic format:

chown [option] owner[:group] file(s)

For example, you want to make linuxuser2 the owner of filename.txt:

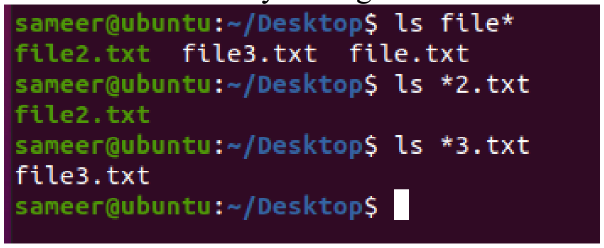
chown linuxuser2 filename.txt

10. WILDCARDS

1. ***** - will match against none or one or a string of more than a character
2. **?** - can be used to match one character
3. **[]** - matches one specified character out of a group of characters

1. *

- **'ls file*'** - list all the files in current directory starting with filename 'file'.
- **'ls *2.txt'** - list all the files in current directory ending with '2.txt'



```
sameer@ubuntu:~/Desktop$ ls file*
file2.txt  file3.txt  file.txt
sameer@ubuntu:~/Desktop$ ls *2.txt
file2.txt
sameer@ubuntu:~/Desktop$ ls *3.txt
file3.txt
sameer@ubuntu:~/Desktop$
```

2. ?

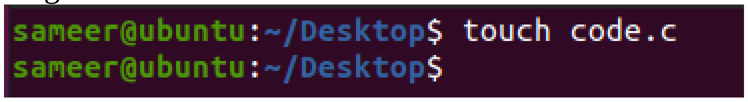
- **'ls file.tx?'** - list all the files that begins with 'file.tx'

3. [ ]

- **'ls file[12345]'** - list all the file that begins with 'file' and has a 1,2,3,4 or 5 after it.

11. C – CODE CREATION AND COMPILATION

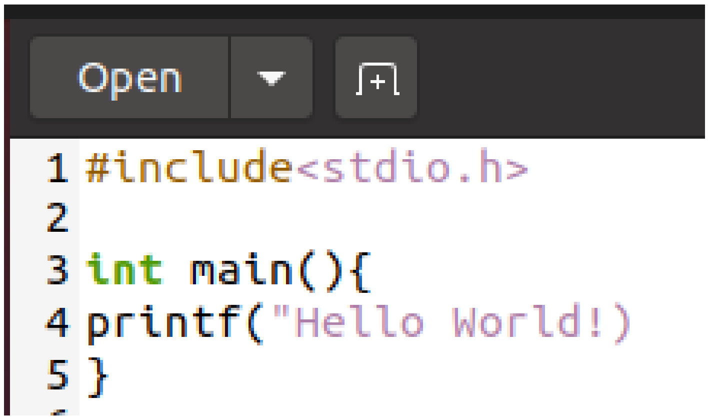
You can create C files with the same method as any other file creation.
To get started use the following command:



```
sameer@ubuntu:~/Desktop$ touch code.c
sameer@ubuntu:~/Desktop$
```

You can use any text edit of your choice to write code in this file (gedit, vi, nano, etc)

Now write some C-Code in the file as an example I am writing a simple Hello World line in it.



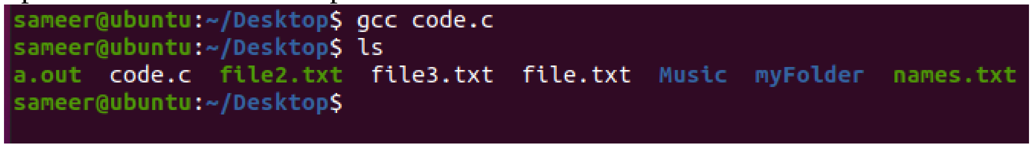
```
#include<stdio.h>

int main(){
printf("Hello World!)
}
```

Compilation

Normally in a OS environment you would have an editor that would do the compilation of the code automatically but since we are using the terminal mainly, we would have to compile it ourselves. There are two popular ways to compile the above file.

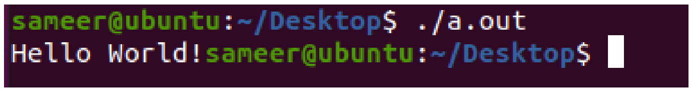
1. gcc filename.c

You need to install gcc first by using the following command: **sudo apt install gcc**
GCC is the GNU compiler collection which contains multiple compilers ranging from C to C++, Objective -C, Fortran, Ada, Go and D as well as the libraries for these programming languages. The above command will read the code and compile the code into an output file know as a.out.



```
sameer@ubuntu:~/Desktop$ gcc code.c
sameer@ubuntu:~/Desktop$ ls
a.out  code.c  file2.txt  file3.txt  file.txt  Music  myFolder  names.txt
sameer@ubuntu:~/Desktop$
```

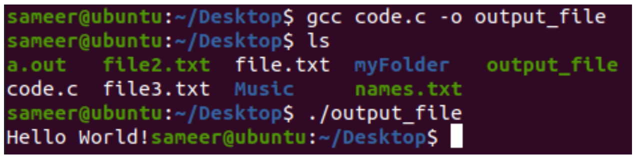
The a.out is the output file which contains the output of the C-code that is compiled. Note if your code has I/O dependencies such as taking input from the user or writing to the file, It will not be completed until the output file is executed.
To run the output file use the following command



```
sameer@ubuntu:~/Desktop$ ./a.out
Hello World!sameer@ubuntu:~/Desktop$ 
```

This will output the contents of the executable to the terminal.
But having multiple C/C++ files with the same a.out will be a problem as the OS will keep overwriting the contents of the output file to the new file that it just compiled. To get around this we can rename this output file via the following command.

gcc code.c -o output_file



```
sameer@ubuntu:~/Desktop$ gcc code.c -o output_file
sameer@ubuntu:~/Desktop$ ls
a.out   file2.txt  file.txt  myFolder   output_file
code.c  file3.txt  Music     names.txt
sameer@ubuntu:~/Desktop$ ./output_file
Hello World!sameer@ubuntu:~/Desktop$ 
```

Here you can see we compiled the output file to a new file which we named as output_file. Notice it has no extension (like we say .out with the old output file). That’s because the operating system interprets it as an executable and it does not require an extension to be mentioned with the file. In GUI based Linux distribution this would be labelled with a green text to signify that this file is an executable file. The -o is a flag to forward the output to the file we want to create.

Tasks

1. Create the following directories with one command. dirOSLAB -> subDir -> subsubdir -> OSLAB1.

2. Write 2 C program one prints “I love Operating System” and other prints “I love Linux”. Compile and Run both programs and print the output to two different files. After then combine both the files in one new file using a single command.

3. Perform the following activity:
   a. Create three file with starting text being test
   b. In the first file write the text first file.
   c. In the second file take the text from the first file and concatenate the string second file with the text from the first file and add it to the second file.
   d. Do the above for the third file but the string would be third file while being concatenated from the text of the second file
   e. Find the string file in all the files that you created and print them to the console
   f. Create a folder called concatenated files.
   g. Copy all the contents of the previous three files and create a new file in the newly created folder and paste it there (Note all this has to be done via the terminal No use of built in Ubuntu GUI features)
   h. Delete the three files that you created earlier with a single command.

4. Perform the following (Note: Some files might not be present so you should create one):
   a. List the files in the directory "/bin" that end in "sh".
   b. On one line, use the "cd" command to first go to your home directory then to the "<rollnumber>" subdirectory. [Ans: cd /home; cd <rollnumber>]
   c. What command lists the files in the current directory that begin with upper case letters?
   d. If they do not already exist, create three new directories "Letters", "Programs", and "Misc" using a single command
   e. Copy all files in the current directory whose names contain the character string "let" into the subdirectory "Letters".
   f. Copy all files in the current directory whose names end in ".c" or ".h" into the subdirectory "Programs".
   g. Copy all files in the current directory whose names contain the character strings "notes" or "misc" into the subdirectory "Misc".
   h. Copy all files which begin with "copy.me" into the "OS" subdirectory. Move all files which begin with "move.me" into the "OS" subdirectory.
   i. Delete all files which contain the sequence "del