

# DS Assignment 1

Name: Himani Verma

Admission No: U19CS075

---

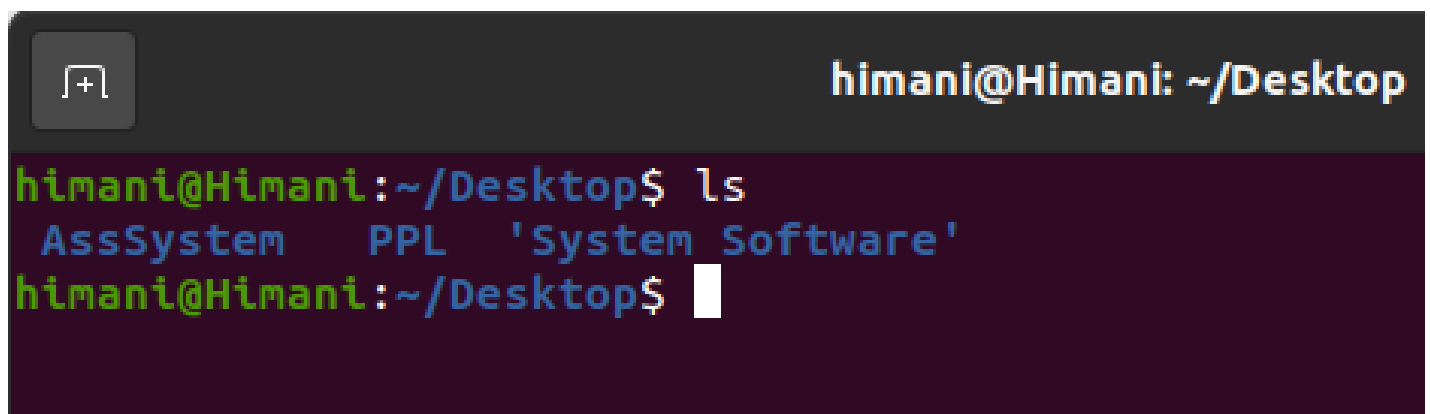
Execute all commands on Linux terminal and note down the functionality of each.

1. Basic Unix/Linux commands:

## ls

**Functionality:** **ls** is a Linux shell command that lists directory contents of files and directories.

**Execution:**

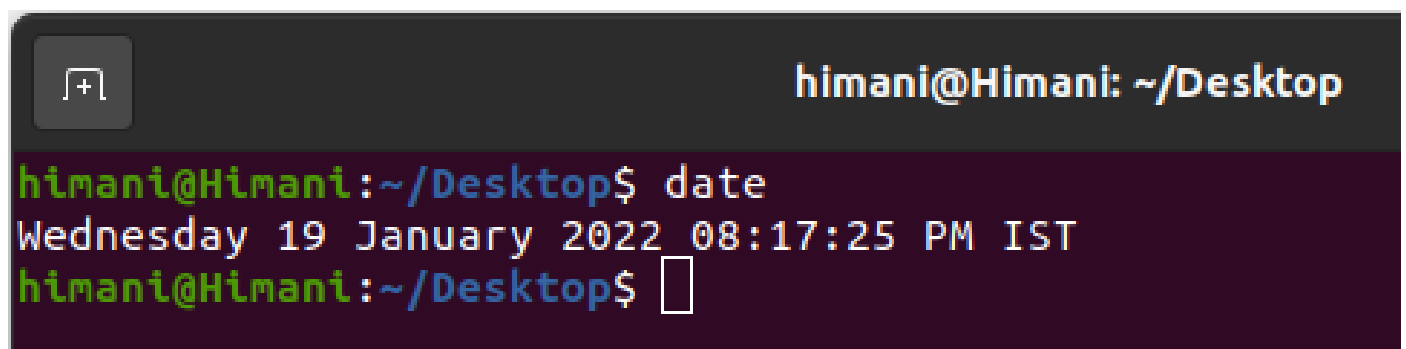
A terminal window with a dark background. The title bar shows a window icon and the text 'himani@Himani: ~/Desktop'. The prompt is 'himani@Himani:~/Desktop\$'. The command 'ls' has been entered, and the output is 'AssSystem PPL 'System Software''. The prompt is now 'himani@Himani:~/Desktop\$' with a cursor.

```
himani@Himani:~/Desktop$ ls
AssSystem  PPL  'System Software'
himani@Himani:~/Desktop$
```

## Date

**Functionality:** Used to display the system data and time. It is also used to set date and time of the system.

**Execution:**

A terminal window with a dark background. The title bar shows a window icon and the text 'himani@Himani: ~/Desktop'. The prompt is 'himani@Himani:~/Desktop\$'. The command 'date' has been entered, and the output is 'Wednesday 19 January 2022 08:17:25 PM IST'. The prompt is now 'himani@Himani:~/Desktop\$' with a cursor.

```
himani@Himani:~/Desktop$ date
Wednesday 19 January 2022 08:17:25 PM IST
himani@Himani:~/Desktop$
```

## Help

**Functionality:** **help** command as told before just displays information about shell built-in commands.

**Execution:**

```
himani@Himani:~/Desktop$ help
GNU bash, version 5.0.17(1)-release (x86_64-pc-linux-gnu)
These shell commands are defined internally.  Type 'help' to see this list.
Type 'help name' to find out more about the function 'name'.
Use 'info bash' to find out more about the shell in general.
Use 'man -k' or 'info' to find out more about commands not in this list.

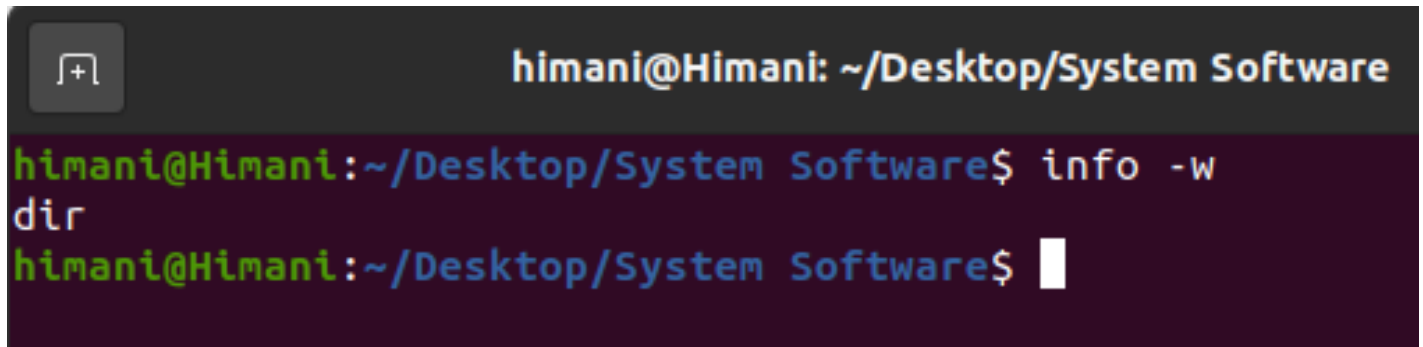
A star (*) next to a name means that the command is disabled.

job_spec [&]
(( expression ))
. filename [arguments]
:
[ arg... ]
[[ expression ]]
alias [-p] [name[=value] ... ]
bg [job_spec ...]
bind [-lpsvPSVX] [-m keymap] [-f file]
break [n]
builtin [shell-builtin [arg ...]]
caller [expr]
case WORD in [PATTERN [| PATTERN]...)>
cd [-L|[-P [-e]] [-@]] [dir]
command [-pVv] command [arg ...]
compgen [-abdefghjksuv] [-o option] [>
complete [-abdefghjksuv] [-pr] [-DEI]>
compopt [-o|+o option] [-DEI] [name .>
continue [n]
coproc [NAME] command [redirections]
declare [-aAfFgIlNrtux] [-p] [name[=v>
dirs [-clpv] [+N] [-N]
disown [-h] [-ar] [jobspec ... | pid >
echo [-neE] [arg ...]
enable [-a] [-dnps] [-f filename] [na>
eval [arg ...]
exec [-cl] [-a name] [command [argume>
exit [n]
export [-fn] [name[=value] ...] or ex>
false
fc [-e ename] [-lnr] [first] [last] o>
fg [job_spec]
for NAME in WORDS ... ] ; do COMMAND>
for (( exp1; exp2; exp3 )); do COMMAN>
function name { COMMANDS ; } or name >
getopts optstring name [arg]
hash [-lr] [-p pathname] [-dt] [name >
help [-dms] [pattern ...]
history [-c] [-d offset] [n] or hist>
if COMMANDS; then COMMANDS; [ elif C>
jobs [-lnprs] [jobspec ...] or jobs >
kill [-s sigspec | -n signum | -sigs>
let arg [arg ...]
local [option] name[=value] ...
logout [n]
mapfile [-d delim] [-n count] [-O or>
popd [-n] [+N | -N]
printf [-v var] format [arguments]
pushd [-n] [+N | -N | dir]
pwd [-LP]
read [-ers] [-a array] [-d delim] [->
readarray [-d delim] [-n count] [-O >
readonly [-aAf] [name[=value] ...] o>
return [n]
select NAME [in WORDS ... ;] do COMM>
set [-abefhkmnptuvxBCHP] [-o option->
shift [n]
shopt [-pqsu] [-o] [optname ...]
source filename [arguments]
suspend [-f]
test [expr]
time [-p] pipeline
times
trap [-lp] [[arg] signal_spec ...]
true
type [-afptP] name [name ...]
typeset [-aAfFgIlNrtux] [-p] name[=v>
ulimit [-SHabcdefiklmnpqrstuvxPT] [l>
umask [-p] [-S] [mode]
unalias [-a] name [name ...]
unset [-f] [-v] [-n] [name ...]
until COMMANDS; do COMMANDS; done
variables - Names and meanings of so>
wait [-fn] [id ...]
while COMMANDS; do COMMANDS; done
{ COMMANDS ; }
```

## Info

**Functionality:** **info** command reads documentation in the *info* format. It will give detailed information for a command when compared with the man page.

**Execution:**

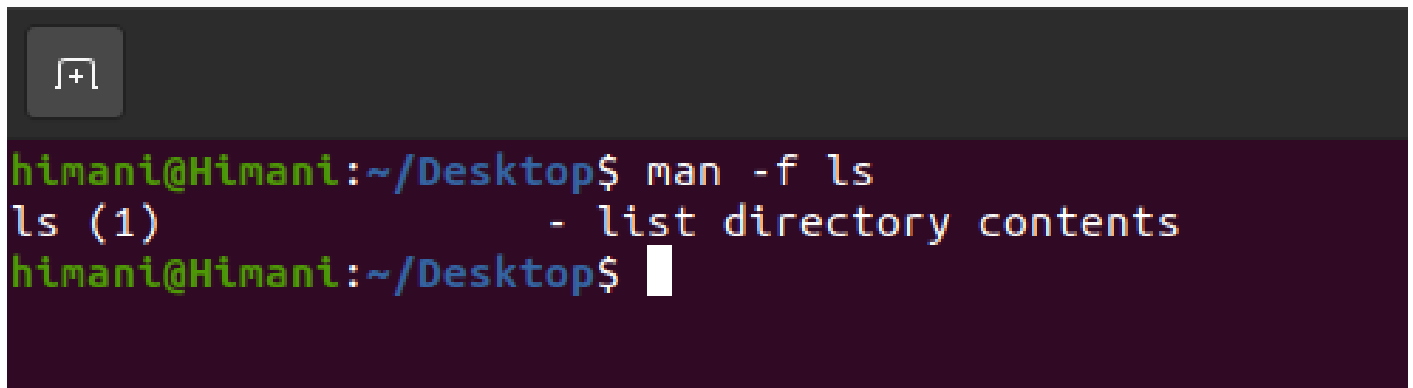
A terminal window with a dark background. The title bar shows a window icon and the text 'himani@Himani: ~/Desktop/System Software'. The terminal text shows the user 'himani@Himani' at the prompt '~/Desktop/System Software\$' typing 'info -w dir'. The prompt changes to 'himani@Himani:~/Desktop/System Software\$' with a cursor.

```
himani@Himani: ~/Desktop/System Software
himani@Himani:~/Desktop/System Software$ info -w
dir
himani@Himani:~/Desktop/System Software$
```

## Man

**Functionality:** **man** command in Linux is used to display the user manual of any command that we can run on the terminal. It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.

**Execution:**

A terminal window with a dark background. The title bar shows a window icon and the text 'himani@Himani: ~/Desktop/System Software'. The terminal text shows the user 'himani@Himani' at the prompt '~/Desktop\$' typing 'man -f ls'. The output shows 'ls (1)' followed by '- list directory contents'. The prompt changes to 'himani@Himani:~/Desktop\$' with a cursor.

```
himani@Himani:~/Desktop$ man -f ls
ls (1) - list directory contents
himani@Himani:~/Desktop$
```

## Who

**Functionality:** The **who** command is used to get information about currently logged in user on to system.

**Execution:**

```
himani@Himani: ~/Desktop/System Software  
himani@Himani:~/Desktop/System Software$ who  
himani      :0                2022-01-19 20:16 (:0)  
himani@Himani:~/Desktop/System Software$
```

## Pwd

**Functionality:** **pwd** stands for Print Working Directory. It prints the path of the working directory, starting from the root.

**Execution:**

```
himani@Himani: ~/Desktop/System Software  
himani@Himani:~/Desktop/System Software$ pwd  
/home/himani/Desktop/System Software  
himani@Himani:~/Desktop/System Software$
```

## Cat

**Functionality:** Cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, concatenate files.

**Execution:**

```
himani@Himani: ~/Desktop  
himani@Himani:~/Desktop$ cat b.txt  
Hello world!  
himani@Himani:~/Desktop$
```

## More

**Functionality:** **more** command is used to view the text files in the command prompt, displaying one screen at a time in case the file is large

**Execution:**

```
himani@Himani:~/Desktop$ more -d b.txt
Hello world!
Lorem ipsum dolor sit amet consectetur adipisicing elit. Maxime mollitia, molestiae quas vel sint commodi repudiandae consequuntur voluptatum laborum numquam blanditiis harum quisquam eius sed odit fugiat iusto fuga praesentium optio, eaque rerum! Provident similique accusantium nemo autem. Veritatis obcaecati tenetur iure eius earum ut molestias architecto voluptate aliquam nihil, eveniet aliquid culpa officia aut! Impedit sit sunt quaerat, odit, tenetur error, harum nesciunt ipsum debitis quas aliquid. Reprehenderit, quia. Quo neque error repudiandae fuga? Ipsa laudantium molestias eos sapiente officiis modi at sunt excepturi expedita sint? Sed quibusdam recusandae alias error harum maxime adipisci amet laborum. Perspiciatis minima nesciunt dolorem! Officiis iure rerum voluptates a cumque velit quibusdam sed amet tempora. Sit laborum ab, eius fugit doloribus tenetur fugiat, temporibus enim commodi iusto libero magni deleniti quod quam consequuntur! Commodi minima excepturi repudiandae velit hic maxime doloremque. Quaerat provident commodi consectetur veniam similique ad earum omnis ipsum saepe, voluptas, hic voluptates pariatur est explicabo fugiat, dolorum eligendi quam cupiditate excepturi mollitia maiores labore suscipit quas? Nulla, placeat. Voluptatem quaerat non architecto ab laudantium modi minima sunt esse temporibus sint culpa, recusandae aliquam numquam totam ratione voluptas quod exercitationem fuga. Possimus quis earum veniam quasi aliquam eligendi, placeat qui corporis!
himani@Himani:~/Desktop$
```

## Mv

**Functionality:** mv stands for move. mv is used to move one or more files or directories from one place to another in a file system like UNIX.

**Execution:**

If the destination file **doesn't exist**, it will be created. In the above command **mv** simply replaces the source filename in the directory with the destination filename(new name). If the destination file **exist**, then it will be **overwrite** and the source file will be deleted.

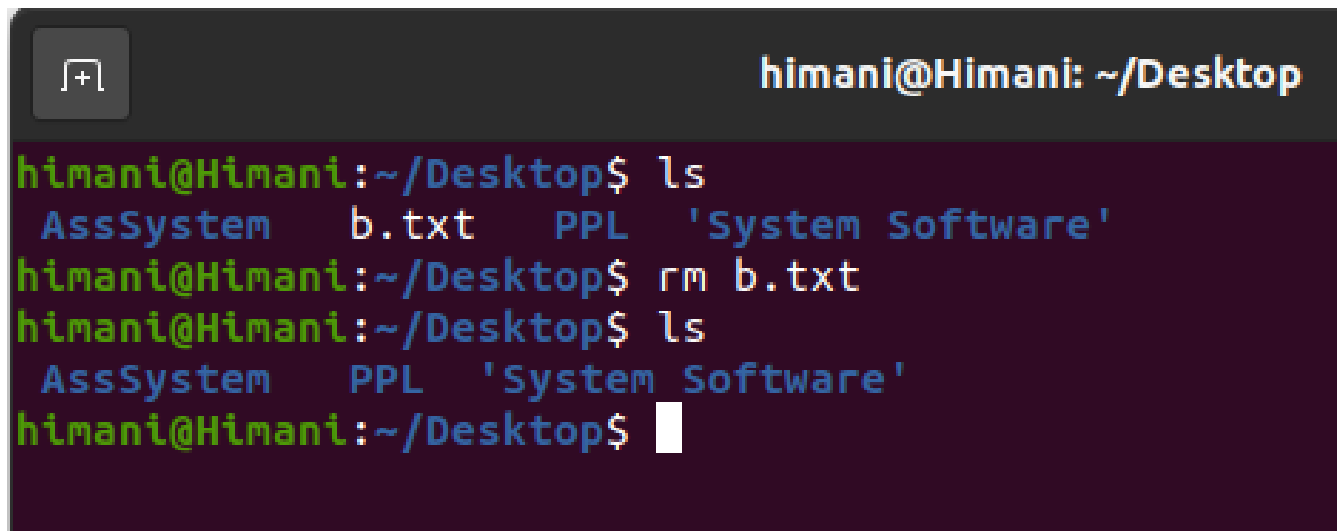
```
himani@Himani: ~/Desktop

himani@Himani:~/Desktop$ ls
AssSystem  a.txt    PPL     'System Software'
himani@Himani:~/Desktop$ mv a.txt b.txt
himani@Himani:~/Desktop$ ls
AssSystem  b.txt    PPL     'System Software'
himani@Himani:~/Desktop$
```

## Rm

**Functionality:** rm stands for **remove** here. rm command is used to remove objects such as files, directories, symbolic links and so on from the file system like UNIX. To be more precise, rm removes references to objects from the filesystem, where those objects might have had multiple references (for example, a file with two different names). **By default, it does not remove directories.**

**Execution:**

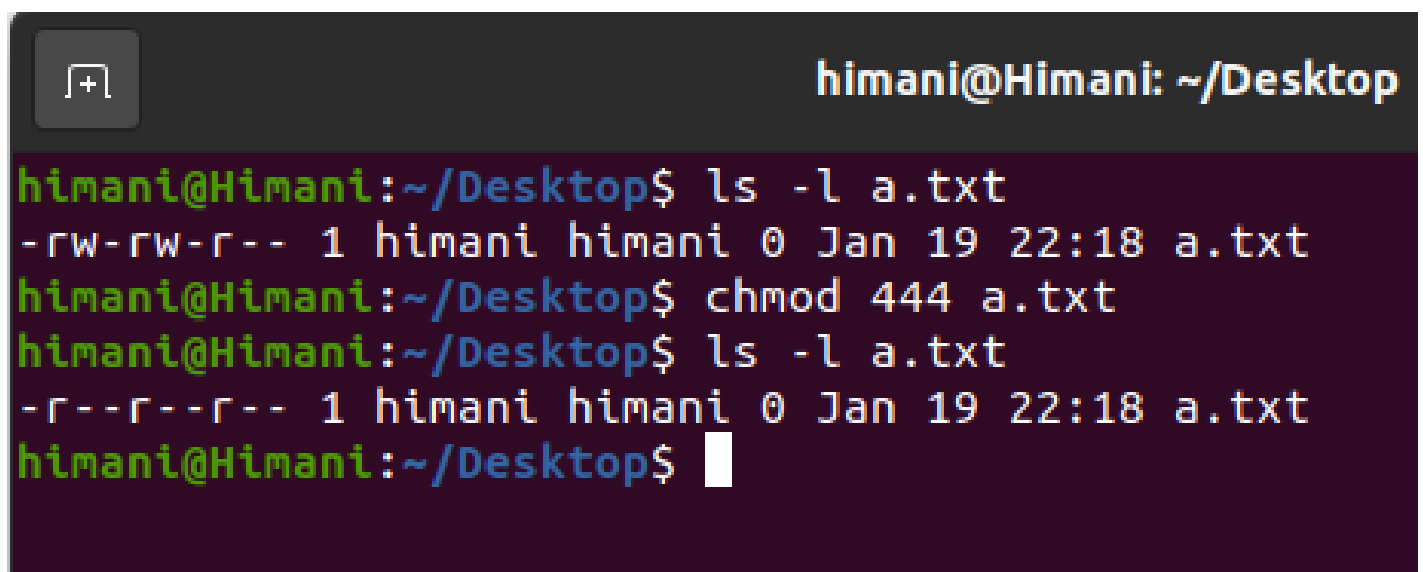
A terminal window with a dark background. The title bar shows a window icon and the text "himani@Himani: ~/Desktop". The terminal content shows the following commands and output:

```
himani@Himani:~/Desktop$ ls
AssSystem  b.txt  PPL  'System Software'
himani@Himani:~/Desktop$ rm b.txt
himani@Himani:~/Desktop$ ls
AssSystem  PPL  'System Software'
himani@Himani:~/Desktop$
```

## Chmod

**Functionality:** In Unix-like operating systems, the **chmod** command is used to change the access mode of a file.

**Execution:**

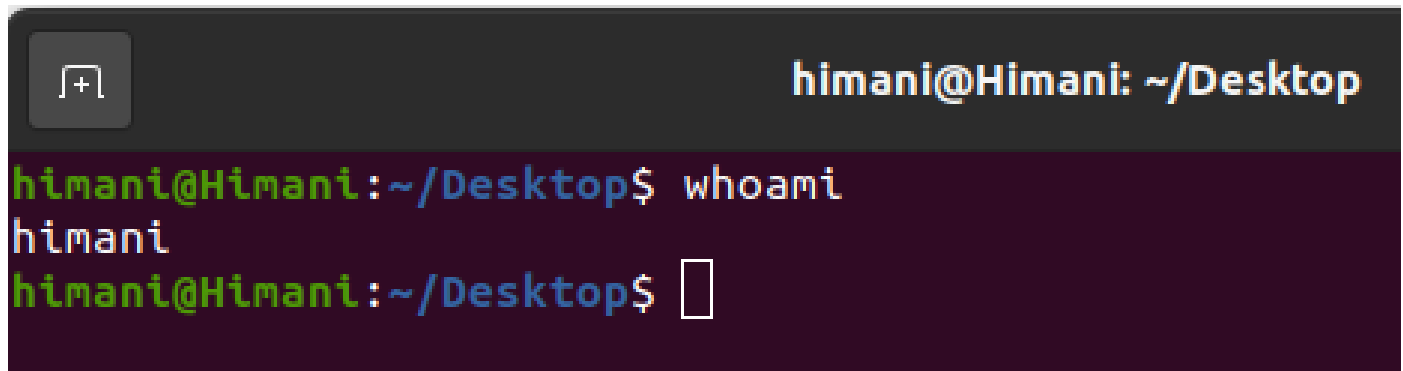
A terminal window with a dark background. The title bar shows a window icon and the text "himani@Himani: ~/Desktop". The terminal content shows the following commands and output:

```
himani@Himani:~/Desktop$ ls -l a.txt
-rw-rw-r-- 1 himani himani 0 Jan 19 22:18 a.txt
himani@Himani:~/Desktop$ chmod 444 a.txt
himani@Himani:~/Desktop$ ls -l a.txt
-r--r--r-- 1 himani himani 0 Jan 19 22:18 a.txt
himani@Himani:~/Desktop$
```

## Whoami

**Functionality:** It is basically the concatenation of strings “**who**”, “**am**”, “**i**” as **whoami**. It displays the username of the current user when this command is invoked. It is similar as running the **id** command with the options **-un**.

**Execution:**

A terminal window with a dark background. The title bar shows a window icon and the text "himani@Himani: ~/Desktop". The prompt is "himani@Himani:~/Desktop\$". The user enters "whoami" and the output is "himani". The prompt returns to "himani@Himani:~/Desktop\$".

```
himani@Himani:~/Desktop$ whoami
himani
himani@Himani:~/Desktop$
```

## Logout

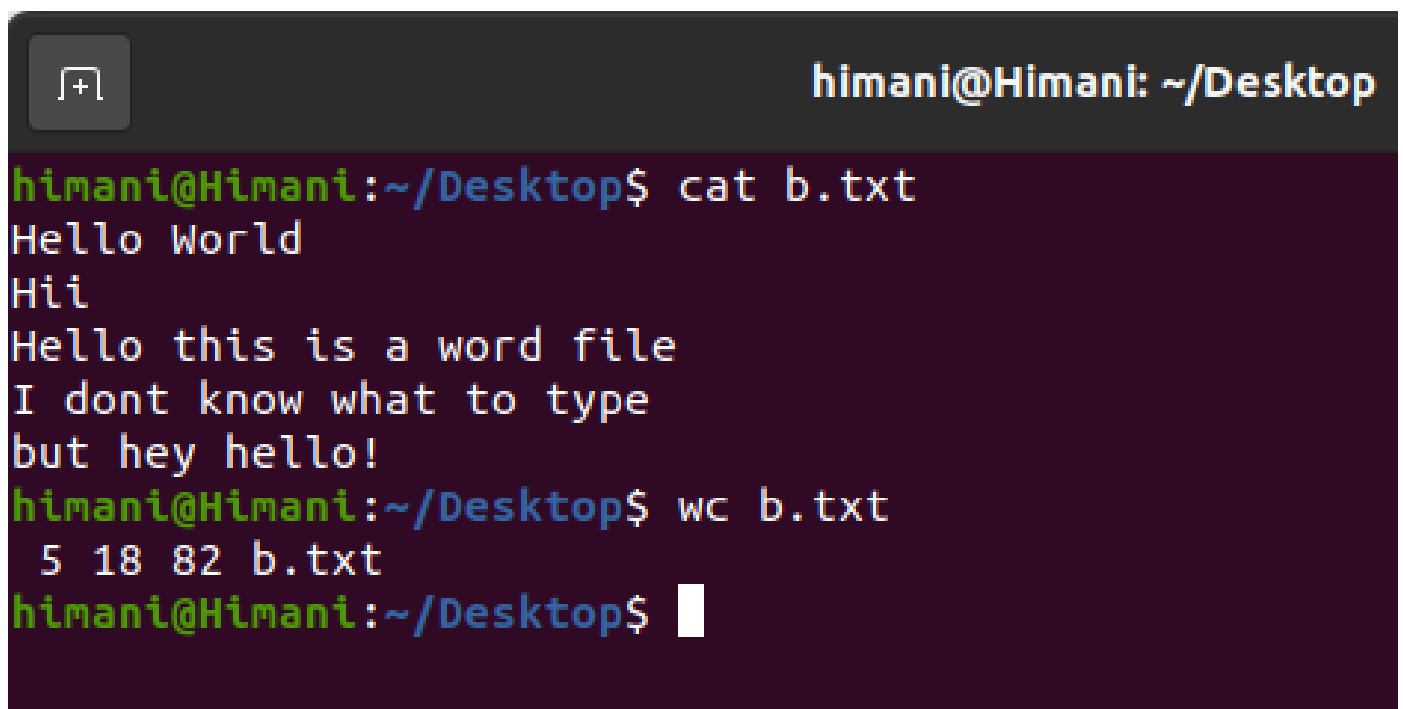
**Functionality:** **logout** command allows you to programmatically logout from your session. causes the session manager to take the requested action immediately.

**Execution:** No output on screen, current user session will be logged out.

## Wc

**Functionality:** **wc** stands for **word count**. As the name implies, it is mainly used for counting purpose. It is used to find out **number of lines, word count, byte and characters count** in the files specified in the file arguments.

**Execution:**

A terminal window with a dark background. The title bar shows a window icon and the text "himani@Himani: ~/Desktop". The prompt is "himani@Himani:~/Desktop\$". The user enters "cat b.txt" and the output is "Hello World", "Hi", and "Hello this is a word file". The user then enters "wc b.txt" and the output is "5 18 82 b.txt". The prompt returns to "himani@Himani:~/Desktop\$".

```
himani@Himani:~/Desktop$ cat b.txt
Hello World
Hi
Hello this is a word file
I dont know what to type
but hey hello!
himani@Himani:~/Desktop$ wc b.txt
 5 18 82 b.txt
himani@Himani:~/Desktop$
```

## Grep

**Functionality:** The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression

**Execution:**

```
himani@Himani: ~/Desktop

himani@Himani:~$ cd Desktop/
himani@Himani:~/Desktop$ grep -i "Hello" b.txt
Hello World
Hello this is a word file
but hey hello!
himani@Himani:~/Desktop$
```

## Sort

**Functionality:** SORT command is used to sort a file, arranging the records in a particular order. By default, the sort command sorts file assuming the contents are ASCII. Using options in the sort command can also be used to sort numerically.

**Execution:**

```
himani@Himani: ~/Desktop

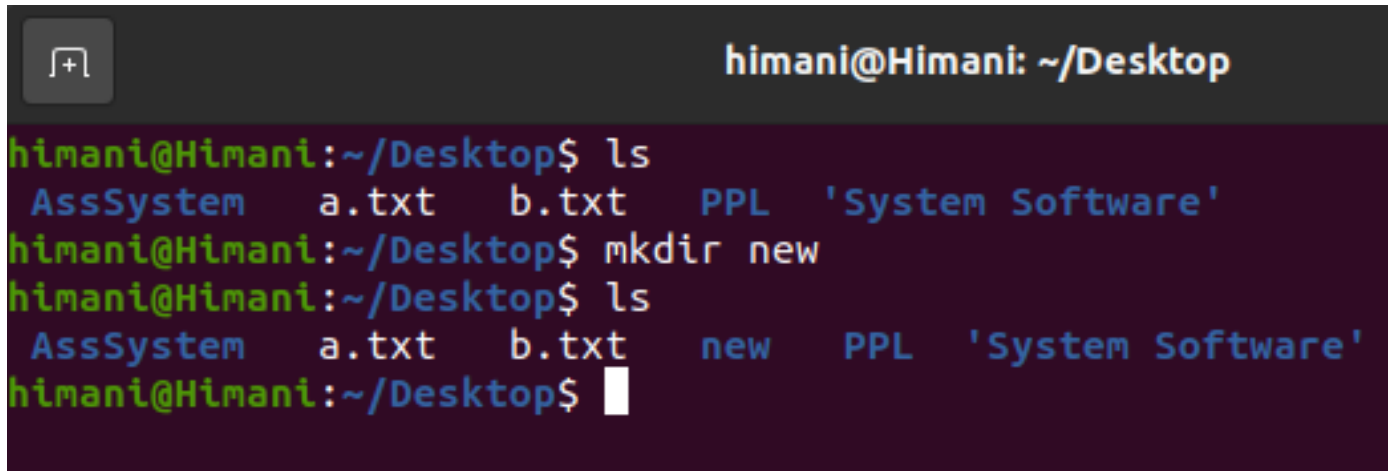
himani@Himani:~/Desktop$ cat b.txt
Hello World
Hii
Hello this is a word file
I dont know what to type
but hey hello!
himani@Himani:~/Desktop$ sort b.txt
but hey hello!
Hello this is a word file
Hello World
Hii
I dont know what to type
himani@Himani:~/Desktop$
```



## Mkdir

**Functionality:** **mkdir** command in Linux allows the user to create directories (also referred to as folders in some operating systems ). This command can create multiple directories at once as well as set the permissions for the directories.

**Execution:**

A terminal window titled 'himani@Himani: ~/Desktop' with a terminal icon in the top-left corner. The terminal shows the following commands and output:

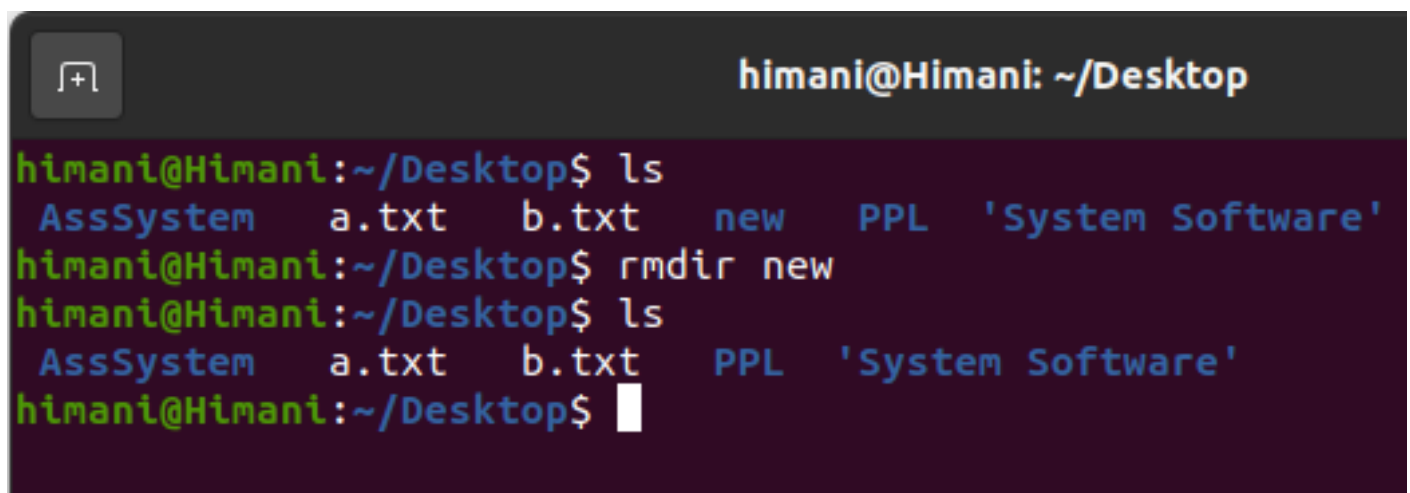
```
himani@Himani:~/Desktop$ ls
AssSystem  a.txt  b.txt  PPL  'System Software'
himani@Himani:~/Desktop$ mkdir new
himani@Himani:~/Desktop$ ls
AssSystem  a.txt  b.txt  new  PPL  'System Software'
himani@Himani:~/Desktop$
```

Created a new empty directory called new

## Rmdir

**Functionality:** **rmdir** command is used remove empty directories from the filesystem in Linux. The rmdir command removes each and every directory specified in the command line only if these directories are empty. So if the specified directory has some directories or files in it then this cannot be removed by *rmdir* command.

**Execution:**

A terminal window titled 'himani@Himani: ~/Desktop' with a terminal icon in the top-left corner. The terminal shows the following commands and output:

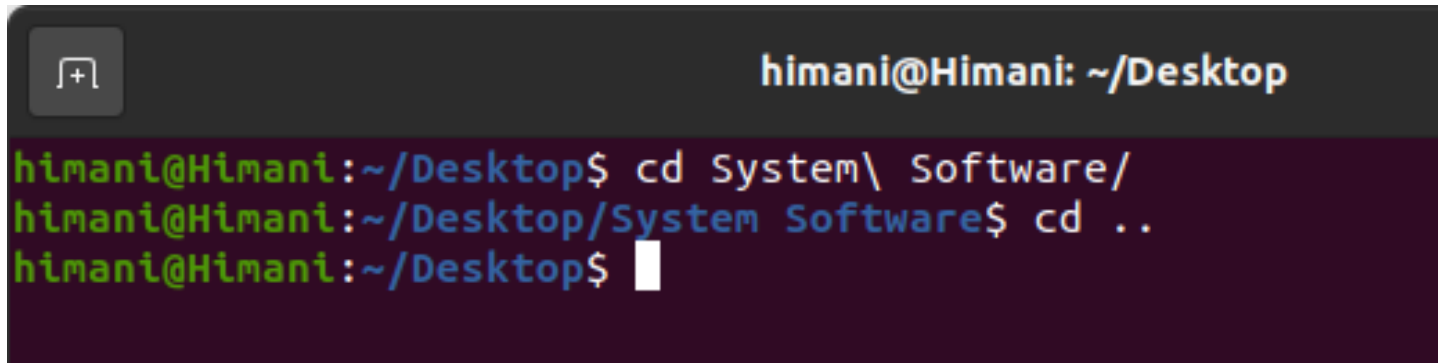
```
himani@Himani:~/Desktop$ ls
AssSystem  a.txt  b.txt  new  PPL  'System Software'
himani@Himani:~/Desktop$ rmdir new
himani@Himani:~/Desktop$ ls
AssSystem  a.txt  b.txt  PPL  'System Software'
himani@Himani:~/Desktop$
```

Removed the empty directory new

## Cd

**Functionality:** Linux **cd** command is used to change the current working directory ( i.e., in which the current user is working). The "cd" stands for '**change directory**.' It is one of the most frequently used commands in the Linux terminal.

**Execution:**

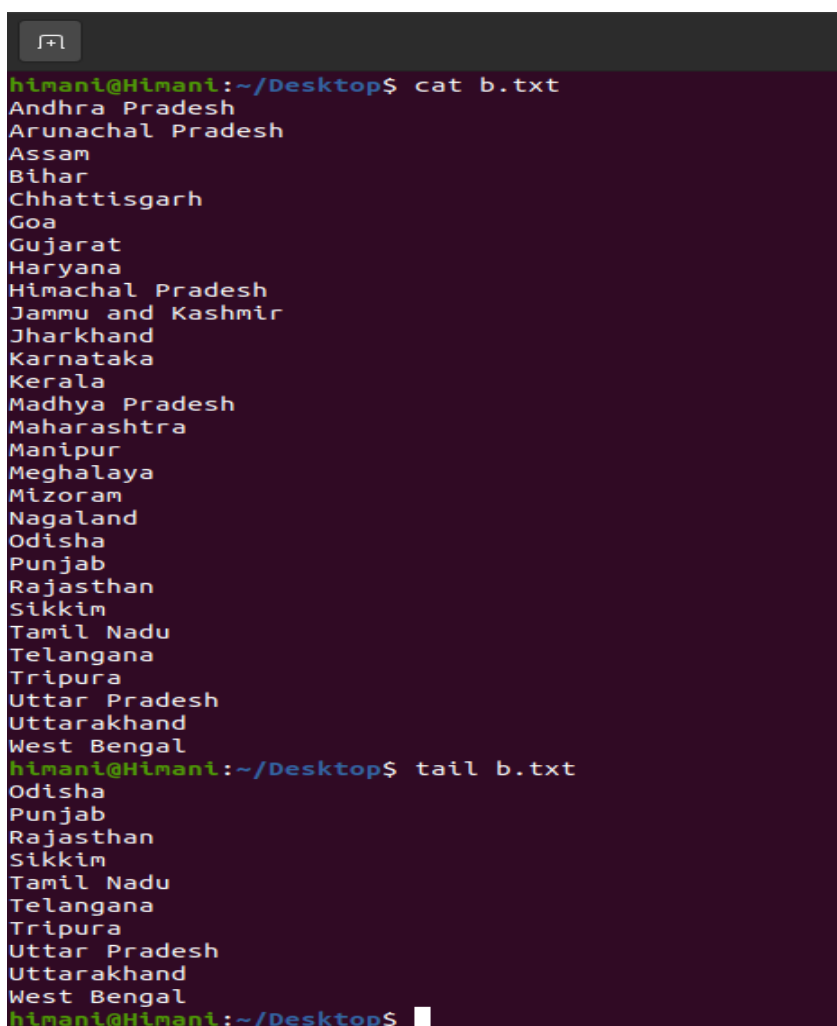
A terminal window with a dark background. The title bar shows a window icon and the text 'himani@Himani: ~/Desktop'. The prompt is 'himani@Himani:~/Desktop\$'. The user enters 'cd System\ Software/'. The prompt changes to 'himani@Himani:~/Desktop/System Software\$'. The user enters 'cd ..'. The prompt returns to 'himani@Himani:~/Desktop\$' with a cursor at the end.

```
himani@Himani:~/Desktop$ cd System\ Software/
himani@Himani:~/Desktop/System Software$ cd ..
himani@Himani:~/Desktop$
```

## Tail

**Functionality:** The tail command, as the name implies, print the last N number of data of the given input. By default it prints the last 10 lines of the specified files. If more than one file name is provided then data from each file is precedes by its file name.

**Execution:**

A terminal window with a dark background. The title bar shows a window icon and the text 'himani@Himani: ~/Desktop'. The prompt is 'himani@Himani:~/Desktop\$'. The user enters 'cat b.txt'. The terminal displays a list of Indian states: Andhra Pradesh, Arunachal Pradesh, Assam, Bihar, Chhattisgarh, Goa, Gujarat, Haryana, Himachal Pradesh, Jammu and Kashmir, Jharkhand, Karnataka, Kerala, Madhya Pradesh, Maharashtra, Manipur, Meghalaya, Mizoram, Nagaland, Odisha, Punjab, Rajasthan, Sikkim, Tamil Nadu, Telangana, Tripura, Uttar Pradesh, Uttarakhand, West Bengal. The user then enters 'tail b.txt'. The terminal displays the last five lines of the file: Odisha, Punjab, Rajasthan, Sikkim, Tamil Nadu. The prompt returns to 'himani@Himani:~/Desktop\$' with a cursor at the end.

```
himani@Himani:~/Desktop$ cat b.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal
himani@Himani:~/Desktop$ tail b.txt
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
himani@Himani:~/Desktop$
```

## Cmp

**Functionality:** **cmp** command in Linux/UNIX is used to compare the two files byte by byte and helps you to find out whether the two files are identical or not.

- When cmp is used for comparison between two files, it reports the location of the first mismatch to the screen if difference is found and if no difference is found *i.e* the files compared are identical.
- cmp displays no message and simply returns the prompt if the the files compared are identical.

**Execution:**

```
himani@Himani: ~/Desktop
himani@Himani:~/Desktop$ cmp a.txt b.txt
a.txt b.txt differ: byte 176, line 17
himani@Himani:~/Desktop$
```

## Diff

**Functionality:** diff stands for **difference**. This command is used to display the differences in the files by comparing the files line by line. Unlike its fellow members, CMP and COMM, it tells us which lines in one file have is to be changed to make the two files identical.

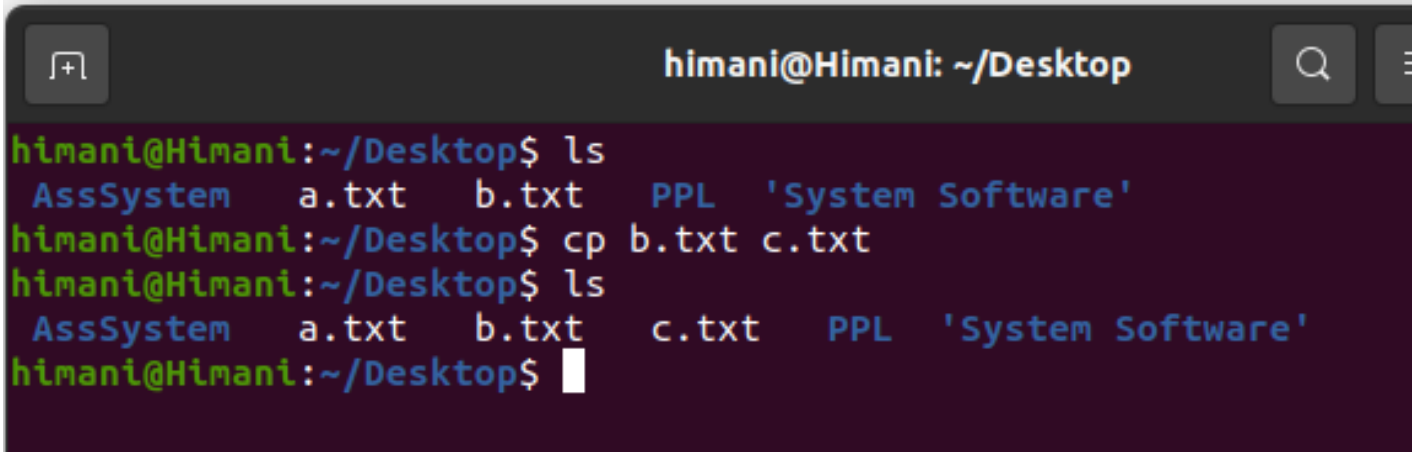
**Execution:**

```
himani@Himani: ~/Desktop
himani@Himani:~/Desktop$ diff a.txt b.txt
16a17,19
> Meghalaya
> Mizoram
> Nagaland
himani@Himani:~/Desktop$
```

## Cp

**Functionality:** This command is used to copy files or group of files or directory. It creates an exact image of a file on a disk with different file name. *cp* command require at least two filenames in its arguments.

**Execution:**

A terminal window titled 'himani@Himani: ~/Desktop' with a search icon and a window control icon. The terminal shows the following commands and output:

```
himani@Himani:~/Desktop$ ls
AssSystem  a.txt  b.txt  PPL  'System Software'
himani@Himani:~/Desktop$ cp b.txt c.txt
himani@Himani:~/Desktop$ ls
AssSystem  a.txt  b.txt  c.txt  PPL  'System Software'
himani@Himani:~/Desktop$
```

## Clear

**Functionality:** **clear** is a standard Unix computer operating system command that is used to clear the terminal screen. This command first looks for a terminal type in the environment and after that, it figures out the **terminfo** database for how to clear the screen. And this command will ignore any command-line parameters that may be present.

**Execution:** \$clear

Will clear the screen

## Df

**Functionality:** The **df** command (short for disk free), is used to display information related to file systems about total space and available space.

**Execution:**

```
himani@Himani:~/Desktop$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            2495008         0   2495008  0% /dev
tmpfs           504928      1352    503576  1% /run
/dev/sda5       95787420 10036832  80841788 12% /
tmpfs           2524624         0   2524624  0% /dev/shm
tmpfs           5120         4      5116  1% /run/lock
tmpfs           2524624         0   2524624  0% /sys/fs/cgroup
/dev/loop0        128        128         0 100% /snap/bare/5
/dev/loop1       63488      63488         0 100% /snap/core20/1270
/dev/loop2       56832      56832         0 100% /snap/core18/2253
/dev/loop4       224256     224256         0 100% /snap/gnome-3-34-1804/72
/dev/loop5       224256     224256         0 100% /snap/gnome-3-34-1804/77
/dev/loop6       66816      66816         0 100% /snap/gtk-common-themes/1519
/dev/loop7       33280      33280         0 100% /snap/snapd/13640
/dev/loop8       66688      66688         0 100% /snap/gtk-common-themes/1515
/dev/loop9       55552      55552         0 100% /snap/snap-store/558
/dev/loop10      44416      44416         0 100% /snap/snapd/14295
/dev/loop11      52224      52224         0 100% /snap/snap-store/547
/dev/sda1        523248         4    523244  1% /boot/efi
tmpfs           504924        36    504888  1% /run/user/1000
/dev/sr0         59650      59650         0 100% /media/himani/VBox_GAs_6.1.24
/dev/loop12      56960      56960         0 100% /snap/core18/2284
himani@Himani:~/Desktop$
```

## Du

**Functionality:** The du command can be used to track the files and directories which are consuming excessive amount of space on hard disk drive.

**Execution:**

```
himani@Himani:~/Desktop$ du
4      ./AssSystem
108    ./System Software
12     ./PPL
140    .
himani@Himani:~/Desktop$
```

## Uname

**Functionality:** The command 'uname' displays the information about the system.

**Execution:**

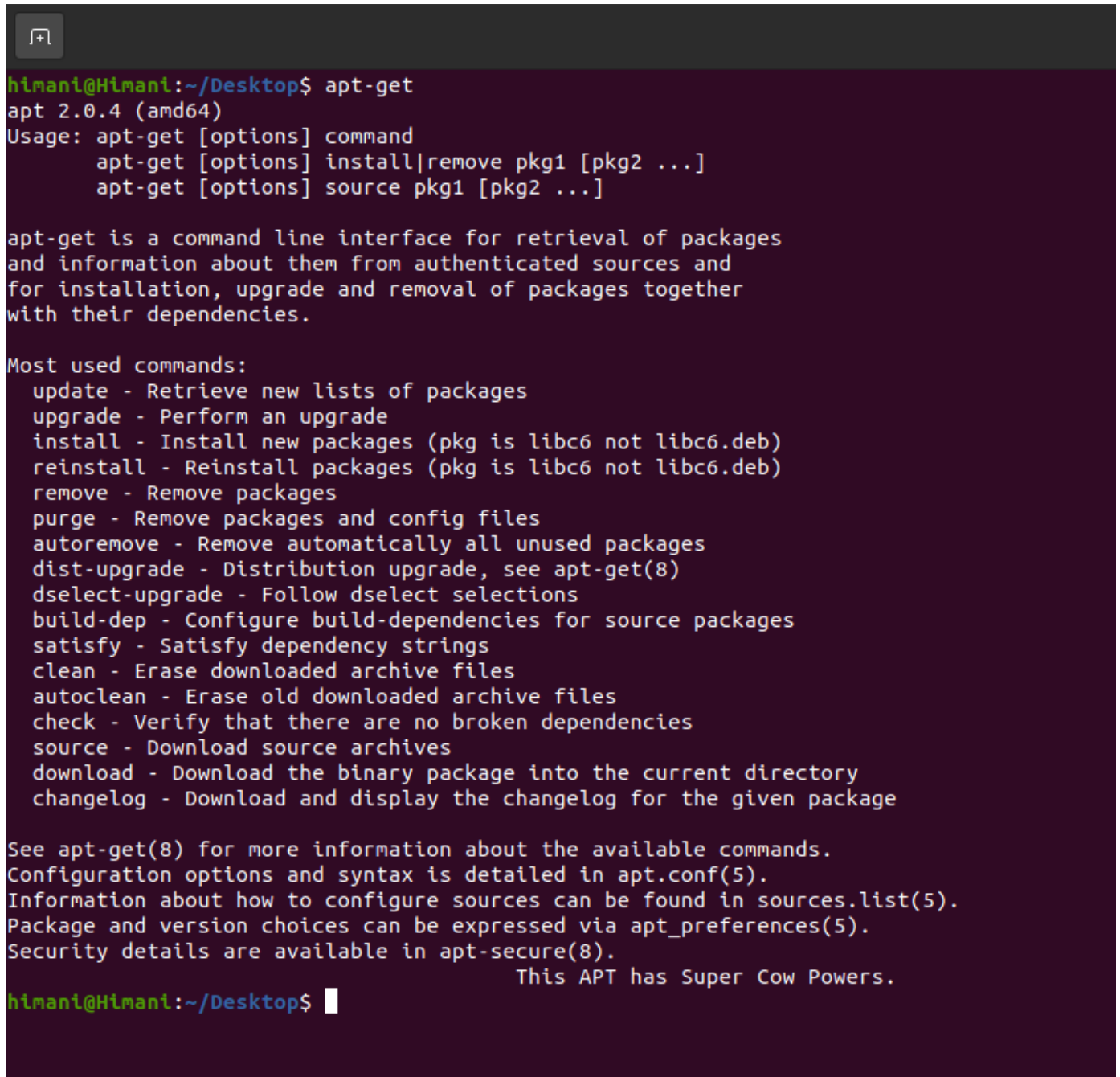
```
himani@Himani: ~/Desktop
himani@Himani:~/Desktop$ uname -a
Linux Himani 5.11.0-27-generic #29~20.04.1-Ubuntu SMP Wed Aug 11 15:58:17 UTC
2021 x86_64 x86_64 x86_64 GNU/Linux
himani@Himani:~/Desktop$
```

## apt-get

**Functionality:** **apt-get** is a command-line tool which helps in handling packages in Linux. Its main task is to retrieve the information and packages from the authenticated sources for installation, upgrade and removal of packages along with their dependencies.

### Execution:

Various apt-get commands are shown below:

A terminal window with a dark background and light-colored text. The prompt is 'himani@Himani:~/Desktop\$'. The command 'apt-get' has been entered, and the output shows the version 'apt 2.0.4 (amd64)' and usage instructions. Below the usage, a detailed description of apt-get is provided, followed by a list of common commands and their functions. At the bottom, there are links to more information and a 'Super Cow Powers' message.

```
himani@Himani:~/Desktop$ apt-get
apt 2.0.4 (amd64)
Usage: apt-get [options] command
       apt-get [options] install|remove pkg1 [pkg2 ...]
       apt-get [options] source pkg1 [pkg2 ...]

apt-get is a command line interface for retrieval of packages
and information about them from authenticated sources and
for installation, upgrade and removal of packages together
with their dependencies.

Most used commands:
  update - Retrieve new lists of packages
  upgrade - Perform an upgrade
  install - Install new packages (pkg is libc6 not libc6.deb)
  reinstall - Reinstall packages (pkg is libc6 not libc6.deb)
  remove - Remove packages
  purge - Remove packages and config files
  autoremove - Remove automatically all unused packages
  dist-upgrade - Distribution upgrade, see apt-get(8)
  dselect-upgrade - Follow dselect selections
  build-dep - Configure build-dependencies for source packages
  satisfy - Satisfy dependency strings
  clean - Erase downloaded archive files
  autoclean - Erase old downloaded archive files
  check - Verify that there are no broken dependencies
  source - Download source archives
  download - Download the binary package into the current directory
  changelog - Download and display the changelog for the given package

See apt-get(8) for more information about the available commands.
Configuration options and syntax is detailed in apt.conf(5).
Information about how to configure sources can be found in sources.list(5).
Package and version choices can be expressed via apt_preferences(5).
Security details are available in apt-secure(8).
                                     This APT has Super Cow Powers.

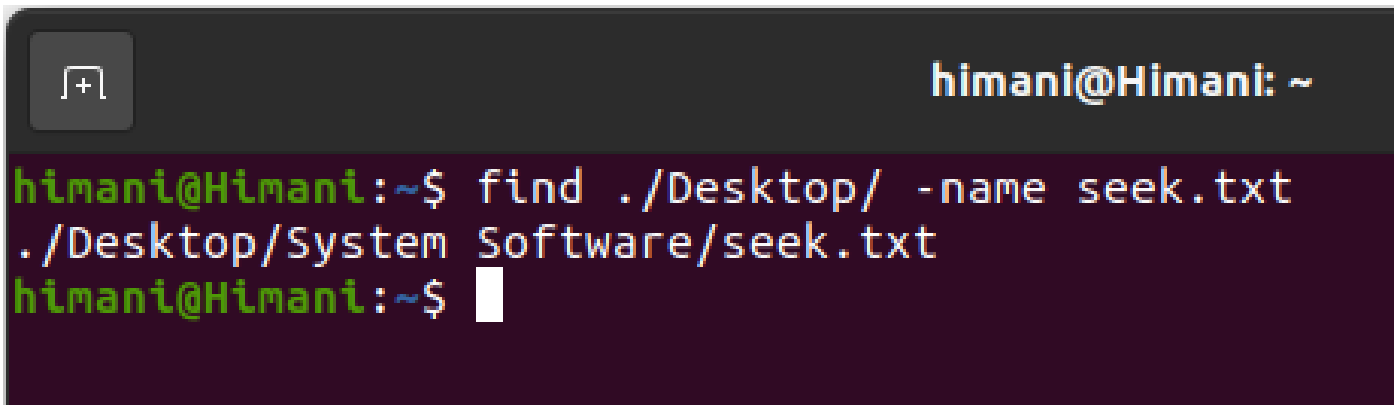
himani@Himani:~/Desktop$
```

## Find

**Functionality:** The **find** command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent

operations on them. It supports searching by file, folder, name, creation date, modification date, owner and permissions. By using the '-exec' other UNIX commands can be executed on files or folders found.

#### Execution:

A terminal window titled 'himani@Himani: ~' with a dark background. The prompt is 'himani@Himani:~\$'. The user enters the command 'find ./Desktop/ -name seek.txt'. The output is './Desktop/System Software/seek.txt'. The prompt returns to 'himani@Himani:~\$' with a cursor.

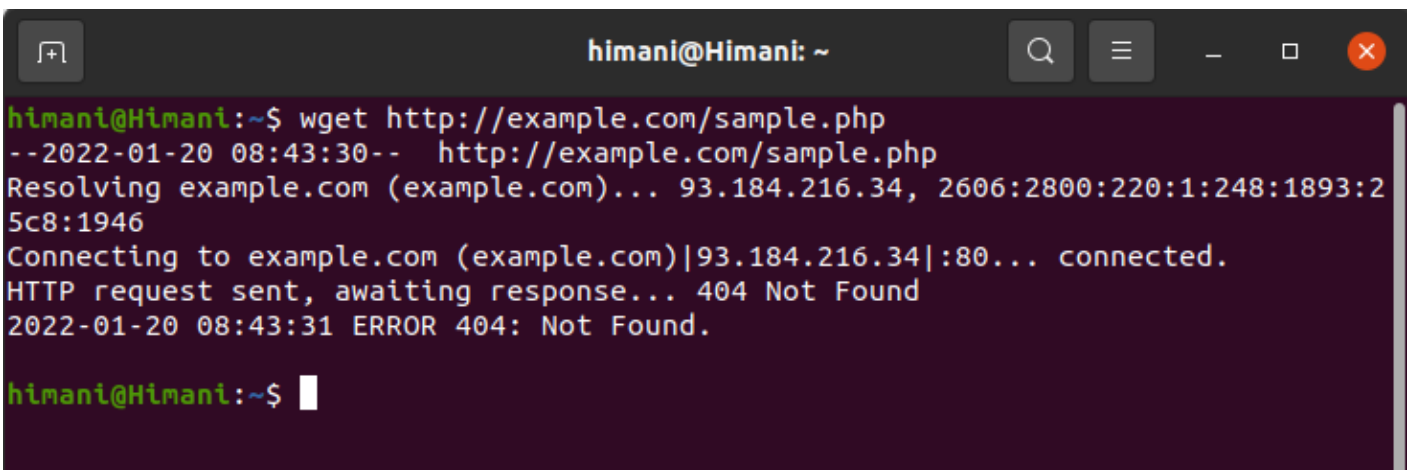
```
himani@Himani:~$ find ./Desktop/ -name seek.txt
./Desktop/System Software/seek.txt
himani@Himani:~$
```

## Wget

**Functionality:** **Wget** is the non-interactive network downloader which is used to download files from the server even when the user has not logged on to the system and it can work in the background without hindering the current process.

#### Execution:

To simply download a webpage

A terminal window titled 'himani@Himani: ~' with a dark background. The prompt is 'himani@Himani:~\$'. The user enters the command 'wget http://example.com/sample.php'. The output shows the process of resolving the URL, connecting to the server, and receiving a '404 Not Found' error. The prompt returns to 'himani@Himani:~\$' with a cursor.

```
himani@Himani:~$ wget http://example.com/sample.php
--2022-01-20 08:43:30-- http://example.com/sample.php
Resolving example.com (example.com)... 93.184.216.34, 2606:2800:220:1:248:1893:2
5c8:1946
Connecting to example.com (example.com)|93.184.216.34|:80... connected.
HTTP request sent, awaiting response... 404 Not Found
2022-01-20 08:43:31 ERROR 404: Not Found.

himani@Himani:~$
```

## Top

**Functionality:** **top** command is used to show the Linux processes. It provides a dynamic real-time view of the running system. Usually, this command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel.

#### Execution:



## Display Specific User Process: \$top -u Himani

himani@Himani										
top - 08:48:17 up 10 min, 1 user, load average: 0.06, 0.26, 0.21										
Tasks: 195 total, 1 running, 194 sleeping, 0 stopped, 0 zombie										
%Cpu(s): 10.0 us, 1.3 sy, 0.0 ni, 87.3 id, 1.3 wa, 0.0 hi, 0.0 si, 0.0 st										
MiB Mem : 4930.9 total, 2506.7 free, 1151.9 used, 1272.4 buff/cache										
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used. 3504.9 avail Mem										
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+ COMMAND
1534	himani	20	0	3762364	384744	139444	S	5.6	7.6	0:34.90 gnome-shell
1325	himani	20	0	602232	127004	71228	S	3.0	2.5	0:15.18 Xorg
1896	himani	20	0	823560	50740	38356	S	1.3	1.0	0:01.98 gnome-terminal-
1931	himani	20	0	429012	31452	21952	S	0.3	0.6	0:00.20 update-notifier
1259	himani	20	0	19284	10460	8088	S	0.0	0.2	0:00.28 systemd
1260	himani	20	0	103628	3544	4	S	0.0	0.1	0:00.00 (sd-pam)
1265	himani	9	-11	1941512	19424	15220	S	0.0	0.4	0:00.24 pulseaudio
1267	himani	39	19	593604	24028	16080	S	0.0	0.5	0:00.15 tracker-miner-f
1270	himani	20	0	248684	7380	6436	S	0.0	0.1	0:00.02 gnome-keyring-d
1274	himani	20	0	8464	5612	3960	S	0.0	0.1	0:00.30 dbus-daemon
1290	himani	20	0	322068	7752	6744	S	0.0	0.2	0:00.03 gvfsd
1295	himani	20	0	378344	6424	5768	S	0.0	0.1	0:00.00 gvfsd-fuse
1297	himani	20	0	322560	9176	7760	S	0.0	0.2	0:00.07 gvfs-udisks2-vo
1308	himani	20	0	246608	6788	6096	S	0.0	0.1	0:00.01 gvfs-gphoto2-vo
1312	himani	20	0	244332	6320	5728	S	0.0	0.1	0:00.01 gvfs-mtp-volume
1316	himani	20	0	172652	6576	5932	S	0.0	0.1	0:00.00 gdm-x-session
1319	himani	20	0	244508	5948	5416	S	0.0	0.1	0:00.00 gvfs-goa-volume
1323	himani	20	0	627588	36756	30772	S	0.0	0.7	0:00.04 goa-daemon
1335	himani	20	0	397120	8844	7788	S	0.0	0.2	0:00.02 goa-identity-se
1337	himani	20	0	325356	8912	7904	S	0.0	0.2	0:00.04 gvfs-afc-volume
1365	himani	20	0	197052	13872	12260	S	0.0	0.3	0:00.03 gnome-session-b
1438	himani	20	0	31244	364	0	S	0.0	0.0	0:00.00 VBoxClient
1439	himani	20	0	163512	3988	3448	S	0.0	0.1	0:00.00 VBoxClient
1450	himani	20	0	31244	360	0	S	0.0	0.0	0:00.00 VBoxClient
1451	himani	20	0	163480	2812	2444	S	0.0	0.1	0:00.00 VBoxClient
1457	himani	20	0	31244	356	0	S	0.0	0.0	0:00.00 VBoxClient
1458	himani	20	0	163996	2692	2328	S	0.0	0.1	0:01.61 VBoxClient
1465	himani	20	0	31244	360	0	S	0.0	0.0	0:00.00 VBoxClient
1466	himani	20	0	165760	3452	2916	S	0.0	0.1	0:00.02 VBoxClient
1472	himani	20	0	6040	460	0	S	0.0	0.0	0:00.00 ssh-agent
1493	himani	20	0	305536	7172	6392	S	0.0	0.1	0:00.01 at-spi-bus-laun
1498	himani	20	0	7248	4240	3776	S	0.0	0.1	0:00.01 dbus-daemon
1513	himani	20	0	98696	4364	3944	S	0.0	0.1	0:00.00 gnome-session-c
1520	himani	20	0	492944	15816	13136	S	0.0	0.3	0:00.08 gnome-session-b
1559	himani	20	0	393180	8232	6784	S	0.0	0.2	0:00.01 ibus-daemon
1563	himani	20	0	171336	6912	6288	S	0.0	0.1	0:00.00 ibus-memconf
1564	himani	20	0	284300	31512	17980	S	0.0	0.6	0:01.08 ibus-extension-
1567	himani	20	0	206524	28052	18244	S	0.0	0.6	0:00.20 ibus-x11
1570	himani	20	0	245136	7236	6584	S	0.0	0.1	0:00.02 ibus-portal
1581	himani	20	0	162912	6376	5720	S	0.0	0.1	0:00.03 at-spi2-registr
1585	himani	20	0	244228	4592	4156	S	0.0	0.1	0:00.00 xdg-permission-

## Mpstate

**Functionality:** **mpstat** is a command that is used to report processor related statistics. It accurately displays the statistics of the CPU usage of the system. It displays information about CPU utilization and performance. It initializes the first processor with CPU 0, the second one with CPU 1, and so on.

### Execution:



```
himani@Himani: ~  
himani@Himani:~$ mpstat  
Linux 5.11.0-27-generic (Himani)      20/01/22      _x86_64_      (1 CPU)  
08:52:25 AM IST  CPU      %usr      %nice      %sys %iowait      %irq      %soft      %steal %  
guest %gnice %idle  
08:52:25 AM IST  all      15.14      1.55      3.79      1.63      0.00      0.12      0.00  
0.00      0.00      77.78  
himani@Himani:~$
```

## Netstat

**Functionality:** Netstat command displays various network related information such as network connections, routing tables, interface statistics, masquerade connections, multicast memberships etc.,

**Execution:**

**-a -all:** Show both listening and non-listening sockets.

**-at :** To list all tcp ports.

```
himani@Himani:~$ netstat -at  
Active Internet connections (servers and established)  
Proto Recv-Q Send-Q Local Address           Foreign Address         State  
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN  
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN  
tcp        0      0 Himani:35546           ec2-52-39-178-28.:https ESTABLISHED  
tcp        0      0 Himani:42098           whatsapp-cdn-shv-:https ESTABLISHED  
tcp6       0      0 ip6-localhost:ipp      [::]:*                 LISTEN  
himani@Himani:~$
```

## Sar

**Functionality:** System Activity Report

It can be used to monitor Linux system's resources like CPU usage, Memory utilization, I/O devices consumption, Network monitoring, Disk usage, process and thread allocation, battery performance, Plug and play devices, Processor performance, file system and more. Linux system Monitoring and analyzing aids understanding system resource usage which can help to improve system performance to handle more requests.

**Execution:**

**Syntax :**

\$ sar -[ options ] time\_interval number\_of\_tines\_to\_display

```
himani@Himani:~$ sar -u 2 5
Linux 5.11.0-27-generic (Himani)      20/01/22      _x86_64_      (1 CPU)

09:03:15 AM IST      CPU      %user      %nice      %system      %iowait      %steal      %idle
09:03:17 AM IST      all      4.02      0.00      0.50      0.50      0.00      94.97
09:03:19 AM IST      all      2.53      0.00      0.51      1.01      0.00      95.96
09:03:21 AM IST      all      3.54      0.00      1.01      0.00      0.00      95.45
09:03:23 AM IST      all      2.03      0.00      0.51      0.51      0.00      96.95
09:03:25 AM IST      all      3.02      0.00      0.50      0.00      0.00      96.48
Average:      all      3.03      0.00      0.61      0.40      0.00      95.96
himani@Himani:~$
```

## Chown

**Functionality:** Different users in the operating system have ownership and permission to ensure that the files are secure and put restrictions on who can modify the contents of the files. **chown** command is used to change the file Owner or group. Whenever you want to change ownership you can use chown command.

### Execution:

Changed ownership of c.txt

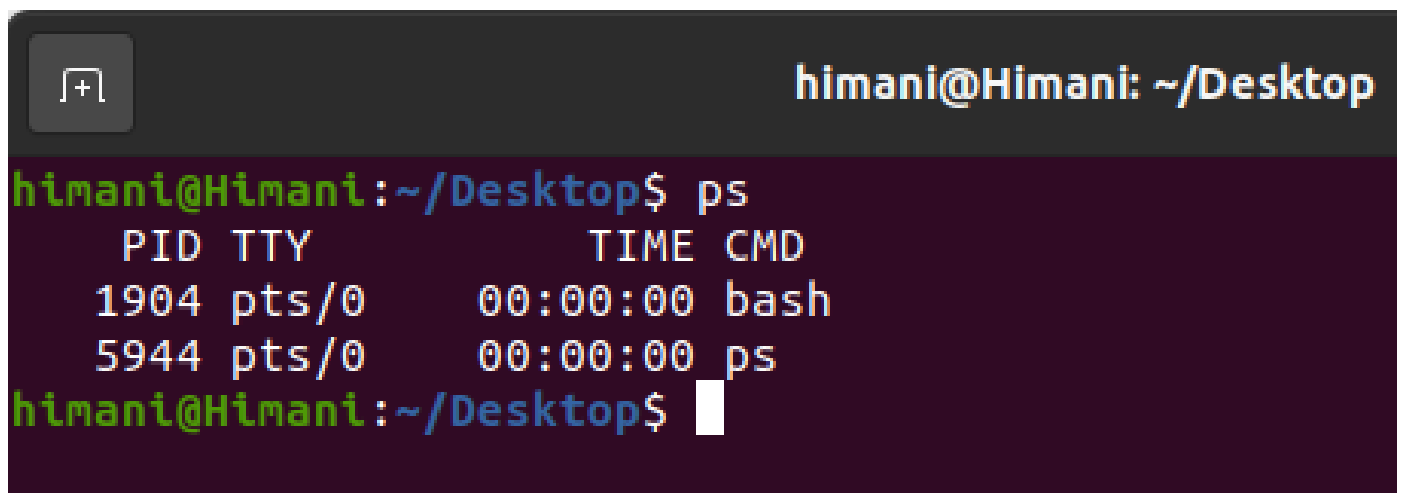
```
himani@Himani:~/Desktop$ ls -l
total 24
drwxrwxr-x 2 himani himani 4096 Jan 20 17:35 AI
-rw-rw-r-- 1 himani himani 273 Jan 19 23:18 a.txt
-rw-rw-r-- 1 himani himani 300 Jan 19 23:12 b.txt
-rw-rw-r-- 1 himani himani 300 Jan 20 00:05 c.txt
drwxrwxr-x 2 himani himani 4096 Jan 9 09:57 PPL
drwxrwxr-x 2 himani himani 4096 Jan 11 11:57 'System Software'
himani@Himani:~/Desktop$ sudo chown root c.txt
[sudo] password for himani:
himani@Himani:~/Desktop$ ls -l
total 24
drwxrwxr-x 2 himani himani 4096 Jan 20 17:35 AI
-rw-rw-r-- 1 himani himani 273 Jan 19 23:18 a.txt
-rw-rw-r-- 1 himani himani 300 Jan 19 23:12 b.txt
-rw-rw-r-- 1 root himani 300 Jan 20 00:05 c.txt
drwxrwxr-x 2 himani himani 4096 Jan 9 09:57 PPL
drwxrwxr-x 2 himani himani 4096 Jan 11 11:57 'System Software'
himani@Himani:~/Desktop$
```

## 2. Linux commands related with process: ps, kill, background processing (with &)

### ps

**Functionality:** Linux provides us a utility called **ps** for viewing information related with the processes on a system which stands as abbreviation for “**Process Status**”. **ps** command is used to list the currently running processes and their PIDs along with some other information depends on different options.

**Execution:**

A terminal window with a dark background. The title bar shows a window icon and the text 'himani@Himani: ~/Desktop'. The prompt is 'himani@Himani:~/Desktop\$'. The command 'ps' has been entered. The output is a table with four columns: PID, TTY, TIME, and CMD. There are two rows of data: one for PID 1904 (bash) and one for PID 5944 (ps).

```
himani@Himani:~/Desktop$ ps
  PID TTY          TIME CMD
 1904 pts/0    00:00:00 bash
 5944 pts/0    00:00:00 ps
himani@Himani:~/Desktop$
```

### Kill

**Functionality:** *kill* command in Linux (located in `/bin/kill`), is a built-in command which is used to terminate processes manually. *kill* command sends a signal to a process which terminates the process. If the user doesn't specify any signal which is to be sent along with *kill* command then default *TERM* signal is sent that terminates the process.

**Execution:**

**kill -l:** To display all the available signals you can use below command option:

```
himani@Himani:~/Desktop$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS      8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM    15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD    18) SIGCONT    19) SIGSTOP    20) SIGTSTP
21) SIGTTIN    22) SIGTTOU    23) SIGURG     24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF    28) SIGWINCH   29) SIGIO      30) SIGPWR
31) SIGSYS     34) SIGRTMIN   35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```

## background processing (with &)

**Functionality:** In Linux terminology (and in Unix in general), a background process is a process that is started from a shell (or terminal) and then runs independently. When a background process has been launched from a terminal session, the same terminal will be immediately available to execute other commands.

### Execution:

To run your process or command/shell script in the background, include an & (an ampersand) at the end of the command/shell script you use to run the job. For example:

```
himani@Himani: ~/Desktop

himani@Himani:~/Desktop$ jobs
[1]+  Running                  sleep 1000 &
himani@Himani:~/Desktop$ cd System\ Software/ &
[2] 6201
himani@Himani:~/Desktop$ jobs
[1]-  Running                  sleep 1000 &
[2]+  Done                     cd System\ Software/
himani@Himani:~/Desktop$
```