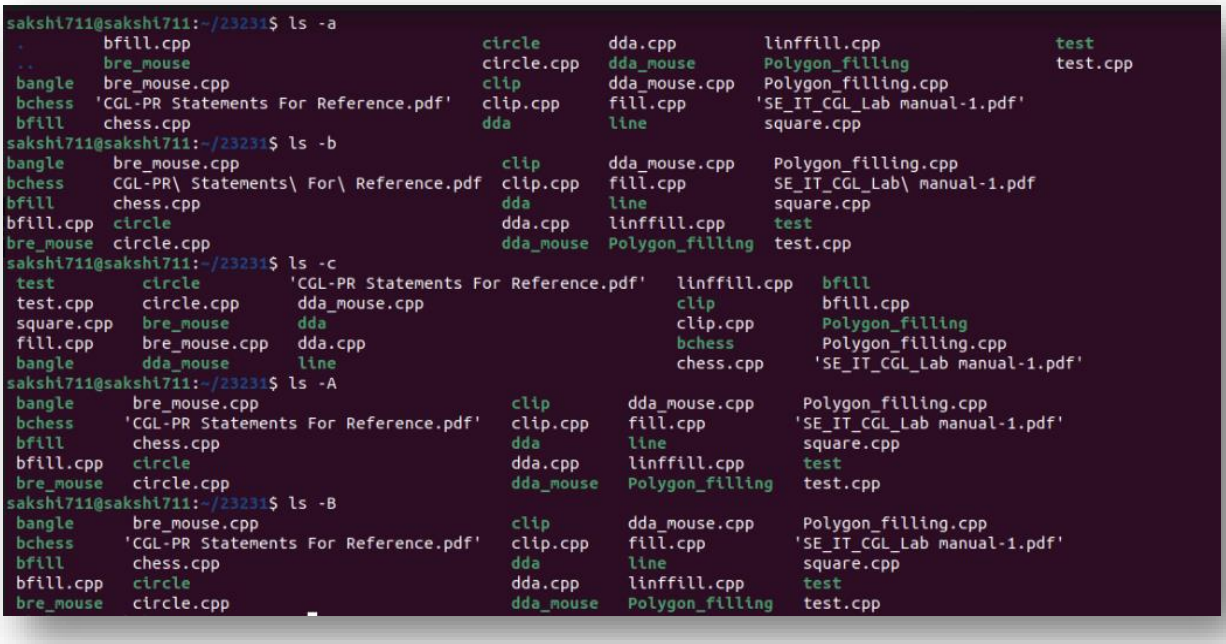
ASSIGNMENT NO: 1

Sakshi Santosh Gangurde
Roll no: 33229

1)ls

Ls is a Linux shell command that lists directory contents of files and directories. It provides valuable information about files, directories, and their attributes

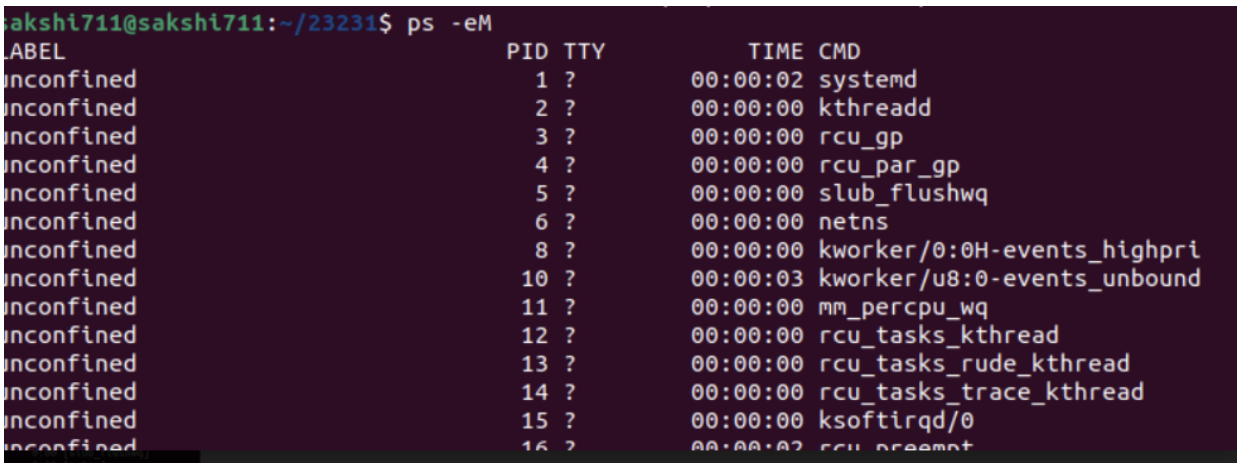
Syntax: *ls [option] [file/directory]*



2) Ps

allows you to view information about the processes running on your Linux system.

Syntax: *ps [options]*



```
sakshi711@sakshi711:~/23231$ ps -eLf
UID          PID    PPID     LWP  C NLWP STIME TTY          TIME CMD
root           1       0       1  0    1 16:04 ?        00:00:02 /sbin/init splash
root           2       0       2  0    1 16:04 ?        00:00:00 [kthreadd]
root           3       2       3  0    1 16:04 ?        00:00:00 [rcu_gp]
root           4       2       4  0    1 16:04 ?        00:00:00 [rcu_par_gp]
root           5       2       5  0    1 16:04 ?        00:00:00 [slub_flushwq]
root           6       2       6  0    1 16:04 ?        00:00:00 [netns]
root           8       2       8  0    1 16:04 ?        00:00:00 [kworker/0:0H-events_highpri]
root          10       2      10  0    1 16:04 ?        00:00:03 [kworker/u8:0-events_unbound]
root          11       2      11  0    1 16:04 ?        00:00:00 [mm_percpu_wq]
root          12       2      12  0    1 16:04 ?        00:00:00 [rcu_tasks_kthread]
root          13       2      13  0    1 16:04 ?        00:00:00 [rcu_tasks_rude_kthread]
root          14       2      14  0    1 16:04 ?        00:00:00 [rcu_tasks_trace_kthread]
```

```
sakshi711@sakshi711:~/23231$ ps -ax
    PID TTY      STAT   TIME COMMAND
      1 ?        Ss     0:02 /sbin/init splash
      2 ?        S      0:00 [kthreadd]
      3 ?        I<     0:00 [rcu_gp]
      4 ?        I<     0:00 [rcu_par_gp]
      5 ?        I<     0:00 [slub_flushwq]
      6 ?        I<     0:00 [netns]
      8 ?        I<     0:00 [kworker/0:0H-events_highpri]
     10 ?        I      0:03 [kworker/u8:0-events_unbound]
     11 ?        I<     0:00 [mm_percpu_wq]
     12 ?        I      0:00 [rcu_tasks_kthread]
     13 ?        I      0:00 [rcu_tasks_rude_kthread]
     14 ?        I      0:00 [rcu_tasks_trace_kthread]
     15 ?        S      0:00 [ksoftirqd/0]
```

```
sakshi711@sakshi711:~/23231$ ps -ejH
    PID    PGID     SID TTY          TIME CMD
      2       0       0 ?        00:00:00 kthreadd
      3       0       0 ?        00:00:00   rcu_gp
      4       0       0 ?        00:00:00   rcu_par_gp
      5       0       0 ?        00:00:00   slub_flushwq
      6       0       0 ?        00:00:00   netns
      8       0       0 ?        00:00:00   kworker/0:0H-events_highpri
     10       0       0 ?        00:00:03   kworker/u8:0-events_freezable_power_
     11       0       0 ?        00:00:00   mm_percpu_wq
     12       0       0 ?        00:00:00   rcu_tasks_kthread
     13       0       0 ?        00:00:00   rcu_tasks_rude_kthread
     14       0       0 ?        00:00:00   rcu_tasks_trace_kthread
     15       0       0 ?        00:00:00   ksoftirqd/0
     16       0       0 ?        00:00:02   rcu_preempt
```

```
sakshi711@sakshi711:~/23231$ ps -e
    PID TTY          TIME CMD
      1 ?        00:00:02 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 ?        00:00:03 kworker/u8:0-events_unbound
     11 ?        00:00:00 mm_percpu_wq
     12 ?        00:00:00 rcu_tasks_kthread
     13 ?        00:00:00 rcu_tasks_rude_kthread
     14 ?        00:00:00 rcu_tasks_trace_kthread
     15 ?        00:00:00 ksoftirqd/0
```

3) echo

The echo command in Linux is a built-in command that allows users to display lines of text or strings that are passed as arguments.

Syntax: *echo [option] [string]*

```
sakshigangurde711@Sakshi:/mnt/d$ echo "Its PICT"
Its PICT
sakshigangurde711@Sakshi:/mnt/d$ echo -n
sakshigangurde711@Sakshi:/mnt/d$ echo -e

sakshigangurde711@Sakshi:/mnt/d$ echo -e "Hello"
Hello
sakshigangurde711@Sakshi:/mnt/d$ echo -E

sakshigangurde711@Sakshi:/mnt/d$
```

4) read

read command in Linux system is used to read from a file descriptor. Basically, this command read up the total number of bytes from the specified file descriptor into the buffer

Syntax: *read*

```
sakshigangurde711@Sakshi:/mnt/d$ read name
Sakshi
sakshigangurde711@Sakshi:/mnt/d$ echo name
name
sakshigangurde711@Sakshi:/mnt/d$ echo $name
Sakshi
```

5) touch

The touch command is a standard command used in the UNIX/Linux operating system which is used to create, change and modify the timestamps of a file.

Syntax: *touch [options] file_name*

```
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ ls -l
total 11632
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 3708882 Jul  2 08:15 'LP1_NBA Lab Manual_2
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 2930953 Jul  2 08:14  Syllabus-teit.pdf
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 3771348 Jul  2 08:15 'TE_IT_HCIL_Lab manua
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 1494278 Jul  2 08:14 'TE_IT_OSL_Lab Manual
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ touch Syllabus-teit
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ ls -l
total 11632
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 3708882 Jul  2 08:15 'LP1_NBA Lab Manual_2
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711       0 Jul  4 17:23  Syllabus-teit
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 2930953 Jul  2 08:14  Syllabus-teit.pdf
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 3771348 Jul  2 08:15 'TE_IT_HCIL_Lab manua
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 1494278 Jul  2 08:14 'TE_IT_OSL_Lab Manual
```

6) cat

The cat command in Linux is more than just a simple tool; it's a versatile companion for various file-related operations, allowing users to view, concatenate, create, copy, merge, and manipulate file contents.

Syntax: *cat [OPTION] [FILE]*

```
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ cat demo.txt
cat: demo.txt: No such file or directory
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ cat >demo.txt
file
.
^C
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ cat demo.txt
file
.
```

7) grep

The grep command in Unix/Linux is a powerful tool used for searching and manipulating text patterns within files.

Syntax: *grep [options] pattern [files]*

```
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ grep linux demo.txt
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ gep -v Linux demo.txt
Command 'gep' not found, but there are 16 similar ones.
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ grep -v Linux demo.txt
file
.
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ grep Linux demo.txt
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ grep -c Linux demo.txt
0
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ grep -o Linux demo.txt
```

8) sed

SED command in UNIX stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion.

Syntax: *sed OPTIONS... [SCRIPT] [INPUTFILE...]*

```
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ cat demo.txt
I am from PICT
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ sed 's/PICT/Nashik.g' demo.txt
sed: -e expression #1, char 15: unterminated `s' command
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ sed 's/PICT/Nashik/g' demo.txt
I am from Nashik
sakshigangurde711@Sakshi:/mnt/e/TE-IT$
```

9) chmod

In Unix operating systems, the chmod command is used to change the access mode of a file. The name is an abbreviation of change mode.

Syntax: *chmod [options] [mode] [File_name]*

```
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ ls -ltr
total 11632
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 2930953 Jul  2 08:14  Syllabus-teit.pdf
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 1494278 Jul  2 08:14 'TE_IT_OSL_Lab Manual (1).pdf'
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 3708882 Jul  2 08:15 'LP1_NBA Lab Manual_2022-23_ ML ,DAA -ADBMS.pdf'
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711 3771348 Jul  2 08:15 'TE_IT_HCIL_Lab manual_Updated.pdf'
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711       0 Jul  4 17:30  Syllabus-teit
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711      14 Jul  4 17:57  demo.txt.txt
-rwxrwxrwx 1 sakshigangurde711 sakshigangurde711      15 Jul  4 17:58  demo.txt
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ chmod go-w demo.txt
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ cat demo.txt
I am from PICT
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ chmod u+rwx demo.txt
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ cat demo.txt
I am from PICT
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ chmod u+rw,go+r demo.txt
sakshigangurde711@Sakshi:/mnt/e/TE-IT$ cat demo.txt
I am from PICT
```

10) fork

The Fork system call is used for creating a new process in Linux, and Unix systems, which is called the child process, which runs concurrently with the process that makes the fork() call (parent process).

Syntax: *fork()*

```
sakshigangurde711@Sakshi:/mnt/d/hugo$ gcc -o os os.c
sakshigangurde711@Sakshi:/mnt/d/hugo$ ./os
Hello from the parent process! PID: 72, Child PID: 73
Hello from the child process! PID: 73
```

```c
#include <stdio.h>
#include <unistd.h>
int main()
{
    pid_t pid = fork();
    if (pid < 0)
    {
        // Fork failed
        fprintf(stderr, "Fork failed\n");
        return 1;
    }
    else if (pid == 0)
    {
        // Child process
        printf("Hello from the child process! PID: %d\n", getpid());
    }
    else
    {
        // Parent process
        printf("Hello from the parent process! PID: %d, Child PID: %d\n", getpid(), pid);
    }

    return 0;
}
```

11) pwd

The ‘pwd,’ which stands for “print working directory.” In this article, we will delve into the ‘pwd’ command, exploring its functionality, usage, and various examples.

Syntax: *pwd [OPTIONS]*

```
sakshigangurde711@Sakshi:~$ pwd
/home/sakshigangurde711
sakshigangurde711@Sakshi:~$ cd /mnt/d
sakshigangurde711@Sakshi:/mnt/d$ pwd
/mnt/d
```

12) kill

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 that terminates the process.

Syntax: *kill [signal] PID*

```
sakshigangurde711@Sakshi:/mnt/e$ 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
sakshigangurde711@Sakshi:/mnt/e$ ps
  PID TTY          TIME CMD
  149 pts/3    00:00:00 bash
  161 pts/3    00:00:00 ps
sakshigangurde711@Sakshi:/mnt/e$ kill 161
-bash: kill: (161) - No such process
```

13) ifconfig

Knowing your IP address is fundamental for network administration, troubleshooting, and various Linux system tasks. In this article, we will explore several methods to find your IP address in a Linux environment.

Syntax: *ifconfig [interface] [options]*

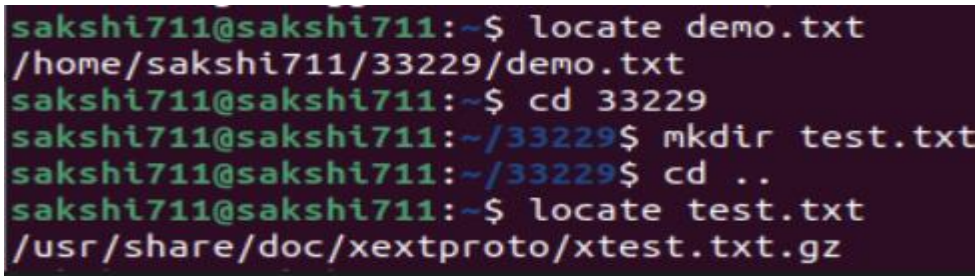
```
sakshigangurde711@Sakshi:/mnt/e$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.26.38.205  netmask 255.255.240.0  broadcast 172.26.47.255
        inet6 fe80::215:5dff:fee2:31f2  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:e2:31:f2  txqueuelen 1000  (Ethernet)
        RX packets 93  bytes 154360 (154.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 80  bytes 6131 (6.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

14) locate

locate command in Linux is used to find the files by name. There are two most widely used file-searching utilities accessible to users called to find and locate.

Syntax: *locate [OPTION]... PATTERN...*



15) ping

Ensuring a stable and reliable internet connection is crucial for seamless navigation and efficient communication in the world of Linux.

Syntax: *ping [options] host_or_IP_address*

```
sakshigangurde711@Sakshi:/mnt/e$ ping www.google.com
PING www.google.com (216.58.196.68) 56(84) bytes of data.
64 bytes from kul01s09-in-f68.1e100.net (216.58.196.68): icmp_seq=2 ttl=117 time=13.1 ms
64 bytes from kul01s09-in-f68.1e100.net (216.58.196.68): icmp_seq=3 ttl=117 time=9.47 ms
64 bytes from kul01s09-in-f68.1e100.net (216.58.196.68): icmp_seq=4 ttl=117 time=14.6 ms
64 bytes from kul01s09-in-f68.1e100.net (216.58.196.68): icmp_seq=5 ttl=117 time=8.68 ms
^C
--- www.google.com ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4026ms
rtt min/avg/max/mdev = 8.679/11.453/14.591/2.453 ms
```

16) sudo

Sudo (Super User DO) command in Linux is generally used as a prefix for some commands that only superusers are allowed to run.

Syntax: *sudo -V | -h | -l | -v | -k | -K | -s | [-H] [-P] [-S] [-b] |*

[-p prompt] [-c class|-] [-a auth_type] [-r role] [-t type]

[-u username|#uid] command

```
sakshigangurde711@Sakshi:/mnt/e$ sudo -l
[sudo] password for sakshigangurde711:
Matching Defaults entries for sakshigangurde711 on Sakshi:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/sn
ap/bin,
    use_pty

User sakshigangurde711 may run the following commands on Sakshi:
    (ALL : ALL) ALL
```

```
sakshigangurde711@Sakshi:/mnt/e$ sudo -h
sudo - execute a command as another user

usage: sudo -h | -K | -k | -V
usage: sudo -v [-ABknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-ABknS] [-g group] [-h host] [-p prompt] [-U user] [-u
            user] [command]
usage: sudo [-ABbEHknPS] [-r role] [-t type] [-C num] [-D directory] [-g
            group] [-h host] [-p prompt] [-R directory] [-T timeout] [-u
            user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-ABknS] [-r role] [-t type] [-C num] [-D directory] [-g
            group] [-h host] [-p prompt] [-R directory] [-T timeout] [-u
            user] file ...
```

17) df

Disk free also known as `df`, which is a powerful utility that provides valuable information on disk space utilization. The df command displays information about file system disk space usage on the mounted file system.

Syntax: *df [options] [filesystems]*