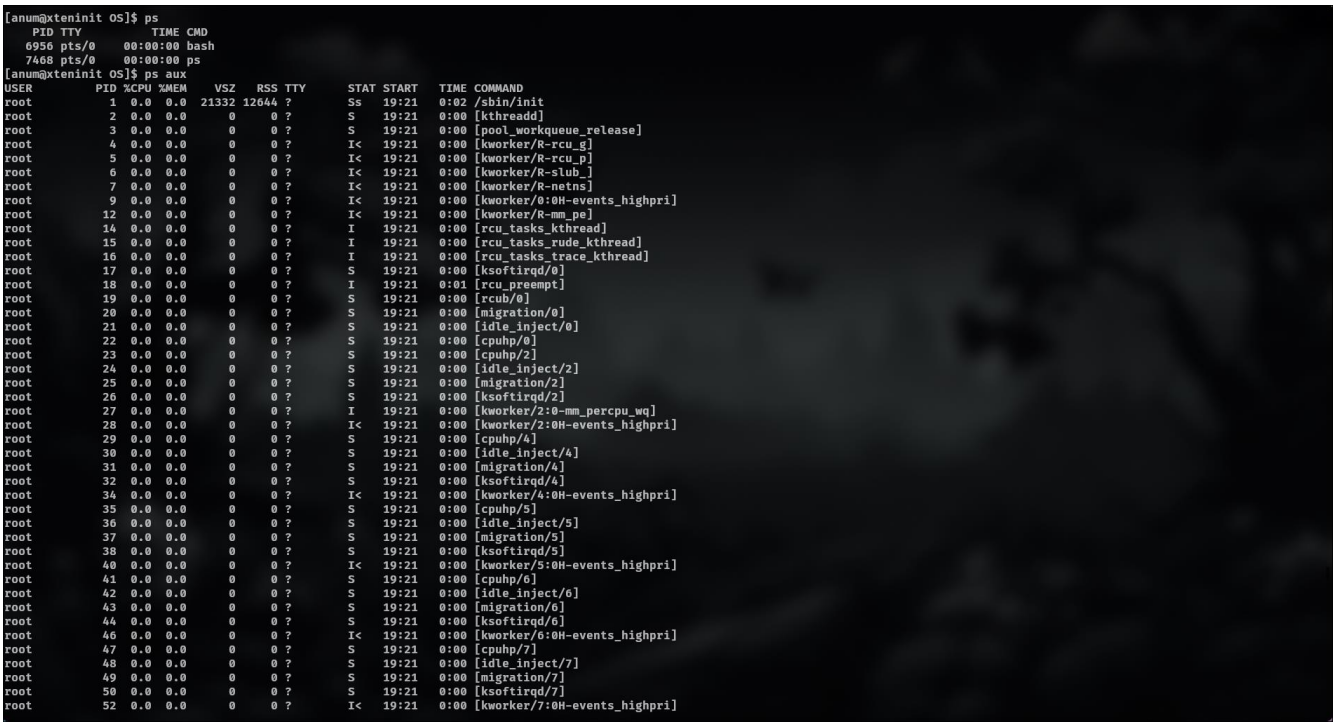
Name: Anum Asghar
Roll No: BCF22M012
Question 01:

ps: Displays current running processes.

Examples:

ps aux: Show all processes.
ps -ef: Show processes in full format.
ps -u username: Show processes for a specific user.



```
[anum@xteninit OS]$ ps
    PID TTY          TIME CMD
   6956 pts/0    00:00:00 bash
   7468 pts/0    00:00:00 ps
[anum@xteninit OS]$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0  21332 12644 ?        Ss   19:21   0:02 /sbin/init
root           2  0.0  0.0      0     0 ?        S    19:21   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    19:21   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   19:21   0:00 [kworker/R-rcu_g]
root           5  0.0  0.0      0     0 ?        I<   19:21   0:00 [kworker/R-rcu_p]
root           6  0.0  0.0      0     0 ?        I<   19:21   0:00 [kworker/R-slub_]
root           7  0.0  0.0      0     0 ?        I<   19:21   0:00 [kworker/R-netns]
root           9  0.0  0.0      0     0 ?        I<   19:21   0:00 [kworker/0:0H-events_highpri]
root          12  0.0  0.0      0     0 ?        I<   19:21   0:00 [kworker/R-mm_pe]
root          14  0.0  0.0      0     0 ?        I    19:21   0:00 [rcu_tasks_kthread]
root          15  0.0  0.0      0     0 ?        I    19:21   0:00 [rcu_tasks_rude_kthread]
root          16  0.0  0.0      0     0 ?        I    19:21   0:00 [rcu_tasks_trace_kthread]
root          17  0.0  0.0      0     0 ?        S    19:21   0:00 [ksoftirqd/0]
root          18  0.0  0.0      0     0 ?        I    19:21   0:01 [rcu_preempt]
root          19  0.0  0.0      0     0 ?        S    19:21   0:00 [rcub/0]
root          20  0.0  0.0      0     0 ?        S    19:21   0:00 [migration/0]
root          21  0.0  0.0      0     0 ?        S    19:21   0:00 [idle_inject/0]
root          22  0.0  0.0      0     0 ?        S    19:21   0:00 [cpuhp/0]
root          23  0.0  0.0      0     0 ?        S    19:21   0:00 [cpuhp/2]
root          24  0.0  0.0      0     0 ?        S    19:21   0:00 [idle_inject/2]
root          25  0.0  0.0      0     0 ?        S    19:21   0:00 [migration/2]
root          26  0.0  0.0      0     0 ?        S    19:21   0:00 [ksoftirqd/2]
root          27  0.0  0.0      0     0 ?        I    19:21   0:00 [kworker/2:0-mm_percpu_wq]
root          28  0.0  0.0      0     0 ?        I<   19:21   0:00 [kworker/2:0H-events_highpri]
root          29  0.0  0.0      0     0 ?        S    19:21   0:00 [cpuhp/4]
root          30  0.0  0.0      0     0 ?        S    19:21   0:00 [idle_inject/4]
root          31  0.0  0.0      0     0 ?        S    19:21   0:00 [migration/4]
root          32  0.0  0.0      0     0 ?        S    19:21   0:00 [ksoftirqd/4]
root          34  0.0  0.0      0     0 ?        I<   19:21   0:00 [kworker/4:0H-events_highpri]
root          35  0.0  0.0      0     0 ?        S    19:21   0:00 [cpuhp/5]
root          36  0.0  0.0      0     0 ?        S    19:21   0:00 [idle_inject/5]
root          37  0.0  0.0      0     0 ?        S    19:21   0:00 [migration/5]
root          38  0.0  0.0      0     0 ?        S    19:21   0:00 [ksoftirqd/5]
root          40  0.0  0.0      0     0 ?        I<   19:21   0:00 [kworker/5:0H-events_highpri]
root          41  0.0  0.0      0     0 ?        S    19:21   0:00 [cpuhp/6]
root          42  0.0  0.0      0     0 ?        S    19:21   0:00 [idle_inject/6]
root          43  0.0  0.0      0     0 ?        S    19:21   0:00 [migration/6]
root          44  0.0  0.0      0     0 ?        S    19:21   0:00 [ksoftirqd/6]
root          46  0.0  0.0      0     0 ?        I<   19:21   0:00 [kworker/6:0H-events_highpri]
root          47  0.0  0.0      0     0 ?        S    19:21   0:00 [cpuhp/7]
root          48  0.0  0.0      0     0 ?        S    19:21   0:00 [idle_inject/7]
root          49  0.0  0.0      0     0 ?        S    19:21   0:00 [migration/7]
root          50  0.0  0.0      0     0 ?        S    19:21   0:00 [ksoftirqd/7]
root          52  0.0  0.0      0     0 ?        I<   19:21   0:00 [kworker/7:0H-events_highpri]
```

```
[anum@xteninit OS]$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 19:21 ?        00:00:02 /sbin/init
root           2       0  0 19:21 ?        00:00:00 [kthreadd]
root           3       2  0 19:21 ?        00:00:00 [pool_workqueue_release]
root           4       2  0 19:21 ?        00:00:00 [kworker/R-rcu_g]
root           5       2  0 19:21 ?        00:00:00 [kworker/R-rcu_p]
root           6       2  0 19:21 ?        00:00:00 [kworker/R-slub_]
root           7       2  0 19:21 ?        00:00:00 [kworker/R-netns]
root           9       2  0 19:21 ?        00:00:00 [kworker/0:0H-events_highpri]
root          12       2  0 19:21 ?        00:00:00 [kworker/R-mm_pe]
root          14       2  0 19:21 ?        00:00:00 [rcu_tasks_kthread]
root          15       2  0 19:21 ?        00:00:00 [rcu_tasks_rude_kthread]
root          16       2  0 19:21 ?        00:00:00 [rcu_tasks_trace_kthread]
root          17       2  0 19:21 ?        00:00:00 [ksoftirqd/0]
root          18       2  0 19:21 ?        00:00:01 [rcu_preempt]
root          19       2  0 19:21 ?        00:00:00 [rcub/0]
root          20       2  0 19:21 ?        00:00:00 [migration/0]
root          21       2  0 19:21 ?        00:00:00 [idle_inject/0]
root          22       2  0 19:21 ?        00:00:00 [cpuhp/0]
root          23       2  0 19:21 ?        00:00:00 [cpuhp/2]
root          24       2  0 19:21 ?        00:00:00 [idle_inject/2]
root          25       2  0 19:21 ?        00:00:00 [migration/2]
root          26       2  0 19:21 ?        00:00:00 [ksoftirqd/2]
root          27       2  0 19:21 ?        00:00:00 [kworker/2:0-events]
root          28       2  0 19:21 ?        00:00:00 [kworker/2:0H-events_highpri]
root          29       2  0 19:21 ?        00:00:00 [cpuhp/4]
root          30       2  0 19:21 ?        00:00:00 [idle_inject/4]
root          31       2  0 19:21 ?        00:00:00 [migration/4]
root          32       2  0 19:21 ?        00:00:00 [ksoftirqd/4]
root          34       2  0 19:21 ?        00:00:00 [kworker/4:0H-events_highpri]
root          35       2  0 19:21 ?        00:00:00 [cpuhp/5]
root          36       2  0 19:21 ?        00:00:00 [idle_inject/5]
root          37       2  0 19:21 ?        00:00:00 [migration/5]
root          38       2  0 19:21 ?        00:00:00 [ksoftirqd/5]
root          40       2  0 19:21 ?        00:00:00 [kworker/5:0H-events_highpri]
root          41       2  0 19:21 ?        00:00:00 [cpuhp/6]
root          42       2  0 19:21 ?        00:00:00 [idle_inject/6]
root          43       2  0 19:21 ?        00:00:00 [migration/6]
root          44       2  0 19:21 ?        00:00:00 [ksoftirqd/6]
root          46       2  0 19:21 ?        00:00:00 [kworker/6:0H-events_highpri]
root          47       2  0 19:21 ?        00:00:00 [cpuhp/7]
root          48       2  0 19:21 ?        00:00:00 [idle_inject/7]
root          49       2  0 19:21 ?        00:00:00 [migration/7]
root          50       2  0 19:21 ?        00:00:00 [ksoftirqd/7]
root          52       2  0 19:21 ?        00:00:00 [kworker/7:0H-events_highpri]
root          53       2  0 19:21 ?        00:00:00 [cpuhp/8]
root          54       2  0 19:21 ?        00:00:00 [idle_inject/8]
root          55       2  0 19:21 ?        00:00:00 [migration/8]
root          56       2  0 19:21 ?        00:00:00 [ksoftirqd/8]
```

job: Lists active jobs in the current shell session.

Examples:

jobs: List all jobs.
jobs -l: List jobs with process IDs.
jobs -n: List only jobs that have changed status.

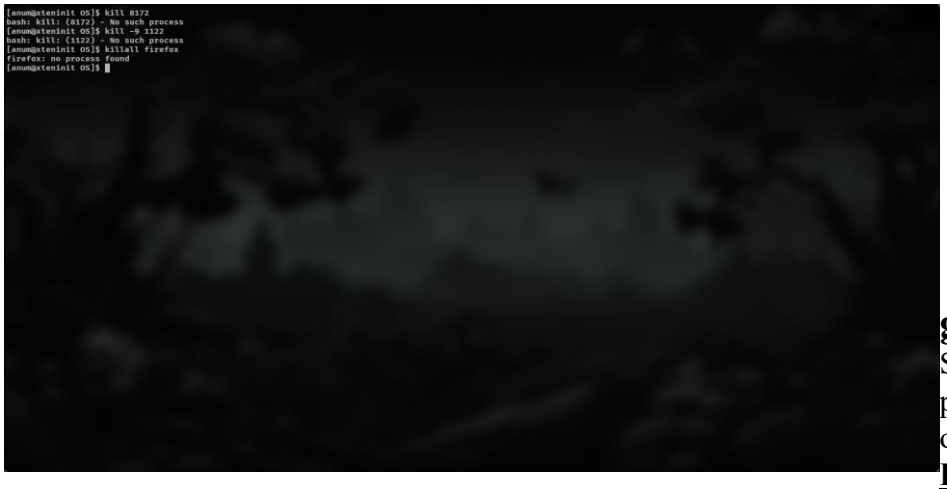
```
[anum@xteninit OS]$ sleep 30 &
[1] 8147
[anum@xteninit OS]$ sleep 30 &
[2] 8150
[anum@xteninit OS]$ sleep 30 &
[3] 8152
[anum@xteninit OS]$ jobs
[1]   Running                 sleep 30 &
[2]-  Running                 sleep 30 &
[3]+  Running                 sleep 30 &
[anum@xteninit OS]$
```

kill: Sends a signal to terminate a process.

Examples:

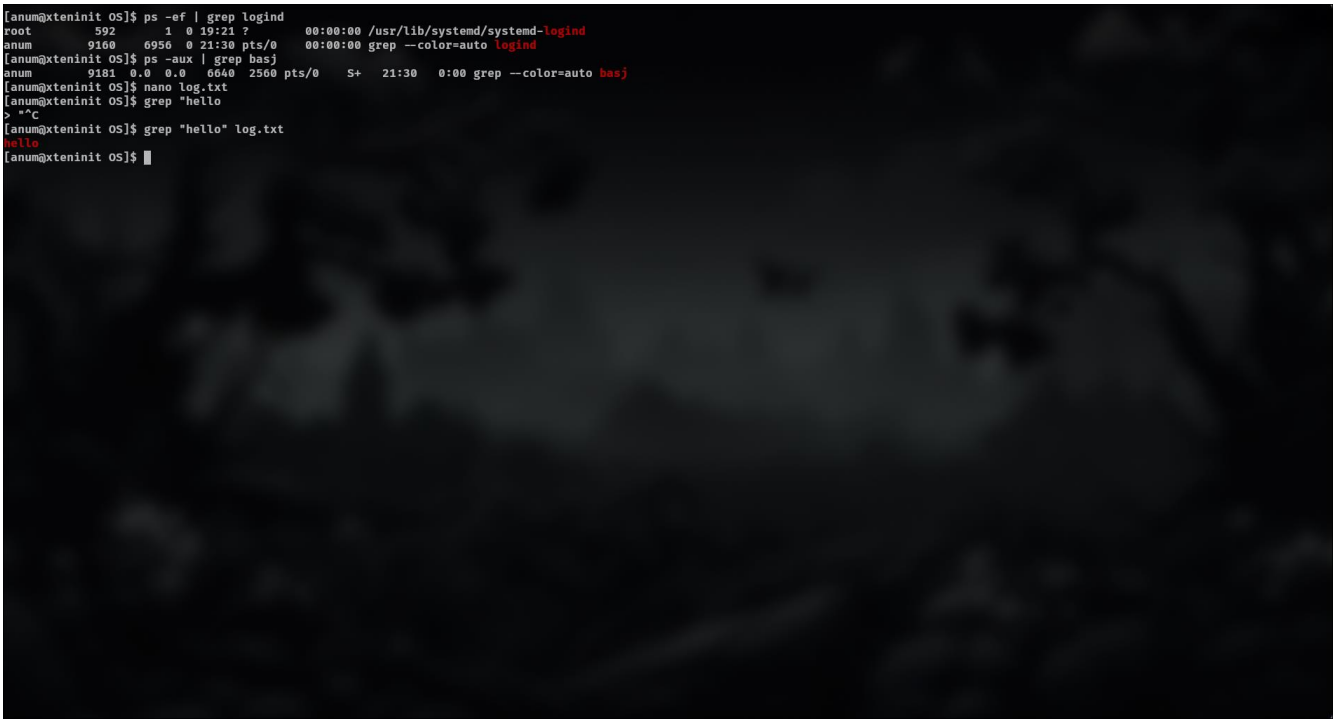
kill PID: Terminate process by ID.

kill -9 PID: Forcefully kill a process (SIGKILL).
killall process_name: Kill all processes with that name.



grep: Searches for patterns in files or input.
Examples:

grep 'pattern' file.txt: Search for 'pattern' in file.txt.
grep -i 'pattern' file.txt: Case-insensitive search.
grep -r 'pattern' /directory: Recursively search in directory.



gzip: Compresses files.
Examples:
gzip file.txt: Compress file.txt.
gzip -k file.txt: Compress and keep original.
gzip -d file.txt.gz: Decompress file.gz.

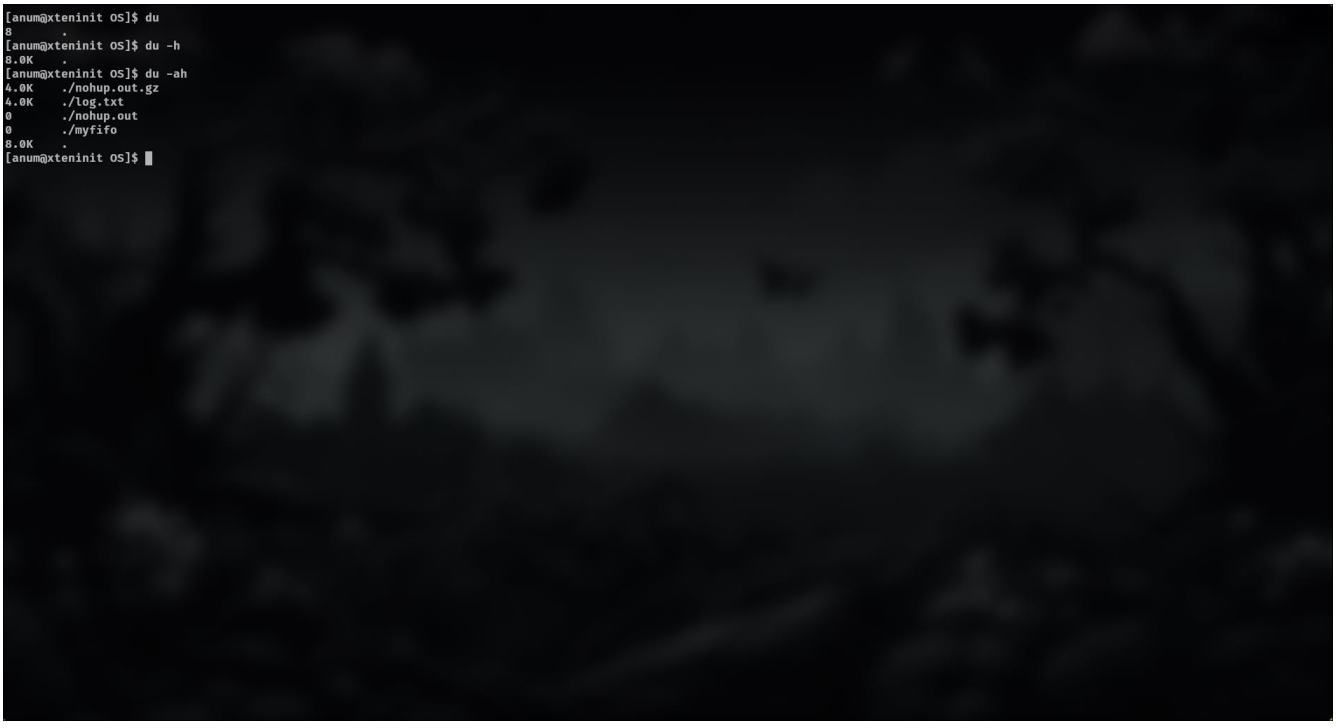
gunzip: Decompresses gzip files.
Examples:
gunzip file.gz: Decompress file.gz.
gunzip -k file.gz: Decompress and keep original.
gunzip *.gz: Decompress all .gz files in current directory.

fifo: Named pipes for inter-process communication.
Examples:
mkfifo myfifo: Create a FIFO named myfifo.
echo "data" > myfifo: Write data to FIFO.
cat < myfifo: Read data from FIFO.
mkfifo: Command to create a FIFO special file.

du: Displays disk usage of files and directories.

Examples:

du -h: Human-readable format.
du -s: Summarize total for each argument.
du -a: Show sizes of all files.



chmod: Changes file permissions.
Mode format: chmod [who][operation][permissions]

who: u (user), g (group), o (others), a (all)
operation: + (add), - (remove), = (set exact)
permissions: r (read), w (write), x (execute)
Examples:
chmod u+x file: Add execute permission for user.
chmod go-w file: Remove write permission for group and others.
chmod 755 file: Set permissions to rwxr-xr-x.

What is a mount point in Linux? What is the purpose of the /mnt folder?
A mount point is a directory in the filesystem where additional filesystems are attached. The **/mnt** folder is typically used for temporarily mounting filesystems.

What is the purpose of /etc and /proc folders?
/**etc:** Contains configuration files for the system and applications.
/proc: A virtual filesystem providing process and system information.

Write a command to display the length of the file or directory having the longest name in the current directory.
Command:

```
ls | awk '{print length, $0}' | sort -n | tail -1
```

Write a command to display names of files in the current directory which contains the word "BCS F24 M".

```
grep -l "BCS F24 M" *
```

What are command line arguments in C/C++?
Write a program to get two numbers from user at command line.

Command line arguments in C/C++ are inputs provided to a program when it is executed from the command line. They allow users to pass parameters to the program, which can be accessed via the argc (argument count) and argv (argument vector) parameters in the main function.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 3) {
        printf("Usage: %s <number1> <number2>\n", argv[0]);
        return 1;
    }

    int num1 = atoi(argv[1]);
    int num2 = atoi(argv[2]);
    int sum = num1 + num2;

    printf("The sum of %d and %d is: %d\n", num1, num2, sum);

    return 0;
}
```

Run this program::

```
gcc -o sum sum.c
./sum 5 10
```