1.Adam is working in an IT company. He has been given a task to reduce the load of a system by killing some of the processes running in the LINUX operating system. Which commands will he use to complete the given task with the help of the following operation?

(i) Kill processes by name
(ii) Kill a process based on the process name
(iii) Kill a single process at a time with the given process ID

CODE:

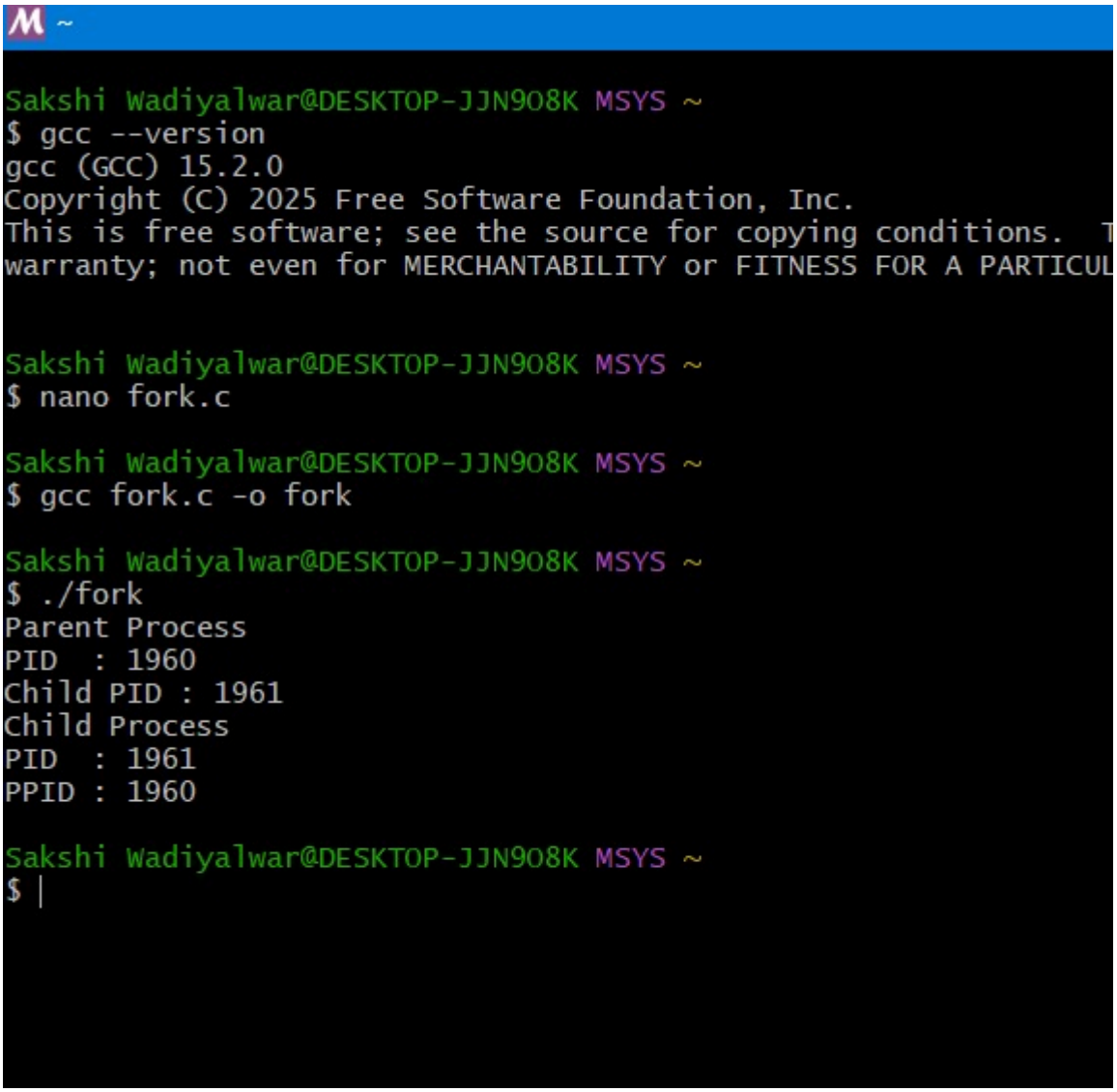
```
~
  GNU nano 8.7
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

int main() {
    pid_t pid;

    pid = fork();

    if (pid < 0) {
        printf("Fork failed\n");
    }
    else if (pid == 0) {
        // Child process
        printf("Child Process\n");
        printf("PID  : %d\n", getpid());
        printf("PPID : %d\n", getppid());
    }
    else {
        // Parent process
        printf("Parent Process\n");
        printf("PID  : %d\n", getpid());
        printf("Child PID : %d\n", pid);
        wait(NULL);
    }
    return 0;
}
```

OUTPUT:


~
Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ gcc --version
gcc (GCC) 15.2.0
Copyright (C) 2025 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICUL
Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ nano fork.c
Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ gcc fork.c -o fork
Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ ./fork
Parent Process
PID : 1960
Child PID : 1961
Child Process
PID : 1961
PPID : 1960
Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$

2. Write a program for process creation using C

(i) Orphan Process
(ii) Zombine Process

CODE:

```
  GNU nano 8.7
#include <stdio.h>
#include <unistd.h>

int main() {
    pid_t pid = fork();

    if (pid == 0) {
        sleep(5);
        printf("Child Process\n");
        printf("PID  : %d\n", getpid());
        printf("PPID : %d\n", getppid());
    }
    else {
        printf("Parent exiting\n");
    }
    return 0;
}
```

OUTPUT:

```
~

Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ nano orphan.c

Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ gcc orphan.c -o orphan

Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ ./orphan
Parent exiting

Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ Child Process
PID  : 1970
PPID : 1
```

3. Create the process using fork () system call.

(i) Child Process creation
(ii) Parent process creation
(iii) PPID and PID

CODE:

```
~
  GNU nano 8.7
#include <stdio.h>
#include <unistd.h>

int main() {
    pid_t pid = fork();

    if (pid == 0) {
        printf("Child exiting\n");
    }
    else {
        sleep(10);
        printf("Parent running\n");
    }
    return 0;
}
```

OUTPUT:

```
Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ nano zombie.c

Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ gcc zombie.c -o zombie

Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ ./zombie
Child exiting
Parent running

Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$ ps -el
      PID    PPID    PGID     WINPID   TTY         UID    STIME COMMAND
     1833       1    1833       1636  ?        197609 15:08:57 /usr/bin/mintty
     1980    1834    1980      16284  pty1     197609 16:06:38 /usr/bin/ps
     1834    1833    1834      10964  pty1     197609 15:08:57 /usr/bin/bash

Sakshi Wadiyalwar@DESKTOP-JJN9O8K MSYS ~
$
```