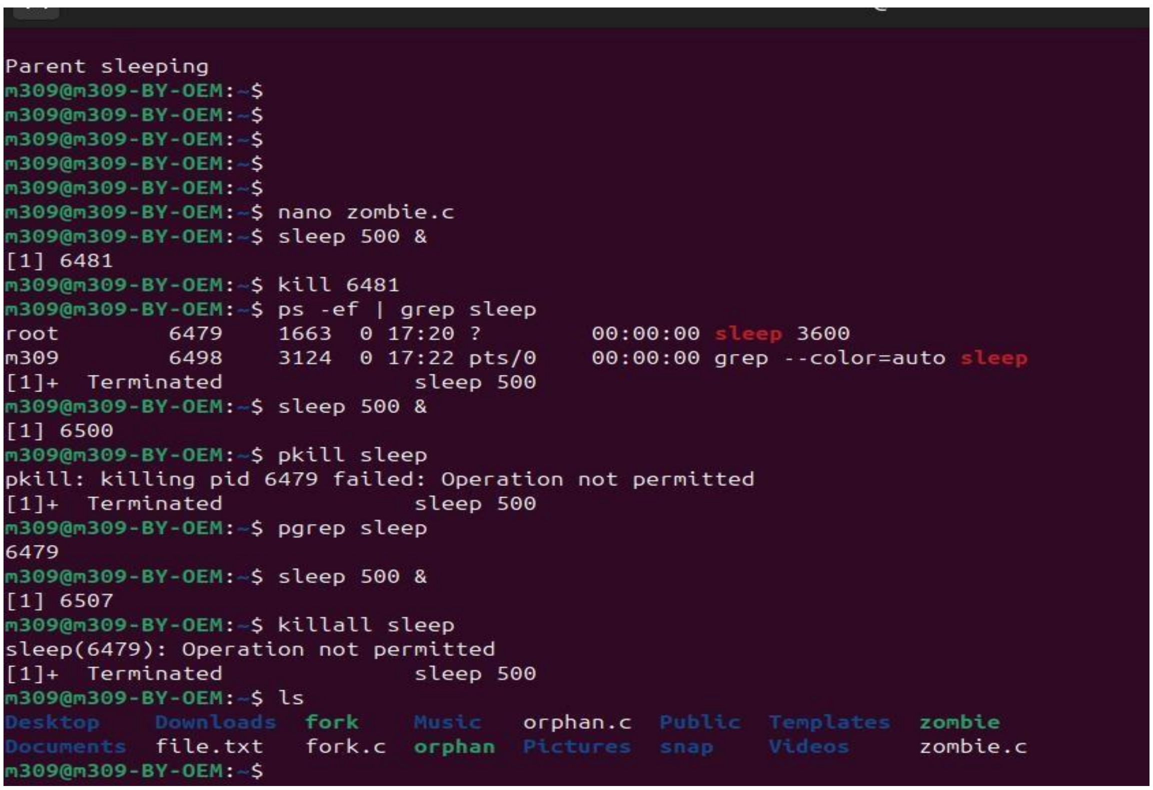
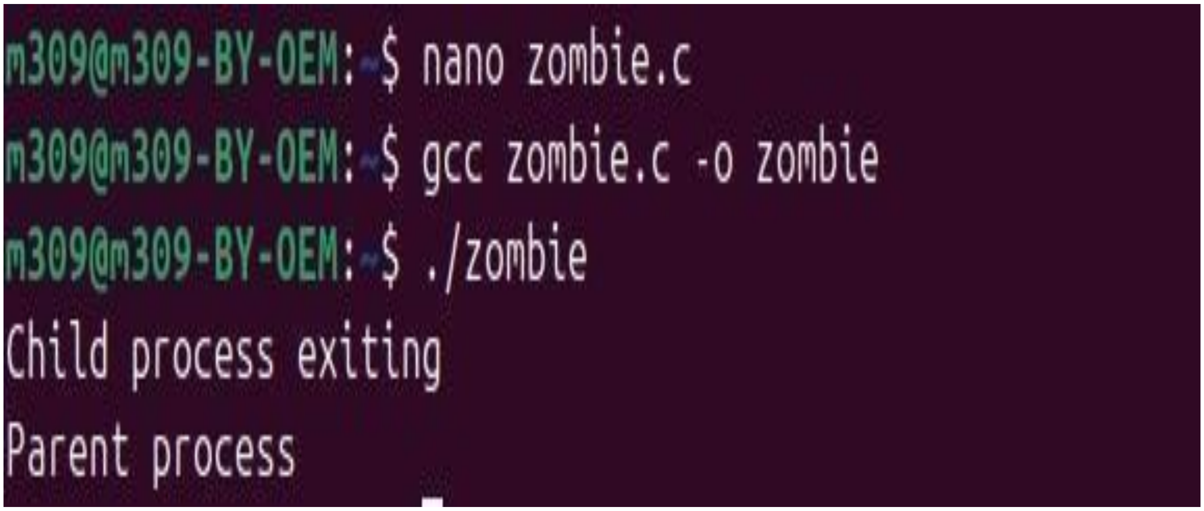
1. Adam is working in an IT company. He has been given a task to reduce the load of a system by killing some of the processes running in the LINUX operating system. Which commands will he use to complete the given task with the help of the following operation?

Kill Processes by name



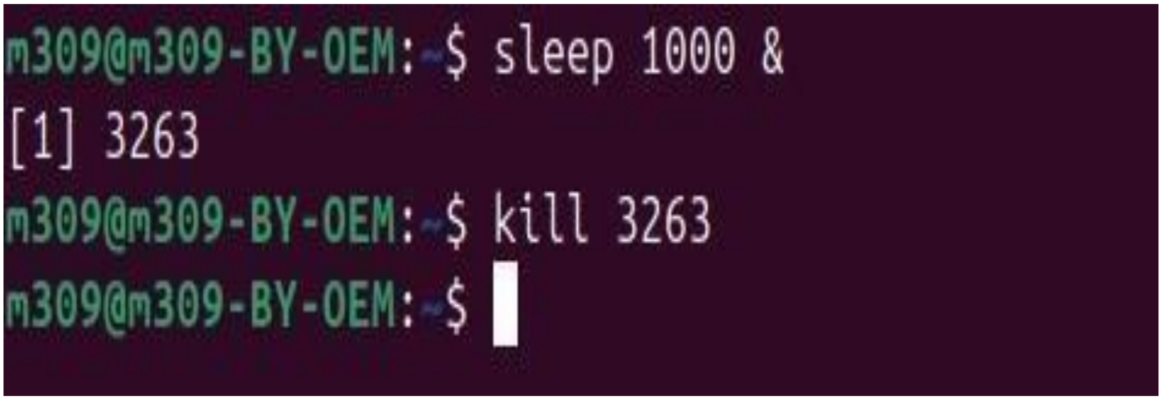
```
Parent sleeping
m309@m309-BY-OEM:~$
m309@m309-BY-OEM:~$
m309@m309-BY-OEM:~$
m309@m309-BY-OEM:~$
m309@m309-BY-OEM:~$
m309@m309-BY-OEM:~$ nano zombie.c
m309@m309-BY-OEM:~$ sleep 500 &
[1] 6481
m309@m309-BY-OEM:~$ kill 6481
m309@m309-BY-OEM:~$ ps -ef | grep sleep
root         6479    1663  0 17:20 ?        00:00:00 sleep 3600
m309         6498    3124  0 17:22 pts/0    00:00:00 grep --color=auto sleep
[1]+  Terminated              sleep 500
m309@m309-BY-OEM:~$ sleep 500 &
[1] 6500
m309@m309-BY-OEM:~$ pkill sleep
pkill: killing pid 6479 failed: Operation not permitted
[1]+  Terminated              sleep 500
m309@m309-BY-OEM:~$ pgrep sleep
6479
m309@m309-BY-OEM:~$ sleep 500 &
[1] 6507
m309@m309-BY-OEM:~$ killall sleep
sleep(6479): Operation not permitted
[1]+  Terminated              sleep 500
m309@m309-BY-OEM:~$ ls
Desktop    Downloads  fork     Music   orphan.c  Public  Templates  zombie
Documents  file.txt   fork.c   orphan  Pictures  snap    Videos     zombie.c
m309@m309-BY-OEM:~$
```

Kill a process based on the process name



```
m309@m309-BY-OEM:~$ nano zombie.c
m309@m309-BY-OEM:~$ gcc zombie.c -o zombie
m309@m309-BY-OEM:~$ ./zombie
Child process exiting
Parent process
```

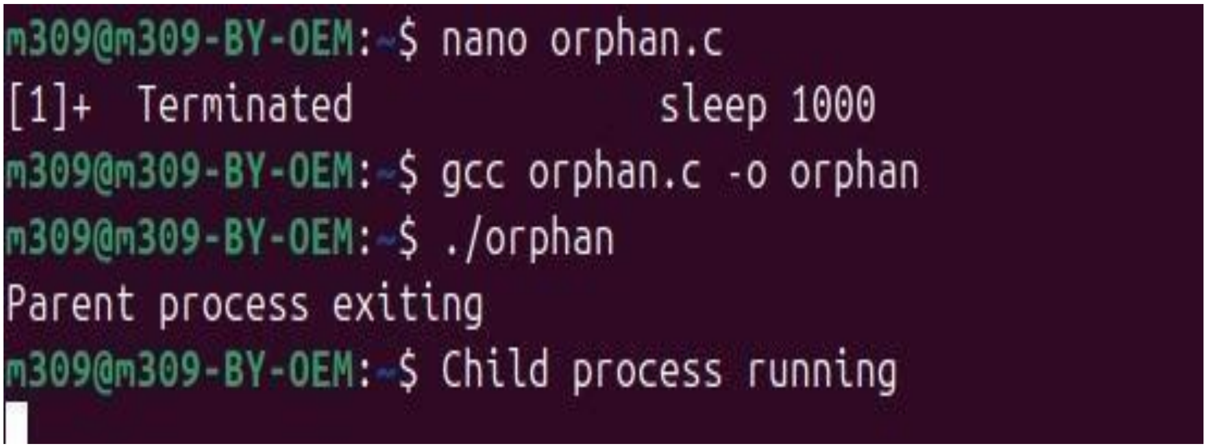
Kill a single process at a time with the given process ID



```
m309@m309-BY-OEM:~$ sleep 1000 &
[1] 3263
m309@m309-BY-OEM:~$ kill 3263
m309@m309-BY-OEM:~$
```

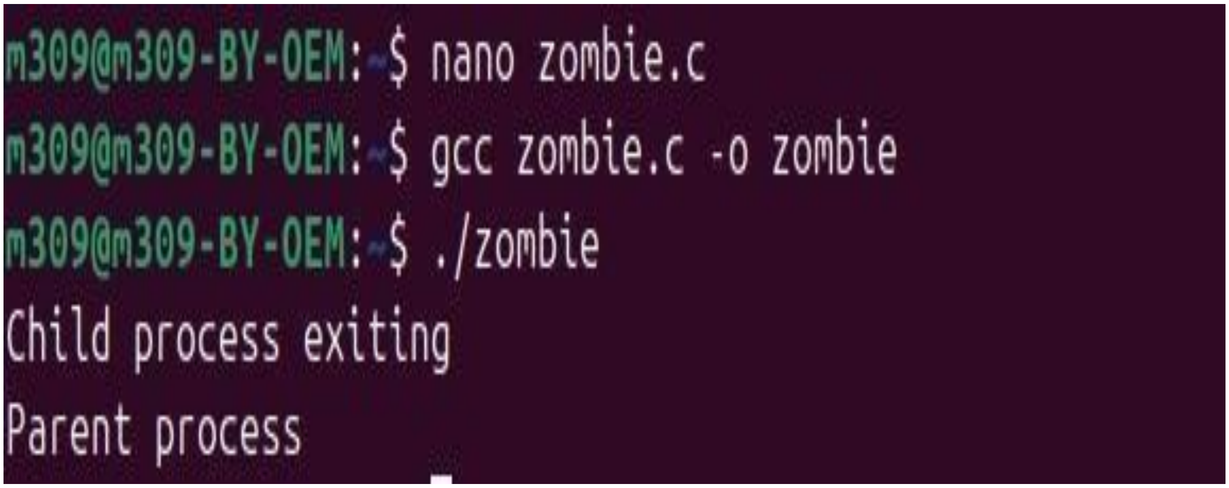
2.Write a program for process creation using C

Orphan Process



```
m309@m309-BY-OEM:~$ nano orphan.c
[1]+  Terminated              sleep 1000
m309@m309-BY-OEM:~$ gcc orphan.c -o orphan
m309@m309-BY-OEM:~$ ./orphan
Parent process exiting
m309@m309-BY-OEM:~$ Child process running
```

Zombie Process

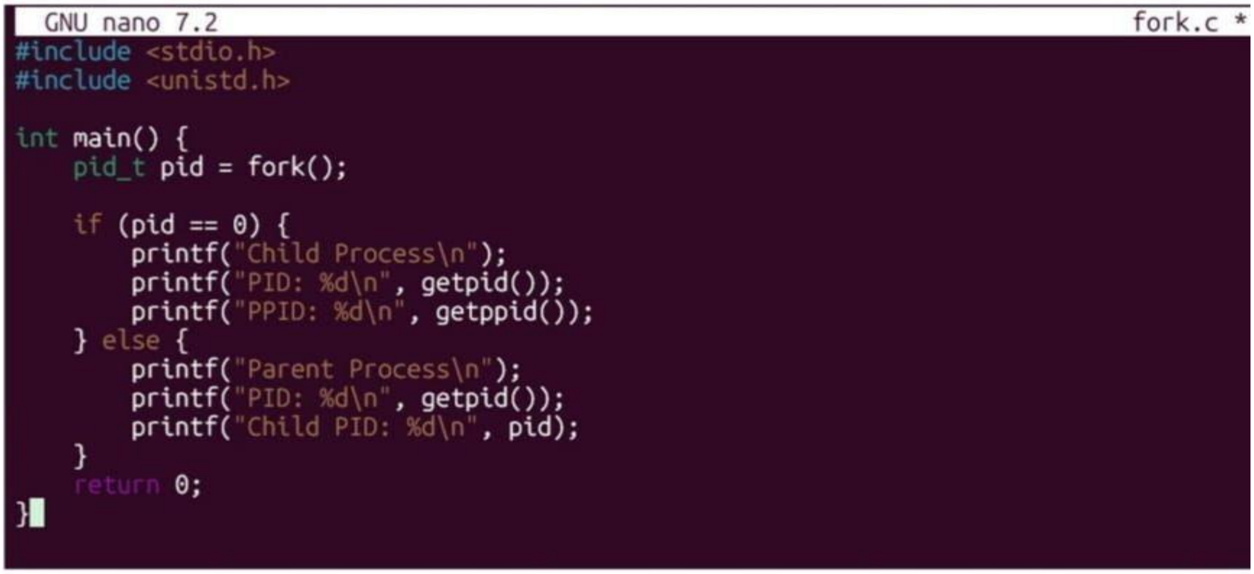


```
m309@m309-BY-OEM:~$ nano zombie.c
m309@m309-BY-OEM:~$ gcc zombie.c -o zombie
m309@m309-BY-OEM:~$ ./zombie
Child process exiting
Parent process
```

3. Create the process using fork () system call

Child Process creation

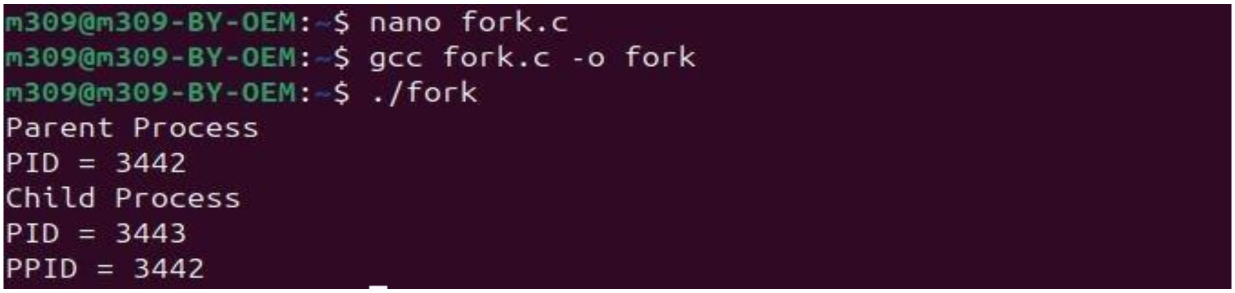
Parent Process creation PIDD and PID



```
  GNU nano 7.2                                                    fork.c *
#include <stdio.h>
#include <unistd.h>

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

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



```
m309@m309-BY-OEM:~$ nano fork.c
m309@m309-BY-OEM:~$ gcc fork.c -o fork
m309@m309-BY-OEM:~$ ./fork
Parent Process
PID = 3442
Child Process
PID = 3443
PPID = 3442
```