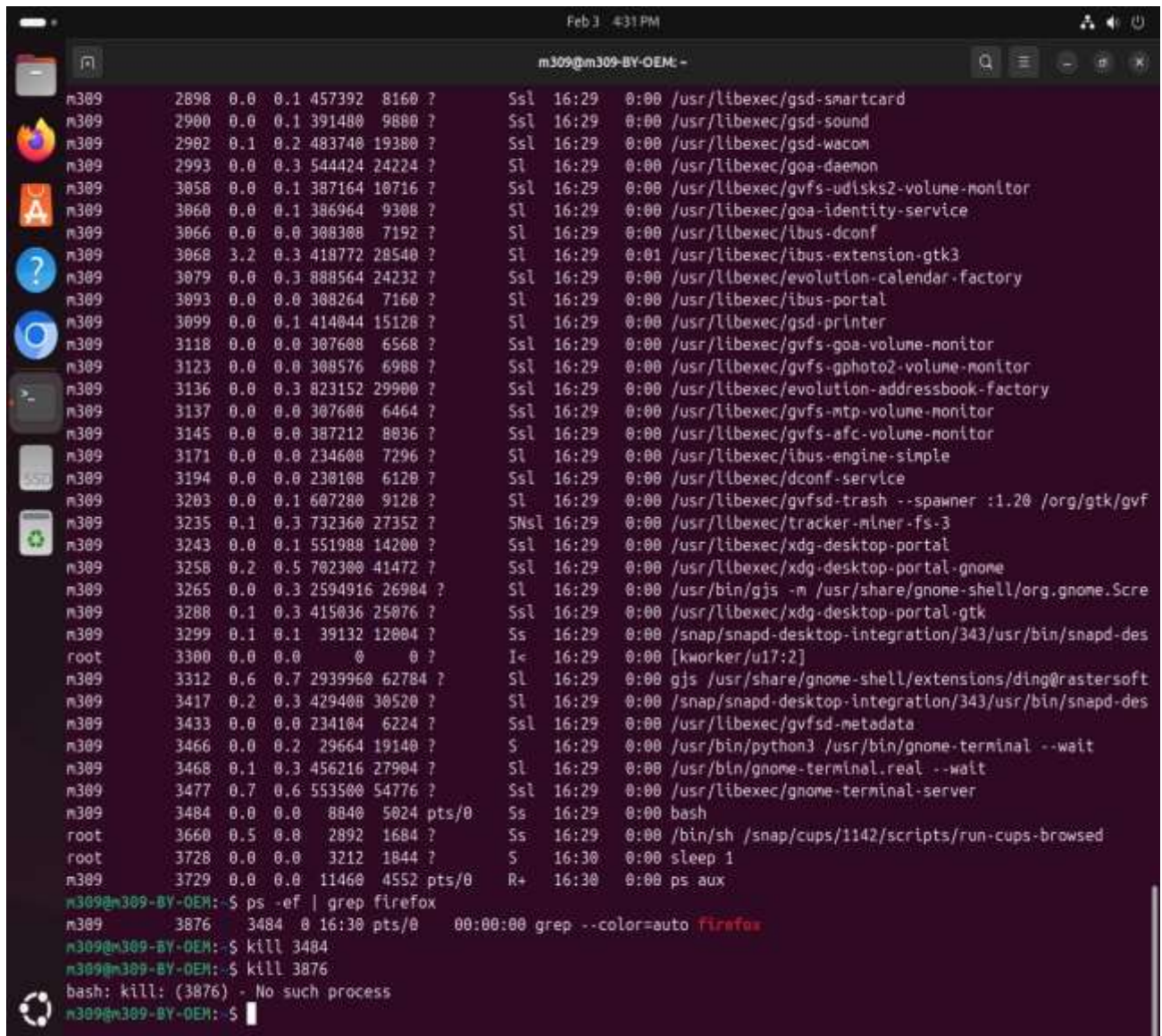# PRACTICAL4

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**

- **Kill a process based on the process name**

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

2.Write a program for process creation using C

• Orphan Process
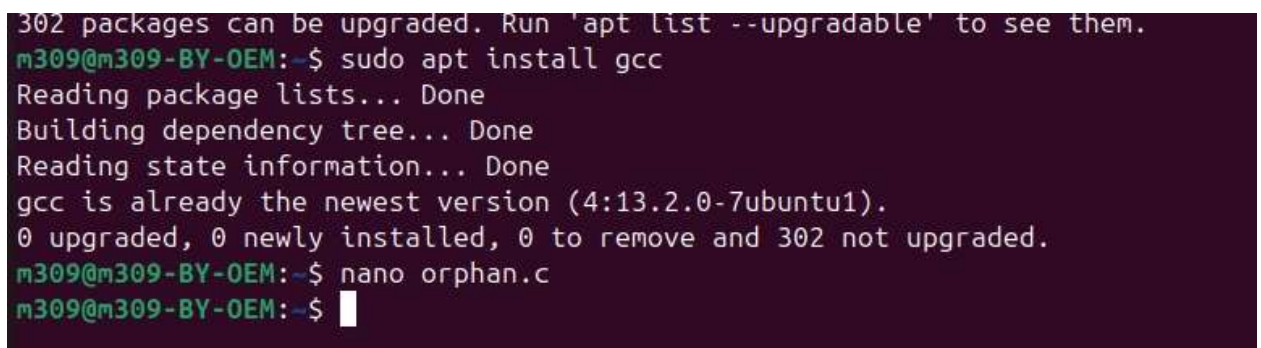

```c
#include <stdio.h>
#include <unistd.h>

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

    if (pid > 0) {
        printf("Parent process exiting\n");
    } else if (pid == 0) {
        sleep(10);
        printf("Child process running. Parent PID: %d\n", getppid());
    }
    return 0;
}
```

Output:


```
302 packages can be upgraded. Run 'apt list --upgradable' to see them.
m309@m309-BY-OEM:~$ sudo apt install gcc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc is already the newest version (4:13.2.0-7ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 302 not upgraded.
m309@m309-BY-OEM:~$ nano orphan.c
m309@m309-BY-OEM:~$
```

- **Zombie Process**


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

int main() {
    if (fork() == 0) {
        printf("Child exiting\n");
    } else {
        sleep(20);
        printf("Parent alive\n");
    }
    return 0;
}
```
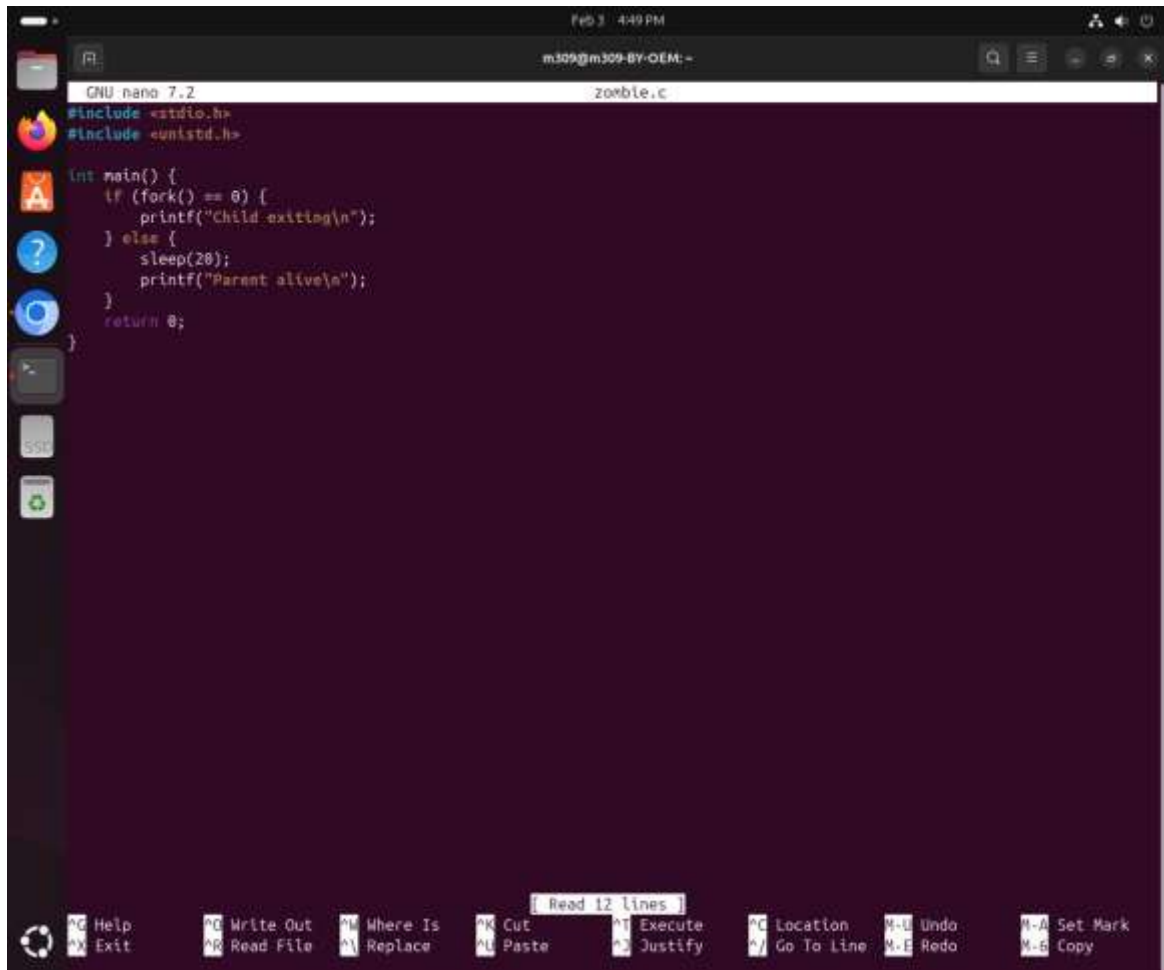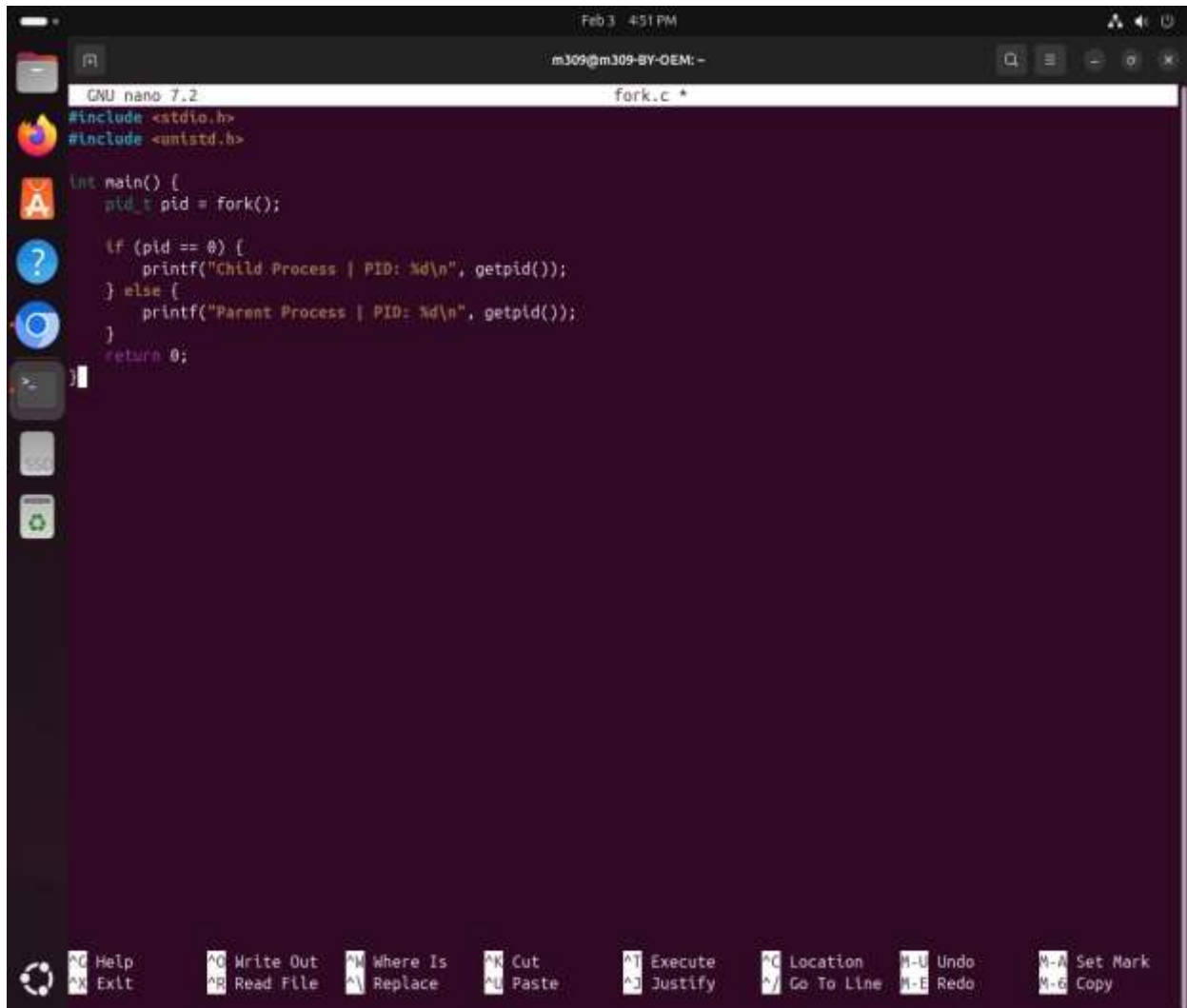
Output:


```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc is already the newest version (4:13.2.0-7ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 302 not upgraded.
m309@m309-BY-OEM:~$ nano orphan.c
m309@m309-BY-OEM:~$ nano orphan.c
m309@m309-BY-OEM:~$ nano zombie.c
m309@m309-BY-OEM:~$
```



```
m309@m309-BY-OEM:~$ nano zombie.c
m309@m309-BY-OEM:~$ ps -ef | grep z
m309      8223   8146  0 16:43 ?       00:00:00 /snap/chromium/3125/usr/lib/chromium-browser/chrome --type=zygote --
no-zygote-sandbox --crashpad-handler-pid=8217 --enable-crash-reporter=,snap --change-stack-guard-on-fork=enable
m309      8224   8146  0 16:43 ?       00:00:00 /snap/chromium/3125/usr/lib/chromium-browser/chrome --type=zygote --
crashpad-handler-pid=8217 --enable-crash-reporter=,snap --change-stack-guard-on-fork=enable
m309      8226   8224  0 16:43 ?       00:00:00 /snap/chromium/3125/usr/lib/chromium-browser/chrome --type=zygote --
crashpad-handler-pid=8217 --enable-crash-reporter=,snap --change-stack-guard-on-fork=enable
m309     10071   3484  0 16:50 pts/0   00:00:00 grep --color=auto z
m309@m309-BY-OEM:~$
```

3. Create the process using fork ()system call

- **Child Process creation**
- **Parent Process creation PPID and PID**


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

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

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

```
m309      8223   8146  0 16:43 ?        00:00:00 /snap/chromium/3125/usr/lib/chromium-browser/chrome --type=zygote --
no-zygote-sandbox --crashpad-handler-pid=8217 --enable-crash-reporter=,snap --change-stack-guard-on-fork=enable
m309      8224   8146  0 16:43 ?        00:00:00 /snap/chromium/3125/usr/lib/chromium-browser/chrome --type=zygote --
crashpad-handler-pid=8217 --enable-crash-reporter=,snap --change-stack-guard-on-fork=enable
m309      8226   8224  0 16:43 ?        00:00:00 /snap/chromium/3125/usr/lib/chromium-browser/chrome --type=zygote --
crashpad-handler-pid=8217 --enable-crash-reporter=,snap --change-stack-guard-on-fork=enable
m309     10071   3484  0 16:50 pts/0    00:00:00 grep --color=auto z
m309@m309-BY-OEM:-$ nano fork.c
m309@m309-BY-OEM:-$ nano fork.c
m309@m309-BY-OEM:-$
```