Name: Muhammad Usman
Roll no: 19P-0116

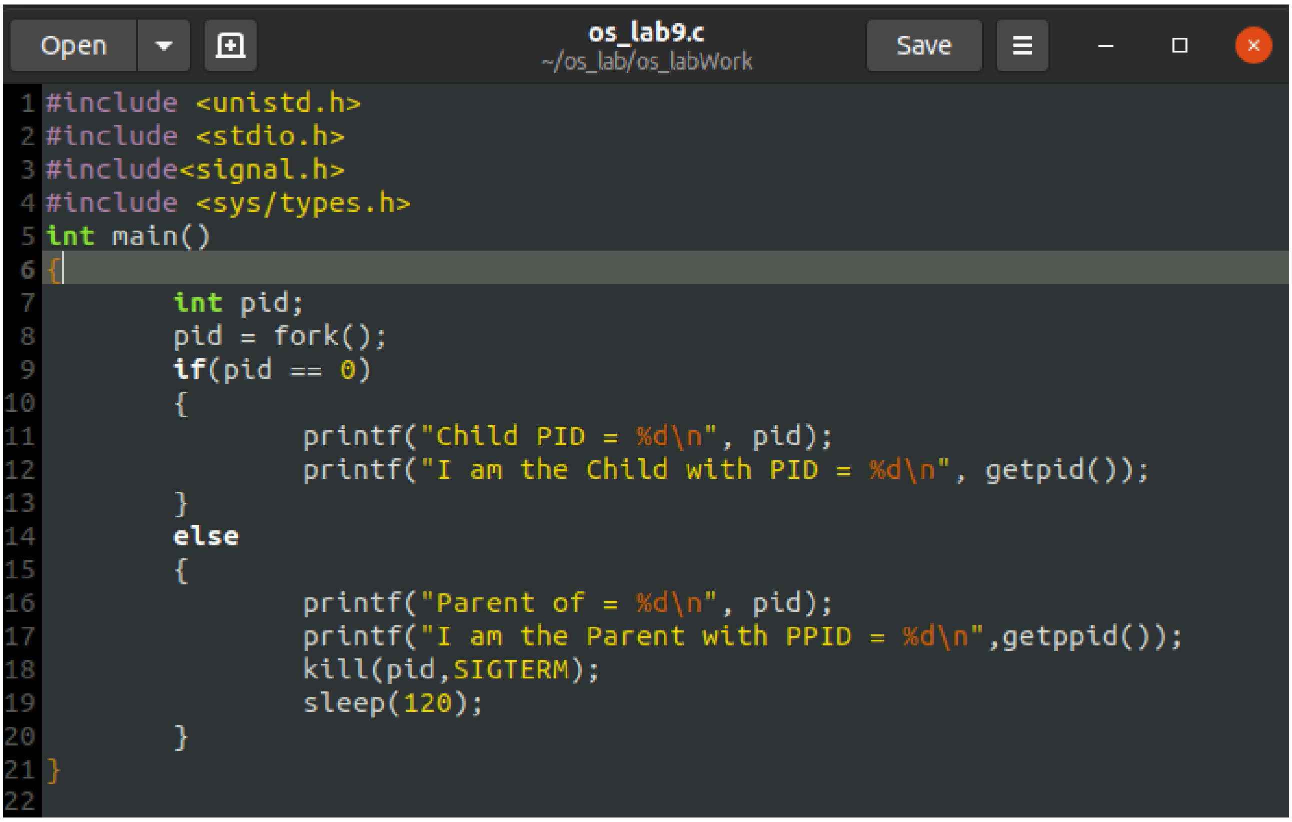
LAB TASK 9

Exercise # 01
Write code which does the following:

- Parent process creates a child process using fork() call (using proper if/else statement blocks).
- Parent sends the terminate signal via kill() call to child and then waits for 120 seconds.

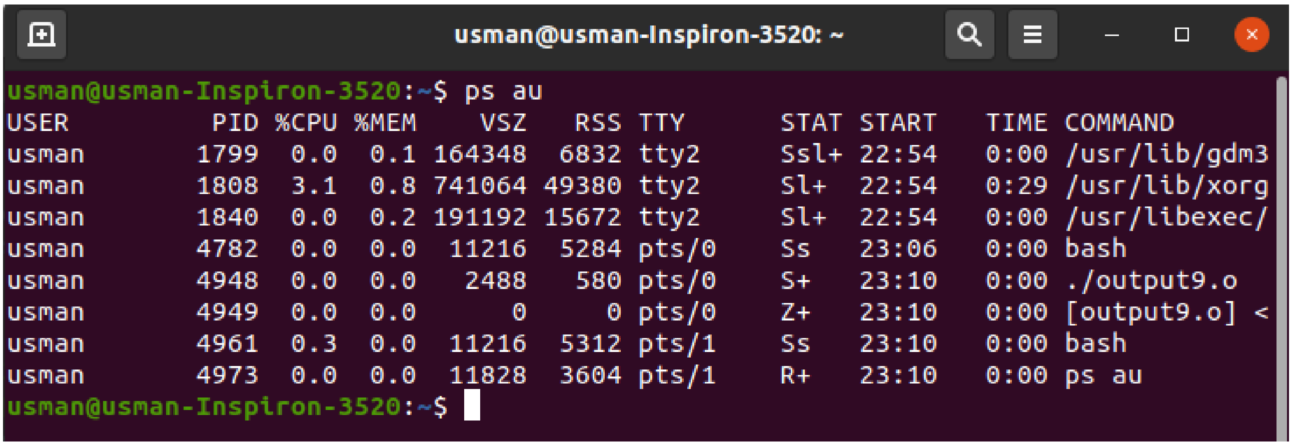
While parent is waiting, the user should check the outcome of **ps au** command.

CODE:

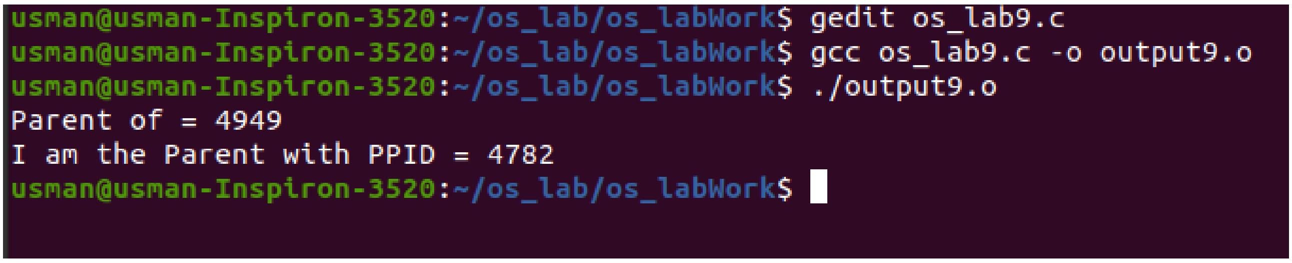


```
#include <unistd.h>
#include <stdio.h>
#include<signal.h>
#include <sys/types.h>
int main()
{
        int pid;
        pid = fork();
        if(pid == 0)
        {
                printf("Child PID = %d\n", pid);
                printf("I am the Child with PID = %d\n", getpid());
        }
        else
        {
                printf("Parent of = %d\n", pid);
                printf("I am the Parent with PPID = %d\n",getppid());
                kill(pid,SIGTERM);
                sleep(120);
        }
}
```

Output:



```
usman@usman-Inspiron-3520:~$ ps au
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
usman       1799  0.0  0.1 164348  6832 tty2     Ssl+ 22:54   0:00 /usr/lib/gdm3
usman       1808  3.1  0.8 741064 49380 tty2     Sl+  22:54   0:29 /usr/lib/xorg
usman       1840  0.0  0.2 191192 15672 tty2     Sl+  22:54   0:00 /usr/libexec/
usman       4782  0.0  0.0  11216  5284 pts/0    Ss   23:06   0:00 bash
usman       4948  0.0  0.0   2488   580 pts/0    S+   23:10   0:00 ./output9.o
usman       4949  0.0  0.0      0     0 pts/0    Z+   23:10   0:00 [output9.o] <
usman       4961  0.3  0.0  11216  5312 pts/1    Ss   23:10   0:00 bash
usman       4973  0.0  0.0  11828  3604 pts/1    R+   23:10   0:00 ps au
usman@usman-Inspiron-3520:~$ 
```



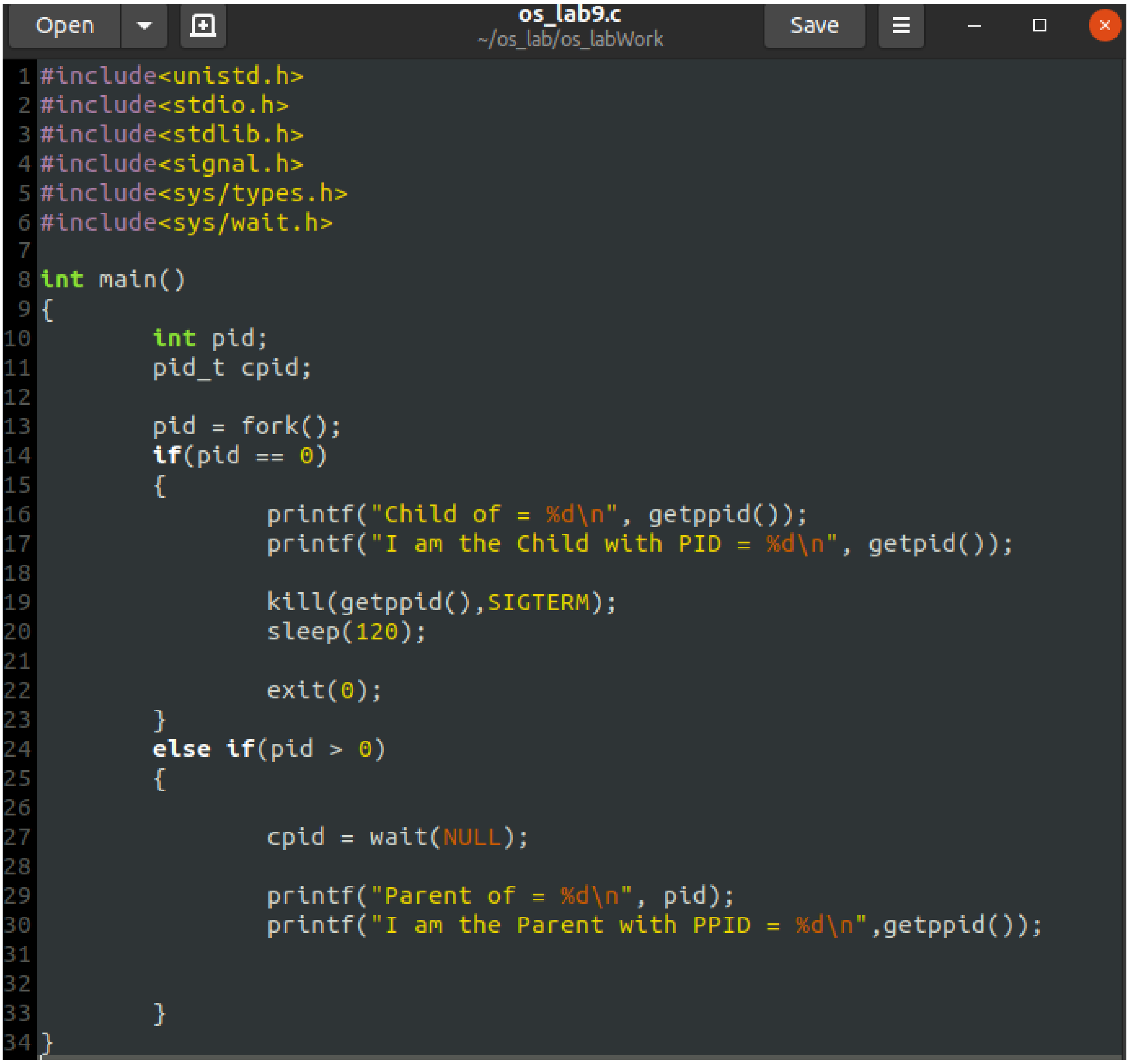
```
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ gedit os_lab9.c
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ gcc os_lab9.c -o output9.o
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ ./output9.o
Parent of = 4949
I am the Parent with PPID = 4782
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ 
```

Exercise # 02

Modify the code in **Exercise # 01** such that signal is sent from child to parent. and then have your child wait for 120 seconds.

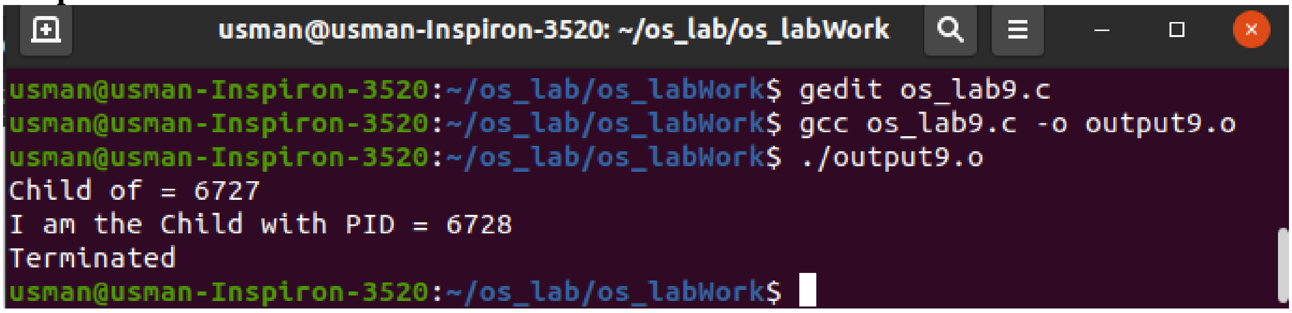
Again, while child is waiting, the user should check the outcome of **ps au** command.

CODE:

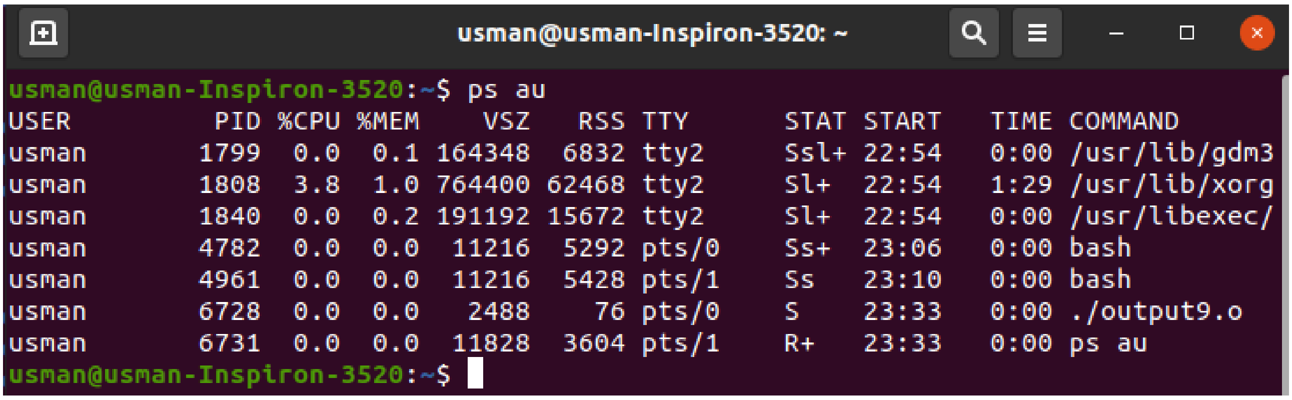


```c
#include<unistd.h>
#include<stdio.h>
#include<stdlib.h>
#include<signal.h>
#include<sys/types.h>
#include<sys/wait.h>

int main()
{
        int pid;
        pid_t cpid;

        pid = fork();
        if(pid == 0)
        {
                printf("Child of = %d\n", getppid());
                printf("I am the Child with PID = %d\n", getpid());

                kill(getppid(),SIGTERM);
                sleep(120);

                exit(0);
        }
        else if(pid > 0)
        {

                cpid = wait(NULL);

                printf("Parent of = %d\n", pid);
                printf("I am the Parent with PPID = %d\n",getppid());


        }
}
```

Output:



```
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ gedit os_lab9.c
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ gcc os_lab9.c -o output9.o
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ ./output9.o
Child of = 6727
I am the Child with PID = 6728
Terminated
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ 
```



```
usman@usman-Inspiron-3520:~$ ps au
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
usman       1799  0.0  0.1 164348  6832 tty2     Ssl+ 22:54   0:00 /usr/lib/gdm3
usman       1808  3.8  1.0 764400 62468 tty2     Sl+  22:54   1:29 /usr/lib/xorg
usman       1840  0.0  0.2 191192 15672 tty2     Sl+  22:54   0:00 /usr/libexec/
usman       4782  0.0  0.0  11216  5292 pts/0    Ss+  23:06   0:00 bash
usman       4961  0.0  0.0  11216  5428 pts/1    Ss   23:10   0:00 bash
usman       6728  0.0  0.0   2488    76 pts/0    S    23:33   0:00 ./output9.o
usman       6731  0.0  0.0  11828  3604 pts/1    R+   23:33   0:00 ps au
usman@usman-Inspiron-3520:~$ 
```