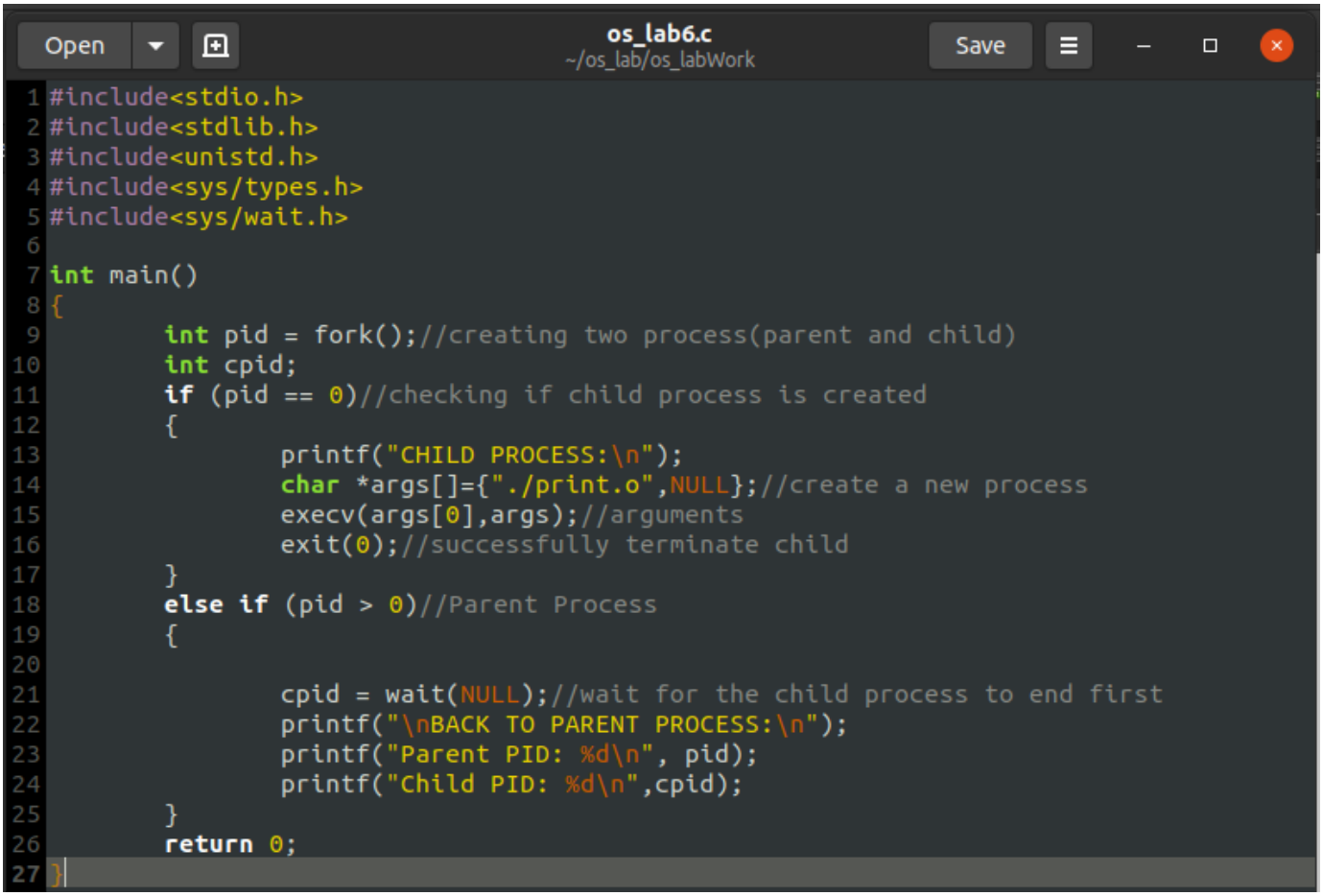
Name: Muhammad Usman
Roll no: 19P-0116
Section: A

LAB TASK 6

CODE:

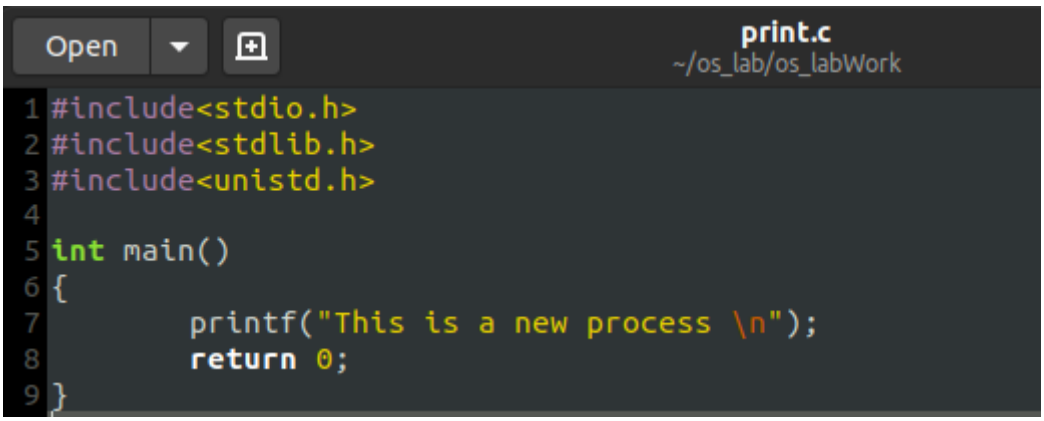



```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>

int main()
{
        int pid = fork();//creating two process(parent and child)
        int cpid;
        if (pid == 0)//checking if child process is created
        {
                printf("CHILD PROCESS:\n");
                char *args[]={"./print.o",NULL};//create a new process
                execv(args[0],args);//arguments
                exit(0);//successfully terminate child
        }
        else if (pid > 0)//Parent Process
        {

                cpid = wait(NULL);//wait for the child process to end first
                printf("\nBACK TO PARENT PROCESS:\n");
                printf("Parent PID: %d\n", pid);
                printf("Child PID: %d\n",cpid);
        }
        return 0;
}
```

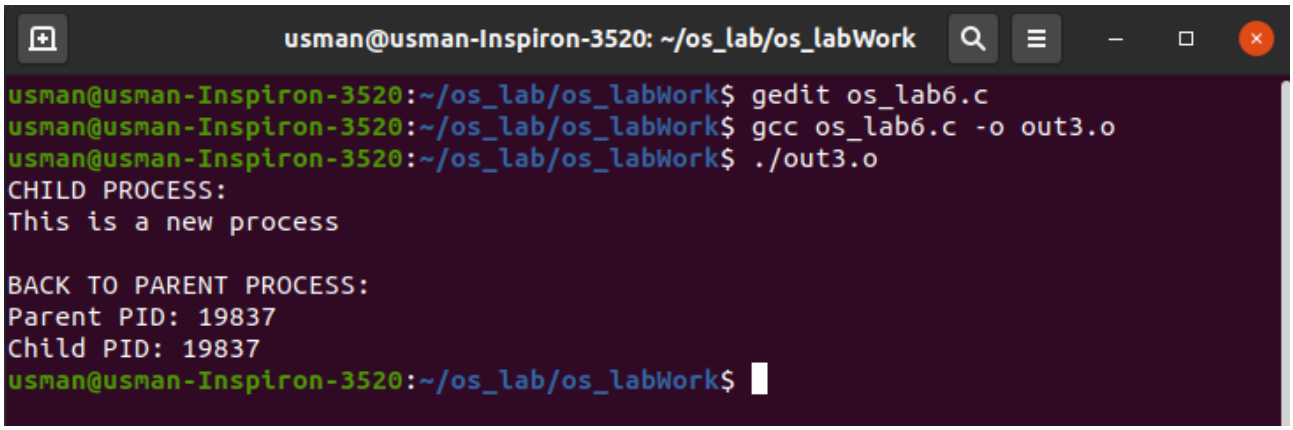
Print.c CODE:




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

int main()
{
        printf("This is a new process \n");
        return 0;
}
```

OUTPUT:


usman@usman-Inspiron-3520: ~/os_lab/os_labWork
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ gedit os_lab6.c
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ gcc os_lab6.c -o out3.o
usman@usman-Inspiron-3520:~/os_lab/os_labWork$ ./out3.o
CHILD PROCESS:
This is a new process
BACK TO PARENT PROCESS:
Parent PID: 19837
Child PID: 19837
usman@usman-Inspiron-3520:~/os_lab/os_labWork$