**PRACTICAL - 1**

**Aim**: - Write a program for creating Child process using fork() system call. Print the process ID of child and parent process. Implement the program in UNIX/Linux**.**

**Theory-**  System call fork() is used to create processes. It takes no arguments and returns a process ID. The purpose of fork() is to create a new process, which becomes the child process of the caller. After a new child process is created, both processes will execute the next instruction following the fork() system call. Therefore, we have to distinguish the parent from the child.

This can be done by testing the returned value of fork():

* If fork() returns a negative value, the creation of a child process was unsuccessful.
* fork() returns a zero then a new child process is created successfully.
* fork() returns a positive value then, the positive value is the process ID of a child’s process to the parent.

**Code:** C Language

#include<stdio.h>

#include<sys/types.h>

#include<unistd.h>

void main(){

int pid=fork();

if (pid==0){

printf("Child PID: %d\n",getpid());

}

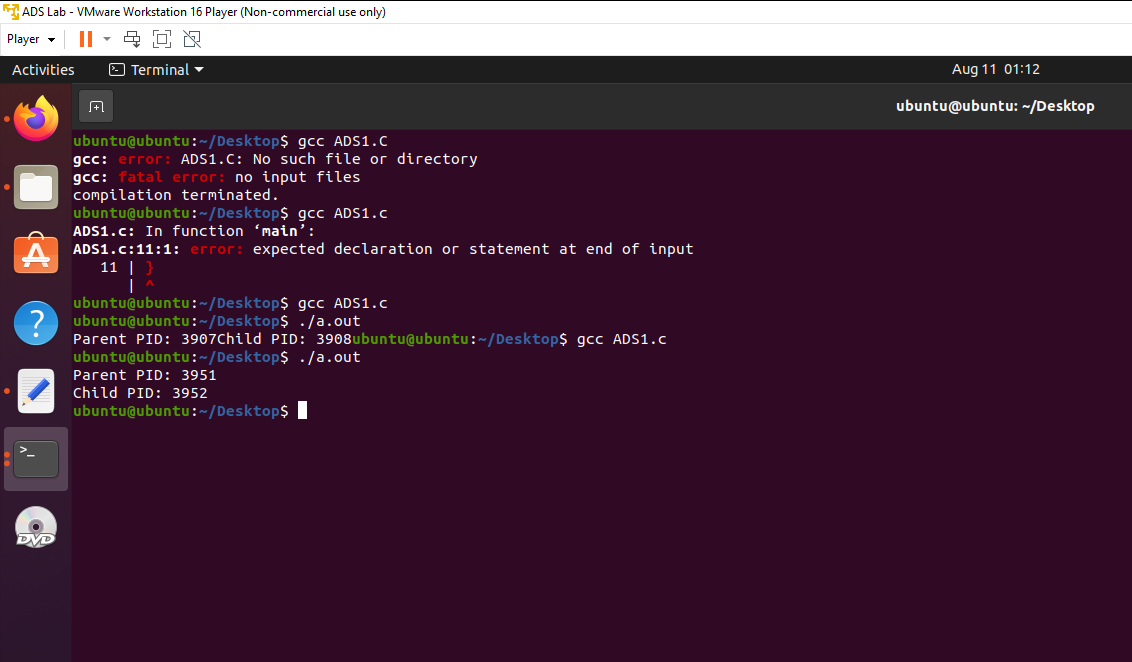
else if(pid>0){

printf("Parent PID: %d\n",getpid());

}

}

**Output:**

****