**OS Lab Exercise – 2**

**Processes and System Calls**

**Exercise 1- Create process and print parent ID and Child ID**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<unistd.h>**

**int main()**

**{**

**int child = fork();**

**if(child>0)**

**{**

**printf("PARENT:\n");**

**printf("PROCESS ID: %d\n",getpid());**

**printf("PROCESS ID OF PARENT: %d\n",getppid());**

**}**

**else**

**{**

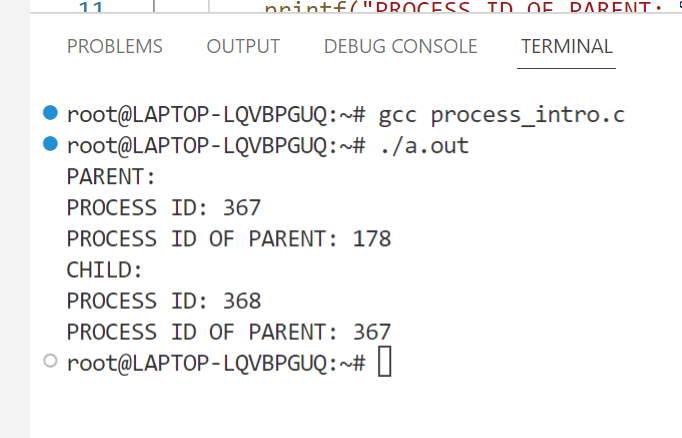
**printf("CHILD:\n");**

**printf("PROCESS ID: %d\n",getpid());**

**printf("PROCESS ID OF PARENT: %d\n",getppid());**

**}**

**}**

****

**Exercise 2- Create a process and compute factorial in child and Fibonacci in parent as executable**

**#include<stdio.h>**

**#include<unistd.h>**

**int fibo(int);**

**long fact(int);**

**int main()**

**{**

**printf("PARENT:\n");**

**printf("PROCESS ID: %d\n",getpid());**

**printf("FIBONACCI NUMBER OF 20: %d\n",fibo(20));**

**int child = fork();**

**if(child==0)**

**{**

**printf("CHILD:\n");**

**printf("PROCESS ID: %d\n",getpid());**

**printf("FACTORIAL OF 10: %ld\n",fact(10));**

**}**

**}**

**int fibo(int n)**

**{**

**if(n<=1)**

**{**

**return n;**

**}**

**else**

**{**

**return (fibo(n-1)+fibo(n-2));**

**}**

**}**

**long fact(int n)**

**{**

**if(n==1 || n==0)**

**{**

**return 1;**

**}**

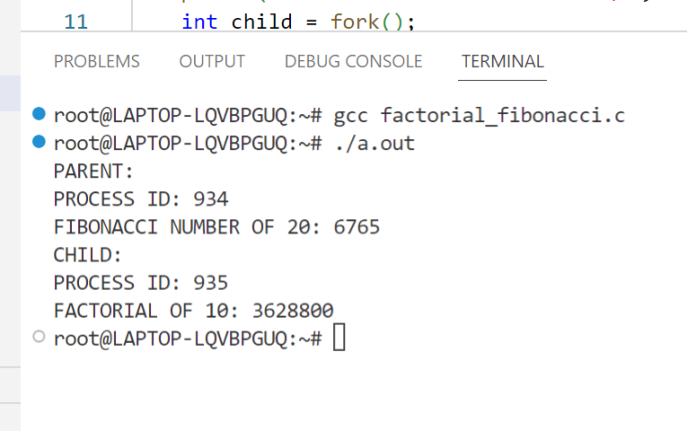
**else**

**{**

**return n\*fact(n-1);**

**}**

**}**

****

**Exercise3: Program to create four processes (1 parent and 3 children) where they terminates in a sequence as Follows:   
(a) Parent process terminates at last   
(b) First child terminates before parent and after second child.  
(c) Second child terminates after last and before first child.  
(d) Third child terminates first.**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <unistd.h>**

**int main()**

**{**

**int pid, pid1, pid2;**

**printf("PARENT CREATED MY PROCESS ID IS %d\n",getpid());**

**pid = fork();**

**if(pid == 0)**

**{**

**printf("CHILD-1 CREATED FROM PARENT %d MY PROCESS ID IS %d\n",getppid(),getpid());**

**sleep(3);**

**printf("CHILD-1 WITH PROCESS ID %d TERMINATED\n",getpid());**

**}**

**else**

**{**

**pid1 = fork();**

**if (pid1 == 0)**

**{**

**printf("CHILD-2 CREATED FROM PARENT %d MY PROCESS ID IS %d\n",getppid(),getpid());**

**sleep(2);**

**printf("CHILD-2 WITH PROCESS ID %d TERMINATED\n",getpid());**

**}**

**else**

**{**

**pid2 = fork();**

**if (pid2 == 0)**

**{**

**printf("CHILD-3 CREATED FROM PARENT %d MY PROCESS ID IS %d\n",getppid(),getpid());**

**printf("CHILD-3 WITH PROCESS ID %d TERMINATED\n",getpid());**

**}**

**else**

**{**

**sleep(4);**

**printf("PARENT WITH PROCESS ID %d TERMINATED\n",getpid());**

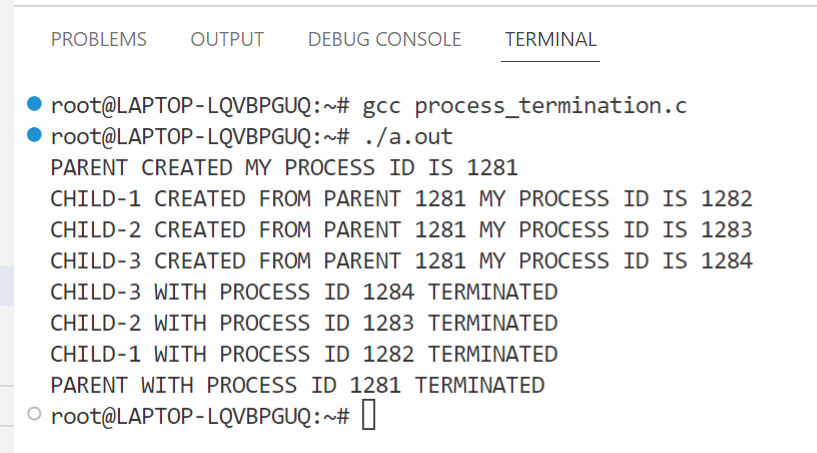
**}**

**}**

**}**

**return 0;**

**}**



**Exercise 4 - Modify above code with necessary system call to ensure the above points so that it works fine.**

**Expected Output:**

**Even numbers are followed by Odd numbers.**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<unistd.h>**

**int main()**

**{**

**int i;**

**if(fork() == 0)**

**{**

**for(i=1;i<10;i++)**

**if(i%2 == 0)**

**printf("even %d\n",i);**

**}**

**else**

**{**

**sleep(2);**

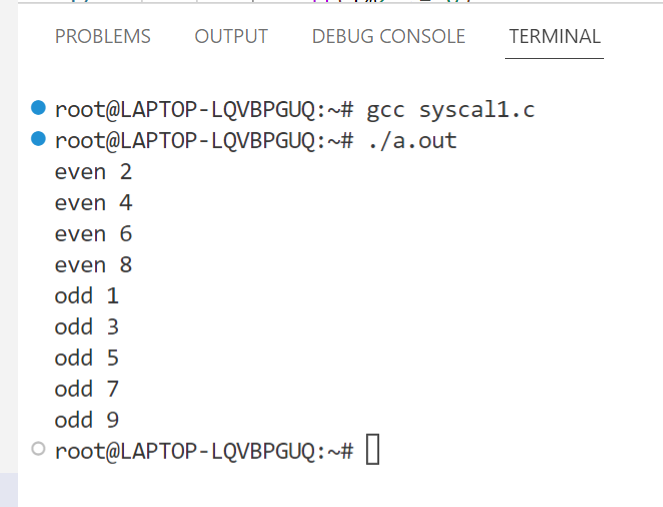
**for(i=1;i<10;i++)**

**if(i%2 != 0)**

**printf("odd %d\n",i);**

**}**

**}**

****

**20BAI1003**

**Raagulbharatwaj K**