Term Work

On

Operating System

(PCS 506)

Submitted to:

Dr. Pardeep Singh
Assistant Professor
Gehu, Dehradun

Submitted by:

Anurag Pandey
University Roll. No.: 2018205
Class Roll No./Section: 11/A

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRTAPHIC ERA HILL UNIVERSITY, DEHRADUN

Graphic Era HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

DEPARTMENT OF CSE
STUDENT LAB REPORT SHEET

Photograph
Passport Size

Name of Student ..... Mob. No .....

Address Permanent .....

Father's Name ..... Occupation ..... Mob. No .....

Mother's Name ..... Occupation ..... Mob. No .....

Section ..... Branch ..... Semester ..... Class Roll No ..... Grade A B C

Local Address ..... Email ..... Marks 5 3 1

S.No.	Practical	D.O.P.	Date of Submission	Grade (Viva)	Grade (Report File)	Total Marks (out of 10)	Student's Signature	Teacher's Signature
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								

PRACTICAL 1

Question: Write a C program to demonstrate the use of fork() system call.

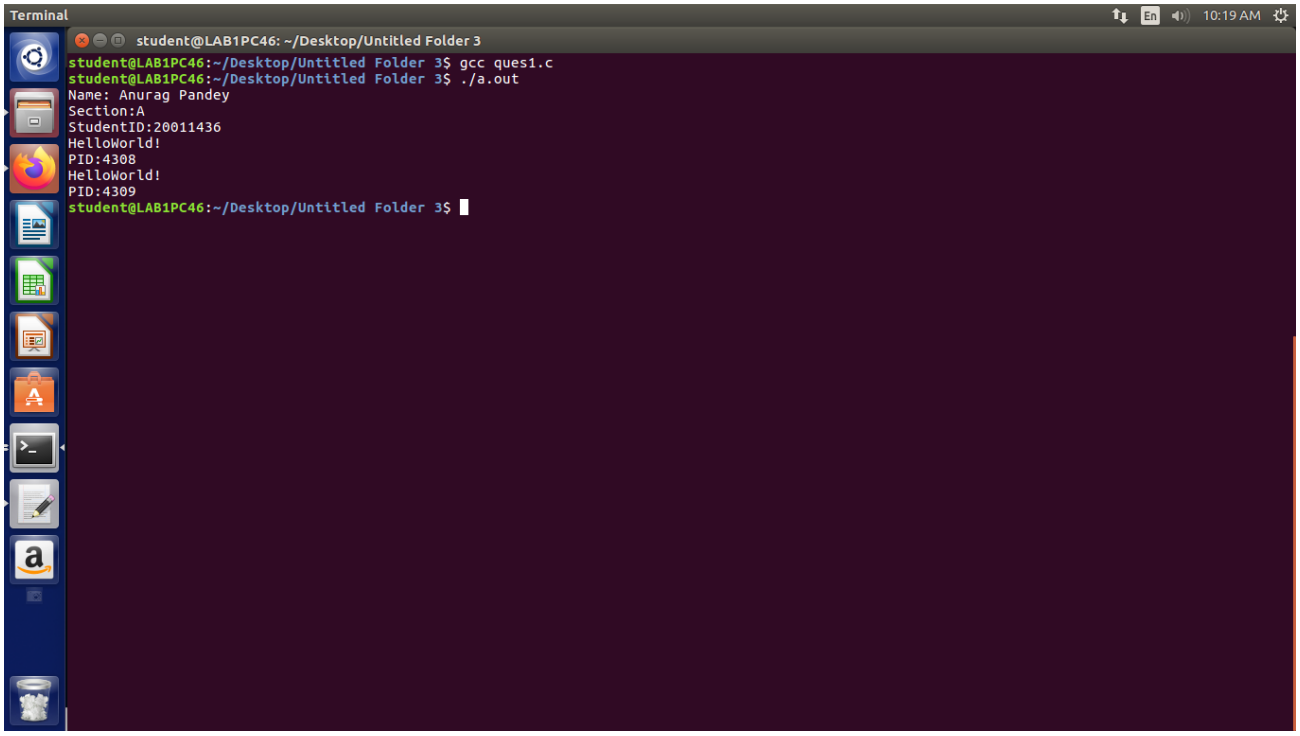
About Fork() function:

Fork system call is used to create new process which is called child process which runs concurrently with the parent process. Parent process is the process which makes the fork() call. Fork() function is defined in header unistd.
Fork() system call is Unix/Linux specific system call.
PID is Process Identification Number on Linux/Unix OS.

Source Code:

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

int main()
{
        printf("Name: Anurag Pandey \nSection: A \nStudent ID: 20011436\n");
        fork();
        printf("Hello World!\n");
        printf("PID: %d\n", getpid());
        return 0;
}
```

Output


Terminal
10:19 AM
student@LAB1PC46: ~/Desktop/Untitled Folder 3
student@LAB1PC46:~/Desktop/Untitled Folder 3$ gcc ques1.c
student@LAB1PC46:~/Desktop/Untitled Folder 3$ ./a.out
Name: Anurag Pandey
Section:A
StudentID:20011436
HelloWorld!
PID:4308
HelloWorld!
PID:4309
student@LAB1PC46:~/Desktop/Untitled Folder 3$

PRACTICAL 2

Question: Write a C program in which parent process computes the sum of even Numbers and child process computes the sum of odd number stored in an array using a fork().
First the child process should print its answer i.e sum of odd number then the parent process should print its answer i.e the sum of even number.

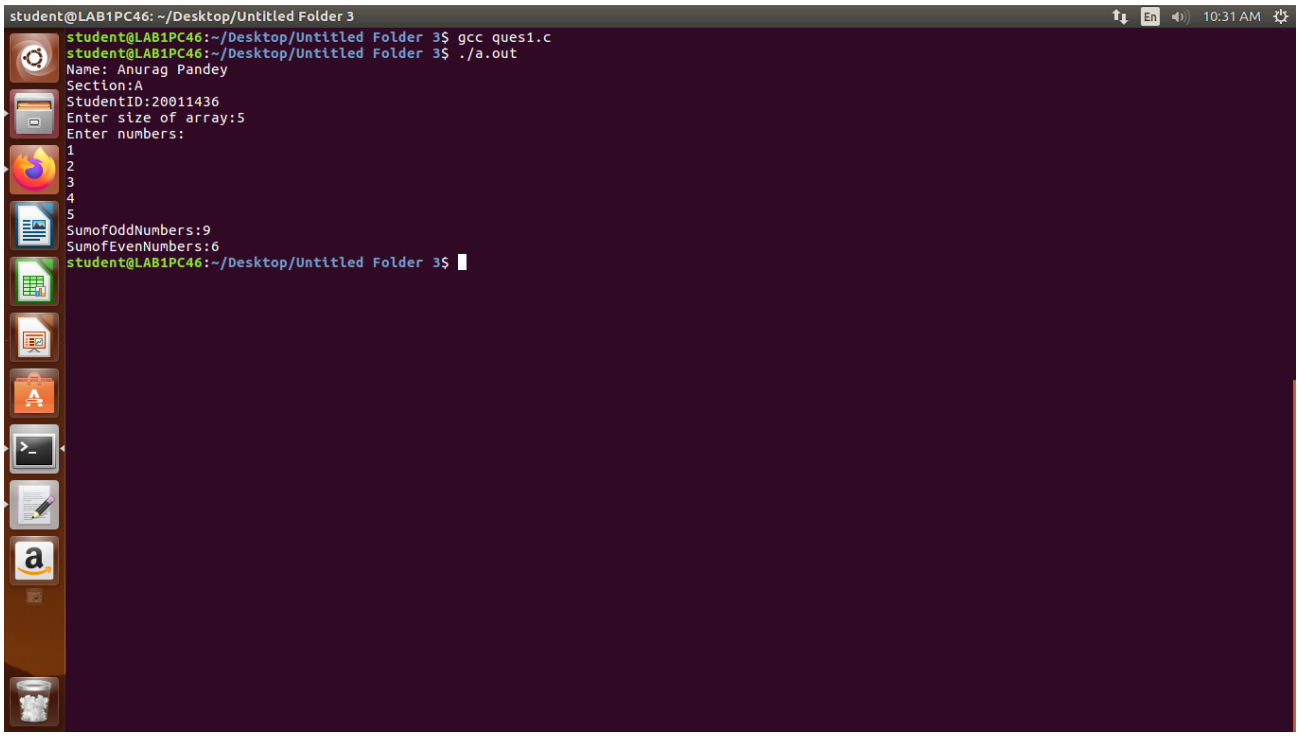
PID: PID is Process Identification Number on Linux/Unix OS.
In child process, it returns 0

Source Code:

```
#include<stdio.h>
#include<unistd.h>
int main()
{
	printf("Name: Anurag Pandey\nSection: A \nStudent ID: 20011436n");
	int even_sum = 0, odd_sum = 0, n;

	printf("Enter size of array: ");
	scanf("%d",&n);
	int arr[n];
	printf("Enter numbers:\n");
	for(int i = 0; i < n; i++)
	{
		scanf("%d",&arr[i]);

	}

	int pid = fork();
	if(pid != 0)
	{
		for(int i = 0; i < n; i++)
		{
			if(arr[i]%2 != 0)
				odd_sum += arr[i];
		}
		printf("Sum of Odd Numbers: %d\n", odd_sum);
	}
	else
	{	for(int i = 0; i < n; i++)
```

```
{
                        if(arr[i]%2 == 0)
                                even_sum += arr[i];
                }
                printf("Sum of Even Numbers: %d\n",even_sum);
        }

        return 0;
}
```

Output:



PRACTICAL 3

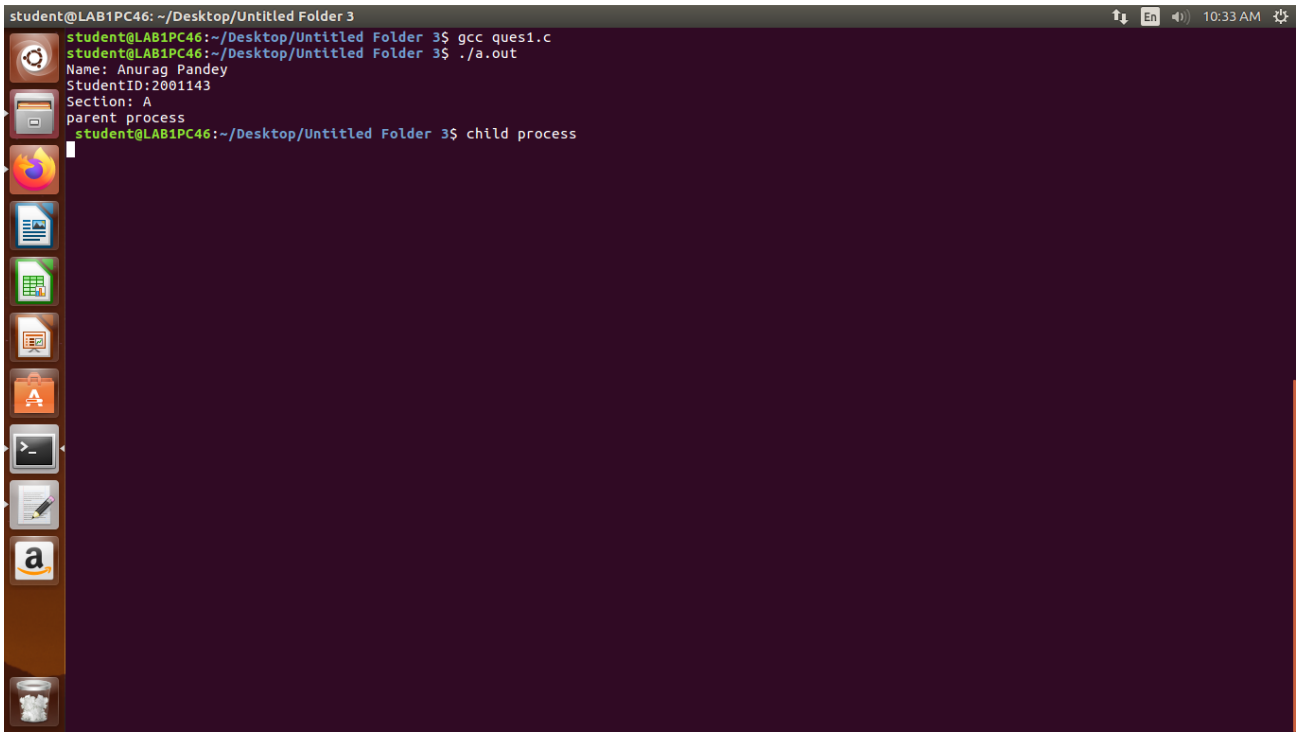
Question: Write a C program to demonstrate Orphan Process using fork function.

Orphan Process: An orphan process is a process whose parent has finished. Suppose P1 and P2 are two process such that P1 is the parent process and P2 is the child process of P1. Now, if P1 finishes before P2 finishes, then P2 becomes an orphan process. The following programs we will see how to create an orphan process**.**

Source Code:

```
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
int main()
{
    int pid = fork();
    if(pid!=0)
   {
            printf("parent process \n ");
            exit(0);
    }
    else if (pid==0){
            sleep(2);
            printf("child process \n");
    }
    return 0;
}
```

Output:


student@LAB1PC46: ~/Desktop/Untitled Folder 3
10:33 AM
student@LAB1PC46:~/Desktop/Untitled Folder 3$ gcc ques1.c
student@LAB1PC46:~/Desktop/Untitled Folder 3$ ./a.out
Name: Anurag Pandey
StudentID:2001143
Section: A
parent process
student@LAB1PC46:~/Desktop/Untitled Folder 3$ child process

PRACTICAL 4

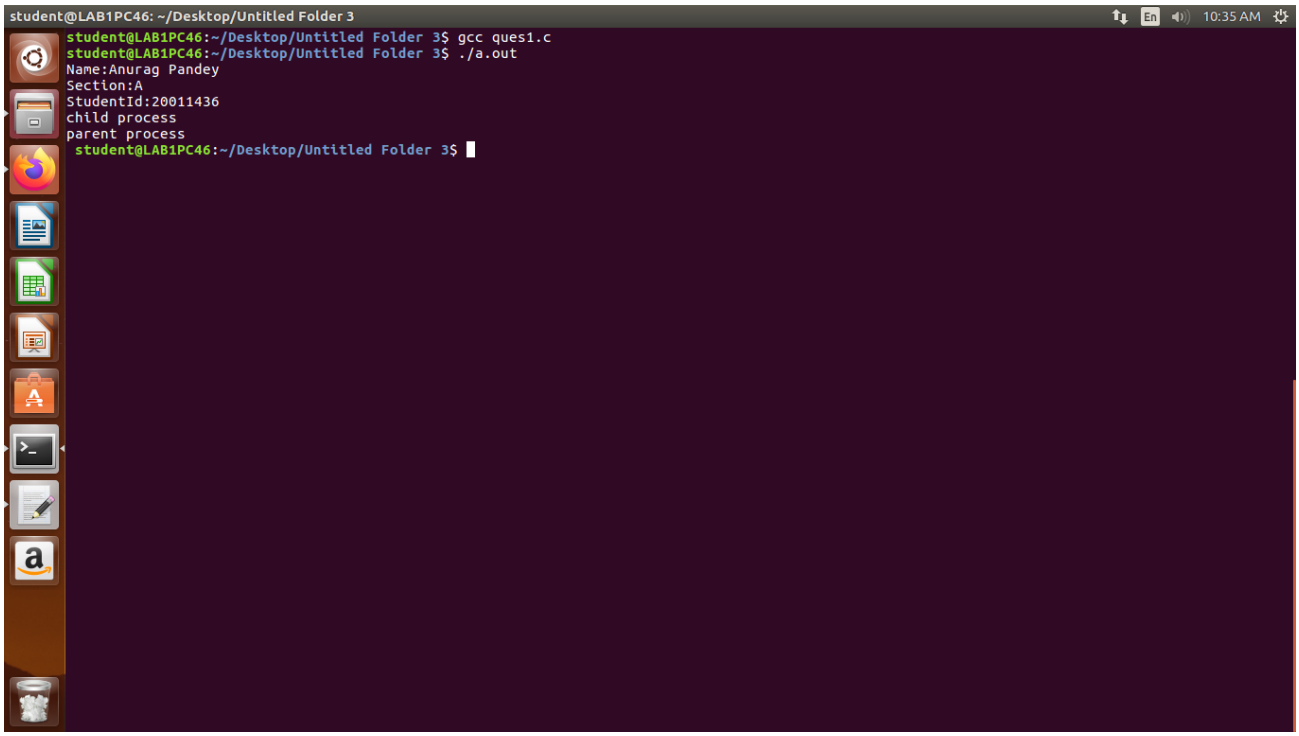
Question: Write a C program to demonstrate Zombie Process using fork function.

Zombie Process: A zombie process is a process in its terminated state. This usually happens in a program that has parent-child functions. After a child function has finished execution, it sends an exit status to its parent function. Until the parent function receives and acknowledges the message, the child function remains in a “zombie” state, meaning it has executed but not exited.

Source Code:

```
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
int main()
{
    int pid = fork();
    if(pid!=0)
    {
            sleep(2);
            printf("parent process \n ");
    }
    else
   {
            printf("child process \n");
            exit(0);
    }
    return 0;
}
```

Output:


student@LAB1PC46: ~/Desktop/Untitled Folder 3
10:35 AM
student@LAB1PC46:~/Desktop/Untitled Folder 3$ gcc ques1.c
student@LAB1PC46:~/Desktop/Untitled Folder 3$ ./a.out
Name:Anurag Pandey
Section:A
StudentId:20011436
child process
parent process
student@LAB1PC46:~/Desktop/Untitled Folder 3$