



**Graphic Era**  
**HILL UNIVERSITY**

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

# **Term Work**

## **On**

# **Operating System**

## **(PCS 506)**

**Submitted to:**

Dr. Pardeep Singh  
Assistant Professor  
Gehu, Dehradun

**Submitted by:**

Malkeet Singh  
University Roll. No.: 2018460  
Class Roll No./Section: 29/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

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

Photograph  
Passport Size

S.N o.	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:**

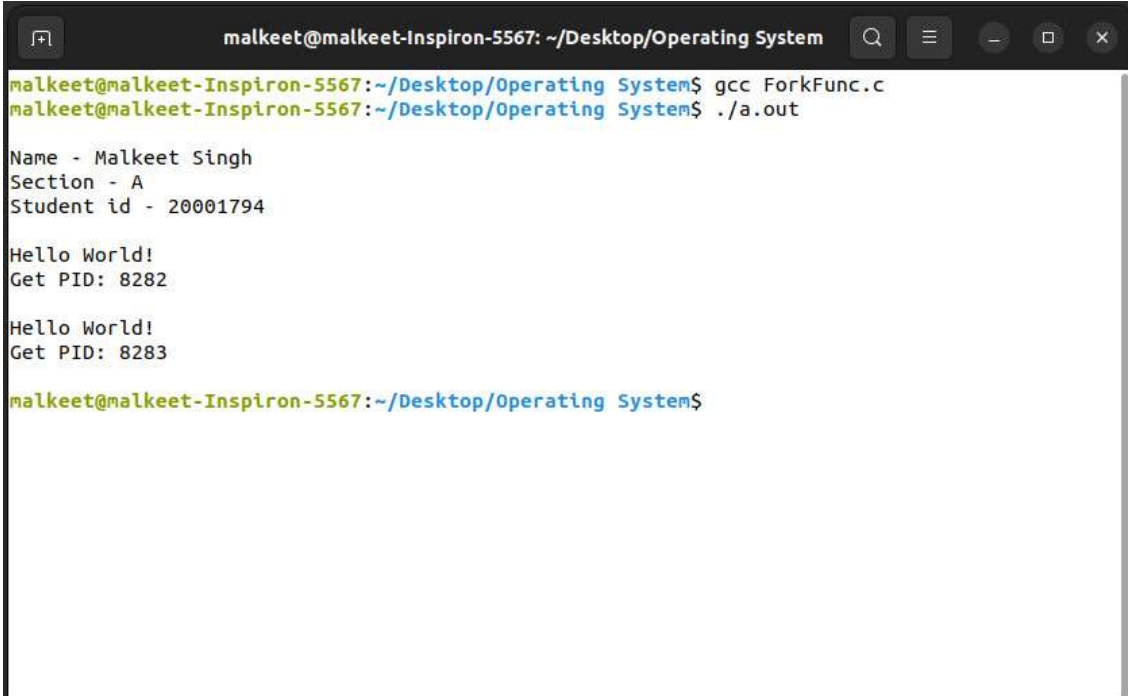
```
#include<stdio.h>
#include<unistd.h>

int main()
{
    printf("\nName - Malkeet Singh \nSection - A \nStudent id - 20001794\n\n");

    fork();
    printf("Hello World!\n");
    printf("Get PID: %d\n\n", getpid());

    return 0;
}
```

## Output:



```
malkeet@malkeet-Inspiron-5567: ~/Desktop/Operating System
malkeet@malkeet-Inspiron-5567:~/Desktop/Operating System$ gcc ForkFunc.c
malkeet@malkeet-Inspiron-5567:~/Desktop/Operating System$ ./a.out

Name - Malkeet Singh
Section - A
Student id - 20001794

Hello World!
Get PID: 8282

Hello World!
Get PID: 8283

malkeet@malkeet-Inspiron-5567:~/Desktop/Operating System$
```

The image shows a terminal window with a dark title bar. The title bar contains the text 'malkeet@malkeet-Inspiron-5567: ~/Desktop/Operating System' and standard window control buttons (search, menu, zoom, and close). The terminal content shows the compilation of 'ForkFunc.c' using 'gcc' and its execution using './a.out'. The program's output consists of two identical blocks of text, each containing personal information and a 'Hello World!' message with a 'Get PID' value. The first block shows a PID of 8282, and the second block shows a PID of 8283, indicating that the program successfully forked twice. The prompt returns to the shell after the execution.

---

## **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 - Malkeet Singh \nSection - A \nStudent ID - 20011794\n\n");
    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:

```
malkeet@malkeet-Inspiron-5567: ~/Desktop/Operating System
malkeet@malkeet-Inspiron-5567:~/Desktop/Operating System$ gcc Even_Odd_Sum.c
malkeet@malkeet-Inspiron-5567:~/Desktop/Operating System$ ./a.out
Name - Malkeet Singh
Section - A
Student ID - 20011794

Enter size of array: 10
Enter numbers:
1 2 3 4 5 6 7 8 9 10
Sum of Even Numbers: 30
Sum of Odd Numbers: 25
malkeet@malkeet-Inspiron-5567:~/Desktop/Operating System$ ./a.out
Name - Malkeet Singh
Section - A
Student ID - 20011794

Enter size of array: 5
Enter numbers:
1 1 1 1 1
Sum of Even Numbers: 0
Sum of Odd Numbers: 5
malkeet@malkeet-Inspiron-5567:~/Desktop/Operating System$
```