Assignment 3

```
Name : Ravish Ranjan
Course : MCA
Semester : 2nd semester
```

```
Write a C/C++ program that creates a file using the open() system call,
then creates two child processes using fork(), where each child process
writes a Fibonacci series along with its process ID and parent process ID
to the same file, and finally observe the effect on the file contents due
to concurrent writes by the child processes.
```

Ans

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/wait.h>
#include <sys/stat.h>
#include <string.h>
using namespace std;

void fiboSer(int n,int fd){
    long long t1 = 0,t2 = 1, temp;
    char buffer[1024] ;
    pid_t pid = getpid(),ppid = getppid();

    int len = snprintf(buffer, sizeof(buffer), "\n[Process pid: %d, ppid:
%d] Starting Fibonacci:\n", (int)pid, (int)ppid);
    write(fd, buffer, len);

    for (int i = 1; i <= n; ++i) {
        len = snprintf(buffer, sizeof(buffer), "pid %d: %lld\n", (int)pid,
(long long)t1);
        if (write(fd, buffer, len) < 0) {
            perror("Write error");
            exit(1);
        }

        temp = t1 + t2;
        t1 = t2;
        t2 = temp;

        usleep(1000);
    }
```

```
}

int main(){
    int fd;

    fd = open("fibo.txt",O_CREAT | O_WRONLY | O_TRUNC,0644);
    if (fd < 0) {
        perror("Error opening file");
        exit(1);
    }

    printf("Parent Process (pid: %d) opened file.\n", getpid());

    pid_t pid1 = fork();

    if (pid1 < 0) {
        perror("Fork 1 failed");
        exit(1);
    }

    if (pid1 == 0) {
        fiboSer(20,fd);
        close(fd);
        exit(0);
    }

    pid_t pid2 = fork();

    if (pid2 < 0) {
        perror("Fork 2 failed");
        exit(1);
    }

    if (pid2 == 0) {
        fiboSer(20,fd);
        close(fd);
        exit(0);
    }

    wait(NULL);
    wait(NULL);

    printf("Both child processes finished.\n");

    close(fd);
    return 0;
}
```

```
[Process pid: 15472, ppid: 15471] Starting Fibonacci:
pid 15472: 0
```

```
[Process pid: 15473, ppid: 15471] Starting Fibonacci:
pid 15473: 0
pid 15472: 1
pid 15473: 1
pid 15472: 1
pid 15473: 1
pid 15472: 2
pid 15473: 2
pid 15472: 3
pid 15473: 3
pid 15472: 5
pid 15473: 5
pid 15472: 8
pid 15473: 8
pid 15472: 13
pid 15473: 13
pid 15472: 21
pid 15473: 21
pid 15472: 34
pid 15473: 34
pid 15472: 55
pid 15473: 55
pid 15472: 89
pid 15473: 89
pid 15472: 144
pid 15473: 144
pid 15472: 233
pid 15473: 233
pid 15472: 377
pid 15473: 377
pid 15472: 610
pid 15473: 610
pid 15472: 987
pid 15473: 987
pid 15472: 1597
pid 15473: 1597
pid 15472: 2584
pid 15473: 2584
pid 15472: 4181
pid 15473: 4181
```