Assignment-4

1901CS75 Siddharth Sanskritayan

Q1)

```
[02/01/22]seed@VM:~/.../OSLab4$ gcc q1.c -o q1
[02/01/22]seed@VM:~/.../OSLab4$ ./q1
Enter the value of n to create n zombie process:
5
Child process executed
Child process executed
Child process executed
Child process executed
Child process executed
Parent process executed
```

```
[02/01/22]seed@VM:~$ ps aux | grep Z+
seed      4796  0.0  0.0      0     0 pts/4    Z+   12:35   0:00 [q1] <defunct>
seed      4797  0.0  0.0      0     0 pts/4    Z+   12:35   0:00 [q1] <defunct>
seed      4798  0.0  0.0      0     0 pts/4    Z+   12:35   0:00 [q1] <defunct>
seed      4799  0.0  0.0      0     0 pts/4    Z+   12:35   0:00 [q1] <defunct>
seed      4800  0.0  0.0      0     0 pts/4    Z+   12:35   0:00 [q1] <defunct>
seed      4828  0.0  0.0   7728  1784 pts/17   S+   12:35   0:00 grep --color=auto Z+
[02/01/22]seed@VM:~$
```

Q2)

```
[02/01/22]seed@VM:~/.../OSLab4$ gcc q2.c -o q2
[02/01/22]seed@VM:~/.../OSLab4$ ./q2
Enter the value of n to create n orphan process:
4
Parent process completed
[02/01/22]seed@VM:~/.../OSLab4$
```

After some time(when sleep over):

```
Enter the value of n to create n orphan process:
4
Parent process completed
[02/01/22]seed@VM:~/.../OSLab4$ Child process executed
Child process executed
Child process executed
Child process executed
```

Q3)

```
[02/01/22]seed@VM:~/.../OSLab4$ gcc q3.c -o q3
[02/01/22]seed@VM:~/.../OSLab4$ ./q3
Enter the value of n to generate first n lucas sequence:
10

Child process 1 executed (lucas sequence generated)

In child process 2 and printing the lucas sequence:
2 1 3 4 7 11 18 29 47 76
Child Process 2 executed

Parent process terminated
[02/01/22]seed@VM:~/.../OSLab4$
```

Q4)

```
[02/01/22]seed@VM:~/.../OSLab4$ gcc q4.c -o q4
[02/01/22]seed@VM:~/.../OSLab4$ ./q4



 The PID of first child is: 5243
Now the source program is copied to f2.c



 THe PID of second child is: 5244
Now printing the contents of f2.c
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
```

```
int main(int argc, char* argv[]){
    fflush(stdout);
    pid_t PID1;
    PID1=fork();
    if(PID1==-1)
    {
        printf("Fork error\n");
    }
    else if(PID1 > 0)                                          //parent process
    {
        while(wait(NULL) > 0);
        fflush(stdout);
        if(fork() > 0)
        {
            while (wait(NULL) > 0);
            fflush(stdout);
            pid_t PID2;
            PID2=fork();
            if (PID2 == 0)                                     // Third child
            {
                printf("\n\n\n");
                printf("\nThe PID of third child is: %d \n", getpid());
                if( access( "f2.c", F_OK ) == 0 )              //check for existence of file
                {
                    remove("f2.c");
                    printf("f2.c is deleted now\n");
                }
                else
```

```
                {
                    printf("file does not exist\n");
                }
            }
            else                                                //parent process
            {
                while (wait(NULL) > 0);
                printf("\n\n\n");
                printf("\nThe parent PID is: %d\n\n", getpid());
                printf("Parent process now executed");
            }
        }
        else                                                    //second child
        {
            printf("\n\n\n");
            printf("\n THe PID of second child is: %d\n", getpid());
            printf("Now printing the contents of f2.c\n");
            FILE *file;
            file = fopen("f2.c", "r");
            char ch = fgetc(file);
            while(ch != EOF)
            {
                fputc(ch, stdout);                          // Print the contents
                ch = fgetc(file);
            }
            fclose(file);
        }
    }
    else                                                    //First child
    {
```

```
    else                                                    //First child
    {
        printf("\n\n\n");
        printf("\n The PID of first child is: %d\n", getpid());
        FILE *file1;
        FILE *file2;
        file1 = fopen("q4.c", "r");
        file2 = fopen("f2.c", "w");
        char c = fgetc(file1);
        while (c != EOF)
        {
            fputc(c, file2);
            c = fgetc(file1);
        }
        printf("Now the source program is copied to f2.c\n");
        fclose(file1);
        fclose(file2);
    }
}
```

```
The PID of third child is: 5245
f2.c is deleted now
```

```
The parent PID is: 5242

Parent process now executed[02/01/22]seed@VM:~/.../OSLab4$
```