Task 9 IPC Using Signals

Name: Muhammad Sherjeel Akhtar

Roll No: 20p-0101

Subject: Operating Systems Lab

Submitted To Respected Sir: Muhammad Ahsan

Section: BCS-4A

——————————————————————————786

Code:

```
#include <stdio.h>
#include <unistd.h>
#include <signal.h>
#include <sys/wait.h>

void sigHandle(int sig)
{
    printf("Child process %d received signal %d\n", getpid(), sig);
}
int main()
{
    pid_t pid1, pid2;
    int status1, status2;
    pid1 = fork();
    if (pid1 == 0)
    {
        pid2 = getppid();
        signal(SIGUSR1, sigHandle);
        printf("First child process %d waiting for signal\n", getpid());
```

```
        while (1) {}
        return 0;
    }

    pid2 = fork();
    if (pid2 == 0)
    {
        sleep(1);
        printf("Sending signal from second child process %d to first child process %d\n", getpid(), pid1);
        kill(pid1, SIGUSR1);
        return 0;
    }
    waitpid(pid1, &status1, 0);
    waitpid(pid2, &status2, 0);
    return 0;
}
```

First Child Process:

```
if (pid1 == 0)
    {
        pid2 = getppid();
        signal(SIGUSR1, sigHandle);
        printf("First child process %d waiting for signal\n", getpid());
        while (1) {}
        return 0;
    }
```

Second Child Process:

```
pid2 = fork();
    if (pid2 == 0)
    {
        sleep(1);
        printf("Sending signal from second child process %d to first child process %d\n", getpid(), pid1);
        kill(pid1, SIGUSR1);
        return 0;
    }
```

Visual Demonstration:

```
spoofy@spoofy-Precision-M4600:~/Downloads/FInal 786 OS$ gcc prac.c -o out
spoofy@spoofy-Precision-M4600:~/Downloads/FInal 786 OS$ ./out
First child process 35153 waiting for signal
Sending signal from second child process 35154 to first child process 35153
Child process 35153 received signal 10
```