CS 333 - Lab 7 - Report

By - 200050133

Part - 0

Following are the outputs obtained after running both the codes as given to us.

- processes.c

```
shubak@MSI:/mnt/c/Users/shuba/Desktop/Labs/CS333/lab7/auxiliary_files$ gcc processes.c -o proc
shubak@MSI:/mnt/c/Users/shuba/Desktop/Labs/CS333/lab7/auxiliary_files$ ./proc
PID of Process: 34
Value of x: 2
PID of Process: 35
Value of x: 7
```

In this process, we can see that two different processes have been created, one being the parent and the other being the child, and both of them can be seen to print 2 different values of x, this is because both processes have 2 separate copies of the variable which they modify separately, hence only the child process increments x while parent doesn't

- threads.c

```
shubak@MSI:/mnt/c/Users/shuba/Desktop/Labs/CS333/lab7/auxiliary_files$ gcc threads.c -o thread -lpthread
shubak@MSI:/mnt/c/Users/shuba/Desktop/Labs/CS333/lab7/auxiliary_files$ ./thread
PID of Foo: 46
PthreadID : 140060475655936
Value of x: 7
PID of Bar: 46
PthreadID : 140060467201792
Value of x: 7
```

In this program, we can see that no new processes have been created, but threads have been spawned in the same process. We can see that both the threads (having different threadIDs) have the same PIDs, and both of them have printed the same value of x, indicating that there is no copy of the variable and both threads modify the same variable.