Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 03

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance:

Date of Submission:

Submitted by

Name: Ashikur Rahman Miran

ID:IT-18014

3rd year 1st semester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment No: 03

Experiment Name: Threads on Operating System.

Objectives:

i) What is thread?
ii) Types of threads.
iii) Implementation of thread.

Theory:

A thread is an execution unit that has its own program counter, a stack and a set of registers that reside in a process. Threads can't exist outside any process. Also, each thread belongs to exactly one process. The information like code segment, files, and data segment can be shared by the different threads.

Threads are popularly used to improve the application through parallelism. Actually only one thread is executed at a time by the CPU, but the CPU switches rapidly between the threads to give an illusion that the threads are running parallelly.

Types of thread:

There are two types of thread. They are

- User level threads.
- Kernel level threads.

User level Threads:

- The User-level threads are managed by users.
- These threads are faster to create and manage.
- It is implemented using user-level libraries and not by the system calls. So, no call to the operating system is made when a thread switches the context.
- Each process has its own private thread table to keep the track of the threads.

Kernel level Thread:

- The kernel knows about the thread and is supported by the OS.
- The threads are created and implemented using system calls.
- Kernel-level threads are slower to create and manage as compared to user-level threads.

Multithreading models:

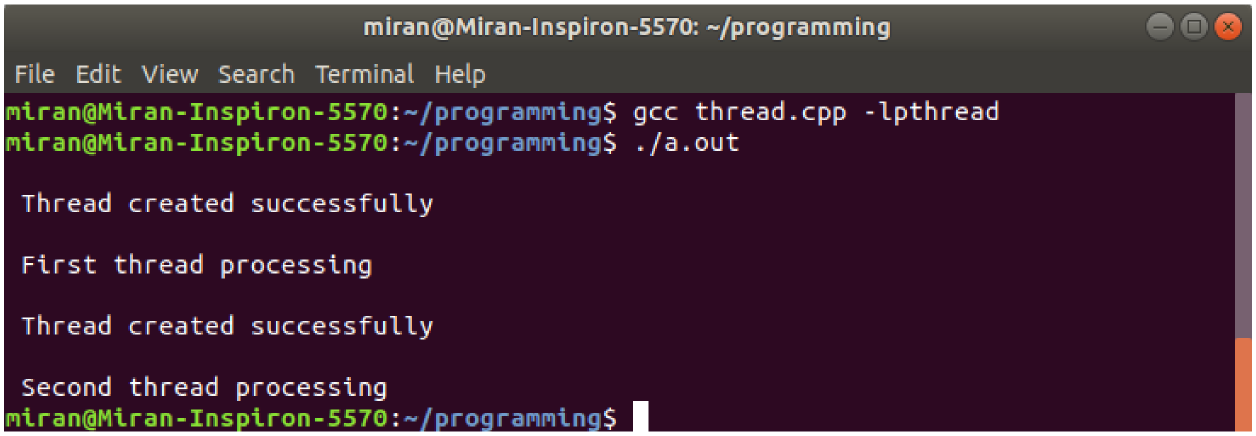
- Many to one models.
- One to one models.
- Many to many models.

Source code:

```
#include<stdio.h>
#include<string.h>
#include<pthread.h>
#include<stdlib.h>
#include<unistd.h>
pthread_t tid[2];
void* doSomeThing(void *arg)
{
   unsigned long i = 0;
   pthread_t id = pthread_self();
   if(pthread_equal(id,tid[0]))
   {
         printf("\n First thread processing\n");
   }
   else
   {
         printf("\n Second thread processing\n");
   }
   for(i=0; i<(0xFFFFFFFF);i++);
         return NULL;
}
int main(void)
{
   int i = 0;
```

```
    int err;
    while(i < 2)
    {
        err = pthread_create(&(tid[i]), NULL, &doSomeThing, NULL);
        if (err != 0)
        printf("\ncan't create thread :[%s]", strerror(err));
        else
        printf("\n Thread created successfully\n");
        i++;
    }
    sleep(5);
    return 0;
}
```

Output:



miran@Miran-Inspiron-5570: ~/programming
File Edit View Search Terminal Help

```
miran@Miran-Inspiron-5570:~/programming$ gcc thread.cpp -lpthread
miran@Miran-Inspiron-5570:~/programming$ ./a.out

 Thread created successfully

 First thread processing

 Thread created successfully

 Second thread processing
miran@Miran-Inspiron-5570:~/programming$
```

Conclusion:

In this lab I learn about thread and different types of thread. I also implement the code and print the output. The output is expected.