



**National University of Sciences and Technology (NUST)**  
**School of Electrical Engineering and Computer Science**

**Department of Computing**

**CS 330: Operating Systems**

**Lab 2: Locks and Pthreads**

**CLO4(Develop programs to interact with OS components through its API)**

**Time: 9.00 AM – 12:00 PM and 02:00 PM – 05:00 PM**



## **Lab 2: Locks and Pthreads**

### **Introduction**

The purpose of this lab is to introduce the concept of Locks and threads and give you an insight about Posix Thread Library.

### **Objectives**

Objectives and expected learning outcomes of this lab are understanding of threads theoretically, thread scheduling and thread pitfalls.

### **Tools/Software Requirement**

Linux OS installed on laptops or systems.

### **Description**

#### **POSIX thread (pthread) libraries**

The POSIX thread libraries are standards based threads API for C/C++. It allows one to spawn a new concurrent process flow. It is most effective on multi-processor or multi-core systems where the process flow can be scheduled to run on another processor thus gaining speed through parallel or distributed processing. Threads require less overhead than "forking" or spawning a new process because the system does not initialize a new system virtual memory space and environment for the process. While most effective on a multiprocessor system, gains are also found on uniprocessor systems which exploit latency in I/O and other system functions which may halt process execution. (One thread may execute while another is waiting for I/O or some other system latency.) Parallel programming technologies such as MPI and PVM are used in a distributed computing environment while threads are limited to a single computer system. All threads within a process share the same address space. A thread is spawned by defining a function and its arguments which will be processed in the thread. The purpose of using the POSIX thread library in your software is to execute software faster.

For further details visit the following link

<http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html#SYNCHRONIZATION>

### **Tasks**



## National University of Sciences and Technology (NUST) School of Electrical Engineering and Computer Science

### **Task 1:**

Study the `lock_init`, `lock_acquire` and `lock_release()` functions in Pintos' `synch.c` file. Clearly describe how each of these functions work. Note that this includes explanations of `sema_up()` and `sema_down()`, functions that are called within `lock_acquire()` and `lock_release()`. Also, identify places in the Pintos codebase where these functions are used. Finally, mention what shared variables these locks are protecting.

### **Task 2:**

Thread programming in Linux is often done using the POSIX Threads (pthreads) library. Introduce yourself to the library by going through the following tutorial:

<http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html#SYNCHRONIZATION>

Compile and run the sample code on the tutorial related to thread creation, thread joining and mutually exclusive locks (mutexes). In your writeup, explain what the sample code does and how you compiled and ran it.

### **Deliverables:**

Text Document containing necessary details.