**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Course Code: CSE 321** | **Credits: 1.5** |
| **Course Name: Operating Systems** | **Semester: Fall 18** |

**Lab 07  
Introduction to Thread**

1. **Overview:**

A thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler. In this lab students will learn the concept of thread, how it works, it’s importance as well as how to create thread.

1. **Lesson Fit:**

Programming knowledge is required for this lab.

1. **Learning Outcome:**

After this lab, students will know the difference between thread and process, be able to create threads in different ways and run threads concurrently within a process.

1. **Anticipated Challenges and Possible Solutions**
   1. When interaction occurs among thread, data race and cached variable problems arise.

**Solutions:** Students should have the knowledge about race conditions, mutex etc.

1. **Acceptance and Evaluation**

Students will show their progress as they complete each task. They will be marked according to their lab performance.

**Activity Detail**

* 1. **Hour: 1  
     Discussion:**

1. Discussion on thread and how it works
2. Why thread is important and its benefits
3. Where we can use threads.
   1. **Hour: 2**
4. Creation of thread in java programming language in different ways.
5. Thread calling and run method implementation.
   1. **Hour: 3**
6. Implementation of multi-threaded program.
7. Thread-sleep, thread priority implementation.
8. **Home tasks**
   1. Create two threads and divide a task between these thread equally. The thread should work concurrently and complete the assigned task.

For example, the task is to calculate the sum of numbers from 1-100. Thread one will calculate sum from 1-50 and thread two will calculate sum from 51-100. And finally the total sum from 1-100 will be printed, where the two threads worked equally.

**Lab Activity List**

**Task 1**

Create a thread by extending thread class and give a name to the thread.

**Task 2**

Create a thread by implementing runnable interface.

**Task 3**

Create two threads and run them. Print some values using loops executing thread and determine what happens.

**Task 4**

Send a thread to sleep for some seconds and let other threads to work in the same method.