**Sec D OS Spring 2018**

**Threads**

pthread\_create(pthread\_t\* , NULL, void\*, void\*)

* First Parameter is pointer of thread ID it should be different for all threads.
* Second Parameter is used to change stack size of thread. Null means use default size.
* Third perimeter is address of function which we are going to use as thread.
* Forth parameter is argument to function.

pthread\_join(i pthread\_t , void\*\*)

Pthread join is used in main program to wait for the end of a particular thread.

* First parameter is Thread ID of particular thread
* Second Parameter is used to catch return value from thread.

**Question 1.**

Write a program which takes some positive integers (let’s say **N** number ofpositive integers) as

command line parameters, creates **N** synchronous threads, and send s the corresponding integer as parameter to the thread function fibonacciGenerator. The function returns the generated series to the main thread. The main thread will then print the thread number and the series generated by that thread. The output will be like:

Thread 1: 0 1 1 2 3 5 8 13

**Example:**

If you pass as command line argument the following numbers: 3 13 34 89

Then the program will create 4 threads. The first thread will find Fibonacci terms until 3 is generated, the second Fibonacci term will find Fibonacci terms until the term generated is 13 , so on and so forth. All generated terms will be output on the screen by the main thread as follows:

Thread 0: 0, 1, 1, 2, 3

Thread 1: 0, 1, 1, 2, 3, 5, 8, 13

Thread 2: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Thread 3: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89

It is possible that the number passed to the thread is not a Fibonacci number. In this case the thread will generate numbers until the term generated is greater than the passed number. For example if 7 is passed as parameter to a thread, then the thread will return the following series:

0, 1, 1, 2, 3, 5, 8

**Question 2.**

Write a multithreaded program that calculates various statistical values for a list of numbers. This program will be passed a series of numbers on the command line and will then create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the minimum value. For example, suppose your program is passed the integers. (The array of numbers must be passed as parameter to threads, and the thread must return the calculated value to main thread).

90 81 78 95 79 72 85

The main thread will print:

The average value is 82

The minimum value is 72

The maximum value is 95