**Project 2 – Programming with pThread**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Language Restrictions: C/C++ only

# Purpose

The purpose of this assignment is to become familiar with the pThreads API and write simple programs using pThreads. The first program is already done for you. You will have to modify it to obtain timing results and explore the use of data parallelism techniques for computing PI. The second part of this programming assignment uses threads in a more functional parallelism paradigm for generating and displaying the Fibonacci sequence.

# Part A: Computing PI With Threads

# The first part of this assignment is to explore the use of threads in an application that computes PI. Download the *compute\_pi\_montecarlo.c (compile: gcc compute\_pi\_montecarlo.c -o pi -lpthread) (execute: ./pi 80000 8)* program to your Project3 subdirectory. Compile and run this program a few times to understand the basic operation and to verify that it does approximate PI. Next, time your program and/or add appropriate timing system calls to your code. Run the application varying the number of threads and number of sample points. You may get a core dump when you exceed the number of threads permitted (note that value). Record the time(s) it takes your program to execute when changing the number of threads and number of sample points.

# Create a plot of time vs. number of threads when the number of sample points is fixed.

# Create a plot of time vs. number of sample points when the number of threads is fixed.

# Use an Excel like application to create your plot.

# The name of this program must be *osproj3a.c* or *osproj3a.cpp*.

# Part B: Computing Fibonacci Numbers with Threads

The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5, 8, .... Formally, it can be expressed as:

fib0 = 0

fib1 = 1

fibn = fibn-1 + fibn-2

Write a multithreaded program that generates the Fibonacci series using the pThreads library. This program should work as follows: The user will enter on the command line the number of Fibonacci numbers that the program will generate. The program will then create a separate thread that will generate the Fibonacci numbers placing the sequence in data that is shared by the threads (an array or vector is probably the most convenient data structure). When the thread finishes execution, the parent thread will output the sequence generated by the child thread. Because the parent thread cannot begin outputting the Fibonacci sequence until the child thread finishes, this will require having the parent thread wait for the child thread to finish.

The name of this program must be ***osproj3b.c*** or ***osproj3b.cpp***

# Other useful information

Please use the provided ***timer-example.c*** as an example for recording times.