

# CS 292 Introduction to Parallel Computing

## Spring 2008

### Programming Assignment #1

Due: Midnight, February 29, 2008

## Objectives

- Practice Pthread programming

## Collaboration

- You may discuss all aspects of the assignment with your classmates. However, you should never show any of your code to another student.

## Description

- **Matrix multiplication with Pthread**

Write a threaded matrix multiplication program. First, you need to write a program (`makemat.c`) that can generate matrix of any dimension and output it into a file in **binary** format. The program takes 3 input parameter from the command line,

```
makemat output_file width height
```

The first two numbers written to the output file should be the width and height of the matrix, followed by the  $n \times m$  random `double` numbers (between 0 and 1). ( $n$  is the width and  $m$  is the height you specified in the command line)

Your matrix multiplication program (`matmul_pthread.c`) should take 4 command line arguments,

```
matmul_pthread a_file b_file c_file number_of_threads
```

where *a\_file* and *b\_file* are the input files for matrix  $A$  and  $B$  generated with the `makemat` program you just wrote. *c\_file* is the output file for matrix  $C (= AB)$ . `number_of_threads` is the number of the threads you want to use. In your program, `number_of_threads` threads are created to do the multiplication and the main thread waits for all of these threads to complete and then write  $C$  to the output file *c\_file*.

**Each thread gets the first uncalculated element of  $C$  and then calculate that element. Then it gets the next uncalculated element of  $C$ , and so on, until  $C$  is finished. When there are no uncalculated elements left, the thread should exit.**

There is a working version in `/home/shiz/mmult/matmul_pthread`. You can play with it. Try different sizes and different number of threads.

You will be using mutex to protect shared variables. However, you will not need condition variables in this part of the assignment.

Test your code extensively. Run it with different number of threads on the gateway and make some comments about the performance improvement you see.

- **Producer/Consumer using Pthread**

Write a C program using Pthread to simulate Producer/Consumer model. In the program, you should first declare an array of integers of size 5. Two threads need to be created. The first thread (the producer) puts numbers in the array, one after the other. The second thread (the consumer) removes numbers from the array. This is done in a **FILO** fashion (First In Last Out). You just need to maintain a pointer/index to the last number of the array. The algorithm of the producer is as follows:

- If the array is not full then put a new number in it (the numbers start at 0 and are incremented by one each time, so the producer produces 0, 1, 2, etc.) until 20 has been added to the array.
- Sleep for 1 to 5 seconds, randomly
- Repeat

The algorithm of the consumer is as follows:

- If the array is not empty then remove its last number
- Sleep for 1 to 5 seconds, randomly
- Repeat until the array becomes empty.
- If the producer has finished adding all the numbers and there are still numbers left in the array, the consumer should remove them one by one (sleep for random seconds between each step).

You should use mutex locks to protect data that must be protected and conditional variables to provide notification mechanism between threads. Each thread should print on the screen what it is doing (e.g, “Thread 0 is putting 8 in the array”, “Thead 1 is removing 12 from the array”, etc.).

Name the source code `pc_pthread.c`.

## What to submit

Submit all the source code, compiled executable etc. as a `tar` file. Along with the tar file, you should also submit a `README` file (preferably in PDF format) describing how to compile and run your program, the known bugs (if any) in your program and any comments.