CSC718 Homework 1

The homework assignment will be graded based on the following criteria:

- Accuracy: 1) the solution meets specific requirements in the problem description; 2) the solution produces correct results; 2) the procedures adopted in the solution are technically sound.
- Efficiency: efficiency will be one of the criteria when grading programing assignment. The solution should produce the desired results efficiently.
- Effort/neatness: the solution includes excellent effort, and all relate work is shown neatly and organized well.

Homework assignment feedback will be available through the DropBox folder on D2L.

For all the programming assignments, you can choose any operating systems you want. I will usually provide C/C++ samples for the programming assignments. If you prefer to use other languages, e.g., Java, they are accepted too. A README.txt is required to submit any programming assignments. In the README.txt, you need to provide the following information:

1) How to compile your program?
2) How to run your program?
3) What is the average running time of your program?
4) What are the expected output results when I run your program?
5) Any descriptions which may help me to compile, run, and verify your answers. (FYI: I check every programming assignment turned in!)

Zip all you source code, project files, supporting files, and README.txt and submit the all-in-one zip file together to the D2L Dropbox. If you have any questions about the homework, please let me know.

1. (10 points) The central processing unit (CPU), also called a processor, receives a program's instructions; decodes those instructions, breaking them into individual parts; executes those instructions; and reports the results, writing them back into memory. The format for that processor comes in one of two primary types: vector and scalar.
    a. What is a vector processor?
    b. What is a scalar processor? How about superscalar processor?
    c. What are the differences between vector and scalar processor?
2. (10 points) Instruction-Level Parallelism (ILP) can be implemented in both hardware and software level to increase the number of instructions executed in parallel. Briefly describe the following ILP techniques:
    a. Instruction pipelining
    b. Speculative execution
    c. Branch prediction

3. (10 pints) Why parallel computing is critical to improve computer performance? In another words, what are the limitations by increasing transistor density in a single chip to improve its performance?
4. (10 points) What are the tradeoffs between preemptive scheduling and non-preemptive scheduling?
5. (10 points) Two sample solutions for the dinning philosopher have been posted in the class web site: one solution is based on mutex and condition variable, the other solution is based on semaphore. Go through the two solutions and answer the following two questions:
    a. (5 points) Describe the difference of these two solutions.
    b. (5 points) which solution is better? Why?
6. (Programming Assignment, 50 points) myhttpd1.cpp is a simple webserver written in C. The sample code has been posted in the class website. You can compile and run the program using the following commands in a Linux environment

```
>gcc myhttpd1 –o myhttpd1

>./myhttpd1 –p 8080
```

After you run the program, open a browser and type http://localhost:8080 and you will see “Welcome to my first page!” in the browser.

a. (10 points) The posted program has a bug. A function call has been skipped on purpose in the program. Review the myhttpd1.cpp code, find the problem, and fix the code.
b. (15 points) myhttpd1.cpp is a single thread program. Create a new program, myhttpd2.cpp, based on myhttpd1.cpp and use multiple threads to process http request.
c. (15 points) Create a new program, myhttpd3.cpp, based on your modification in b). Change the program to a daemon (running in background).
d. (10 points) WeChat is a popular social networking application in China. Red envelope is an application in WeChat which can be used to send “lucky money” to friends. As estimated in 2017, a total of 14.2 billion red envelopes were exchanged via WeChat on New Year’s Eve alone, peaking at midnight with 760,000 transactions per second. It usually takes seconds for WeChat to process each transaction. For such high-volume concurrent transaction requests, what remediations could be used on the server side to ensure each request is served promptly?