COSC 350 System Software
Lab #8

How to submit

- For each program, you need write detailed comments for each statement.
- You need demonstrate each task in front of instructor during the next lab hours.

Task #1: <u>Concurrent Calculation</u>

Write a C program which accepts an integer argument and calculate sum up and factorial (1 + 2 + 3 +… + n, and 1 × 2 × 3 × … × n). Instead of calling sequence of two functions, two threads will be created and each thread works for different task concurrently.

Task #2: <u>Wait for a Thread for Concurrent Calculations</u>

Three treads are sharing an integer array (you may define as global) and each thread do different jobs.

- Thread #1: get test scores from keyboard up to 20 and save into the array.
- Thread #2: Calculate an average score and Medium value and display.
- Thread #3: get the minimum and the maximum score and display.
- Thread #4: clear buffer and set to 0 and display after Thread #2, Thread #3 finishes its job.

Thread #2 and #3 must wait for thread #1 finish its job. Once thread #1 finish its job, thread #2 and thread #3 works concurrently. Thread#4 must wait for thread #2 and Thread #3 finish its job.

Task #3: <u>Interprocess communication using pipe</u>

A parent process asks two integers from command line and send to child by using pipe. The child process make sure two inputs are integers. The child process calculate sum of two integer and output on standard output. The child process continue until input from the parent are EOF.

Task #4: <u>Interprocess communication using popen() and pclose()</u>

popen() create a child and pipe. A parent process asks two arguments from command line: a shell command and a file name. And then send to child by using popen. Child process will implement the command with file as an input then sent to parent. Parent simply display output from the child on standard output.

Ex)
./Task4 cat Task4.c
./Task4 sort Task4.c
./Task4 cat Task4.c | grep main