

CPSC 351 Project 2 (100 points)

Due by Sunday, April 2, 11:59 PM

This project can be done individually or in a group of maximum **3** people. For a group of 2 or 3 members, each of the group members will have to submit, even though the code will be the same for all the members of the group. All members of the group will get the same grade. If one group member does not submit, that person gets 0. Indicate in an additional .txt file, the names and email ids of members in the group. If working individually, indicate in the .txt, your name and email id.

Printing Alpha Numeric Characters

Write a C++ program **alphanumeric.cpp (using the pthread library; Do not use C++ threads)** that accepts a phrase of unspecified length as a command line argument. For example:

```
prompt$: ./alphanumeric "Operating Systems 351 begins at 4:00pm or 7:00pm"
```

The main() in this program should read the phrase from the terminal as input. This phrase should be saved into a global variable. This phrase or its parts can be directly accessed from the main() and / or from the threads. The main() has to create two concurrent threads. The threads should print as follows:

The alpha thread should print the words that begin with an alphabet.

The numeric thread should print the words that begin with a number.

Note that the main thread (running main()) should not print anything itself, the output should be printed by the threads that it creates. **The order of words in the phrase should not be changed in printing the characters.** Your program should work for any phrase of any reasonable length, not just the one given in the example.

Given the example run below:

```
$: ./alphanumeric "Operating Systems 351 begins at 4:00pm or 7:00pm"
```

The output of your program should look like:

```
alpha: Operating
alpha: Systems
numeric: 351
alpha: begins
alpha: at
```

numeric: 4:00pm
alpha: or
numeric: 7:00pm

Enable multiple processors in the virtual machine, for example: 4.

Goal 1: From the terminal, the program should read in the phrase. This phrase should be saved into a global variable.

Goal 2: The main should create 2 concurrent threads: one thread for printing words that begin with an alphabet, and the second thread for printing words that begin with a number. Each thread should call their own thread function: alpha and numeric respectively. Each thread should be sent their name as an argument through the thread function: alpha or numeric: this is what they will print in the output as shown above.

You may encounter race conditions.

To solve it:

You are NOT allowed to use ANY pthread library synchronization primitives including but not limited to: semaphores, mutexes, mutex locks such as pthread_mutex_lock and pthread_mutex_unlock and conditional variables such as pthread_cond_wait and pthread_cond_signal etc. You should solve that race condition (if any) using another way, such as C++ language constructs.

You are also not allowed to use sleep() or sched_yield()

Goal 3: You will have to solve any race conditions using **C++ language constructs**, to simulate the working of locks.

Goal 4: If you split the input string into a set of words, say a vector, then such an object should be global. You should simultaneously be iterating over the vector in each of the alpha and the numeric thread functions. The index or the iterator used for iterating should be global.

On canvas, we went over some example programs for threads. Take insights from these. The Concurrent threads examples can serve as something to start with. Just using basic programming skills and interplay of loops and if conditions, checking on some variables will help you solve this problem.

FILES TO BE SUBMITTED

- 1) alphanumeric.cpp
- 2) .txt file will name/s and email id/s

Please read in entirety. All specifications need to be adhered to. Where it says for example: output must be printed by which thread, then that must be followed. Mutex locks from pthreads library cannot be used, then that must be followed. And so on.

Anything not specified in the description, there you are free to choose.

MORE USEFUL INFORMATION

The main() function (thread) should **ONLY** create 2 threads : one thread calls the alpha function, the other thread calls the numeric function.

If you have a loop that is creating 2 threads per iteration: when the loop runs n times: $2n$ threads are created.

But the requirement states: "The main() has to create two threads running functions (alpha and numeric respectively)"

Blurb for your resume

Use your GitHub account as a ready-to-show portfolio of your programming projects to potential employers.