**ECE 3220 Lab 8**

Implement a C++ program that is explained below using UNIX operating system.

In this Lab, we want you to implement a multi-threaded program that creates a file and performs a word (sub-string) search and count program. First your program will need to create multiple threads whose tasks are threefold.

1. Create a file
2. Write the file (fill in the file with random letters without blanks)
3. Search and count the given **sub-strings**.

**Detailed Explanation of Tasks**

1. **Create a file**

Your program should create an empty file by a thread. Name the file as “**big.txt**”.

1. **Write the file**

Your program should fill in the file by a thread with random letters. The letters can be lower or upper case ([a-zA-Z]). Your file must be filled with 250.000.000 letters that are concatenated to each other.

Example of file content

“.....

lfnqopSrehglaKQIAksdlfkahQWMNsldfkqMerhPQIBMAqnlgkaIASEGKCXJavjqovAJGi

KnvhajwIGJahgIqAVHBqhvahsJqbJZubENguaOSasdiQNvuasJhasqHMNKzorBJAaarEbh

.....”

1. **Search and count the given sub-strings**

Your thread/s should search for 5 sub-strings with different sizes. The sub-strings you are responsible to find are;

* x
* os
* cpu
* disk
* cache

However, your search shouldn’t be case sensitive. If a sub-string “os/OS/Os/oS” is found, you should increment the sub-string count for “os”.

Run your program with different **numbers of threads** and **how many of the threads are responsible of a specific task**. Running your program with only “n” number of threads will not gain you points.

**Prepare a report containing a table that shows the execution time of each configuration (how many threads are used for tasks). And your comments on the results.**

And answers to these questions;

* What is the optimal number of threads for your computer? Why do you think it is the optimal number?
* When the use of threads does become necessary or unnecessary? Why? Explain your answer.