Programming Assignment 2: Threads
CSE3320 001/003/900
Due: Wednesday March 24nd, 2021 5:30PM CST

Description

The purpose of this project is to practice threaded programming by solving various problems. The objectives of this project is to learn:

1. Get familiar with threading.
2. How to use mutexes, semaphores, and conditional variables in a threading library.
3. How to design efficient solutions for mutual exclusion problems.

Project submission

For this assignment, create a gzipped file containing the following items, and submit it through Canvas.

1. A report that briefly describes how you solved the problems and what you learned for each part. For each part of the assignment your report must:
    1. Explain the problem and identify evaluation metrics for experiments
    2. Explain your choice of threading libraries
    3. Explain the design of the experiment and develop programs for evaluation
    4. Detail your collected experimental results
    5. Analyze, graph, and interpret experimental results and draw conclusions
2. The source code files containing your test cases for parts 1-3.

Part 1 (40 pts)

Given two character strings s1 and s2. Write a threaded program to find out the number of substrings, in string s1, that is exactly the same as s2. For example, suppose number_substring(s1, s2) implements the function, then number_substring("abcdab", "ab") = 2, number_substring("aaa", "a") = 3, number_substring("abac", "bc") = 0. The

size of s1 and s2 (n1 and n2) as well as their data are input by users. Assume that n1 mod NUM_THREADS = 0 and n2 < n1/NUM_THREADS.

The provided program is a sequential solution of the problem. read_f() reads the two strings from a file named “string.txt and num_substring() calculates the number of substrings.

You can find the source for this assignment on GitHub at: https://www.github.com/CSE3320/Substring-Assignment .

Write a parallel program using a threading library based on this sequential solution. You may use pthreads, C++ threads, Intel TBB, or another threading library. You must provide your rationale for choosing the library AND provide instructions for setting up the VM to use your library if additional libraries need to be installed.

HINT: Strings s1 and s2 are stored in a file named “string.txt”. String s1 is evenly partitioned for NUM_THREADS threads to concurrently search for matching with string s2. After a thread finishes its work and obtains the number of local matchings, this local number is added into a global variable showing the total number of matched substrings in string s1. Finally this total number is printed out. You can find an example of the “string.txt” in the attached source code.

Part 2 (35 pts)

Use condition variables to implement the producer-consumer algorithm. Assume two threads: one producer and one consumer. The producer reads characters one by one from a string stored in a file named “message.txt”, then writes sequentially these characters into a circular queue. Meanwhile, the consumer reads sequentially from the queue and prints them in the same order. Assume a buffer (queue) size of 5 characters. Write a threaded program using condition variables.

Part 3 (25 pts)

Write two micro-benchmarks to quantify the costs of context switch between multiple processes and multiple threads. Learn from the lat_ctx benchmark from the lmbench benchmark suite about how to measure the context switch cost between multiple processes. Understand how this benchmark works and adapt it to write your own benchmarks. Configure the number of vCPUs to more than one but run all processes/threads on one vCPU. Vary the number of processes/threads to study if the context switch cost varies. Below is a related work on quantifying the cost of context switch. https://www.cs.rochester.edu/~kshen/papers/expcs2007.pdf

Tips

(1) Shutdown your VM and change to VM setting to use 4 vCPUs.

(2) Verify that your VM has 4 vCPUs:

```
$ cat /proc/cpuinfo
```
You should have 4 CPUs (processor: 0-3).

Administrative

This assignment must be coded in C or C++. Any other language will result in 0 points. You programs will be compiled and graded on a CSE 3320 VM. Please make sure they compile and run on theVM before submitting them. Your report MUST provide compilation steps.

Your program is to be turned in via Canvas. Submission time is determined by the Canvas system time. You may submit your programs as often as you wish. Only your last submission will be graded.

There are coding resources and working code you may use on Canvas and in the course github repository at: https://github.com/CSE3320/Substring-Assignment . You are free to use any of that code in your program if needed. You may use no other outside code. You MAY use any threading library available for C or C++/

Academic Integrity

This assignment must be 100% your own work. No code may be copied from friends, previous students, books, web pages, etc. All code submitted is automatically checked against a database of previous semester's graded assignments, current student's code and common web sources. By submitting your code on Canvas you are attesting that you have neither given nor received unauthorized assistance on this work. Code that is copied from an external source or used as inspiration, excluding the course github or Canvas, will result in a 0 for the assignment and referral to the Office of Student Conduct.