CSC 410 Final Exam – Final Part 1

Part 1 Questions 1-7

Instructions: (1) Questions are provided in this file. (2) You should plan out your thoughts and review your answers. (3) Submit answers to the D2L Quiz Final1.

Final Part 1 has a 150 minute time limit to answer all of the short-answer/essay questions.

Guidelines: the work should be your own; some unrelated information pasted from another source is not appropriate.

Be precise, brief, and thorough. Demonstrate your understanding & mastery of the material. Don’t just add fluff.

1) Discuss the terms/measures of Speed-up and Efficiency as we discussed. What do they mean? How are they different. How are they related? What consideration would you give to these concepts as you develop a concurrent solution. Explain. **( 8 pt )**

2) Amdahl’s Law provides guidance on potential speed-up as more processes are dedicated to a problem. On this site (<https://en.wikipedia.org/wiki/Amdahl's_law> ) you’ll find an informative graph and a number of examples. In your own words, provide an explanation of the meaning of the graph and ultimately the implications of Amdahl’s formula. You don’t need to restate the Law, explain what the graph shows, how to interpret the concept….  **( 8 pt )**

3) We’ve used a MUTEX-Lock many times. But let’s talk about a potentially catastrophic situation with this solution. I’ll use pseudocode here rather than precise notation. For a MUTEX m, we use Lock(m) and Unlock(m) to interact. This is frequently used to implement a critical section: Lock(m). Critical Code. Unlock(m).

We believe this solution to be sufficient. But, what happen in the extremely unlucky case that two processes arrive at exactly & precisely the same moment. This is not likely, but probabilistically can happen from time to time. Even more coincidentally, let’s assume the lock is OPEN so that both processes would find the lock available. What happens? **( 8 pt )**

4) For moderately sized problems, why will a PThread solution perform better than a MPI solution? What problem characteristics might make MPI the best solution platform? **( 8 pt )**

5) We learned about the OpenMP reduce operation: reduction( + : result ). Explain how you could implement this same behavior without using the reduce operator. **( 8 pt )**

6) Requirements for Critical-Section Solutions **( 8 pt )**

*Explain or restate each of the 3 properties below in your own works. Perhaps you can also provide some sort of real-world analogy to illustrate the meaning.*

**Mutual Exclusion.** If process P is executing in its critical section (CS), then no other process can execute in its CS.

**Progress**. If no process is executing in its CS and there exist some processes that wish to enter their CS, then the selection of the process that will enter the CS next cannot be postponed indefinitely.

**Bounded Waiting.** There exists a bound on the number of times that other processes are allowed to enter their CS after a process has made a request to enter its CS and before that request is granted.

7) In MPI, illustrate the code fragment (program in your cs.dsunix account not needed) that would allow a process to accept a message (consisting of a single int) from anyone; print out the value and also the id of the process that sent the information. Explain each step of the process - don't just give me a few lines of code, explain it. **( 8 pt )**