Assignment #2

Due Date: November 7, 2019

Total: 100 marks

Processes/threads and synchronization

Each program is worth 50 marks (10 marks for submission, comments, readme, makefile, and demonstration using a text file that it works).

1. Write a multi-threaded application for multiplying computing your CS 3520 mark: one thread reads data, and another one computes the multiplication, using multiple threads. The third thread collects the result and displays it. You should use shared memory space within the main program to communicate between threads. Marking scheme:
   (a) creating the right number of threads: 10 marks
   (b) creating the threads : 10 marks
   (c) computing the right result: 10 marks
   (d) overall program: 10 marks
2. Create a client-server configuration for the previous program where the server computes the mark, but accepts an arbitrary number of clients requesting the final mark from the server. Requirements:
   (a) On the communication protocol
      i. The server should be able to talk with the right client using shared memory.
      ii. If the buffer for shared memory is full, the program sending data should wait.
      iii. If the buffer for shared memory is empty, the program reading data should wait.
   (b) On computation on server side:
      i. each mark is multiplied by a percent, then the result is added to the final mark. Thus, if your marks are 50, 60, 70, 80, and 90 and the percentages are 10, 20, 30, 30, and 10, your final mark is $\frac{50 \cdot 10}{100} + \frac{60 \cdot 20}{100} + \frac{70 \cdot 30}{100} + \frac{80 \cdot 30}{100} + \frac{90 \cdot 10}{100} = 62$, i.e., $FM = \sum_{i=1}^{n} a_i \cdot p_i$.
      ii. Each multiplication is computed in a distinct thread.
      iii. The sum is computed in another thread.
   (c) Marking scheme:
      i. creating the right number of threads: 10 marks
      ii. creating the threads : 10 marks
      iii. computing the right result: 10 marks
      iv. overall program: 10 marks
3. Write a multi-threaded application to solve a Sudoku problem, on an $n^2 \times n^2$ grid (maximum 25x25). See more details in Project 1, page 197, from the textbook on verifying a solution.

   Marking scheme:

(a) creating the right number of threads: 10 marks
(b) programming the threads : 10 marks
(c) computing the right result: 10 marks
(d) overall program: 10 marks

Note:

- You can try to use the following alphabets for:
  (a) $n = 2$: $\{1, 2, 3, 4\}$
  (b) $n = 3$: $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ – this is the classical version of the game.
  (c) $n = 4$: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f\}$ or $\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p\}$
  (d) $n = 5$ $\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y\}$
- The n letters of the alphabet must occur only once in each
  - of the n^2 squares
  - line
  - column.

Example

2	9	5	7	4	3	8	6	1
4	3	1	8	6	5	9	2	7
8	7	6	1	9	2	5	4	3
3	8	7	4	5	9	2	1	6
6	1	2	3	8	7	4	9	5
5	4	9	2	1	6	7	3	8
7	6	3	5	2	4	1	8	9
9	2	8	6	7	1	3	5	4
1	5	4	9	3	8	6	7	2

Marks for this problem cannot exceed the following bounds computed of the maximum:

(a) 60% for one size, 70% for two, 80% for three.
(b) For just verifying a solution you get 60% of the previous maximum.