Operating Systems–2
Programming Assignment 1: Validating Sudoku Solution
Submission Deadline: 29th January 2025, ~~9:00~~ 11:00 pm

Goal:- This assignment aims to design a multithreaded program which will determine whether a solution to a Sudoku puzzle is valid or not.

Details:- This assignment is Programming Projects 1 given in the 10th edition of the textbook (pdf page 262). Please read the details given in the book carefully. Solve this problem using **pthreads**.

For a given N X N sudoku, where N is a square, there will be N rows, N columns and N grids to be validated as shown below.

Validation Condition:- For a N X N sudoku, the validation condition is:

- Each row contains unique values from 1 - N.
- Each column contains unique values from 1 - N.
- Each of the nXn sub-grids, contains a unique value from 1 - N where, $N = n^2$

For example:- For 4X4 sudoku, you need to check each row, each column and each of the four 2X2 subgrids have values from 1 to 4.

Algorithm Details: Chunk and Mixed.
To implement the above-mentioned tests, you will have to divide the threads into three sets (approximately equally sized) for validating the given sudoku: **rows, columns and sub-grids** as mentioned above.

Next, you will have to implement two methods namely **chunk** and **mixed** described as follows.

Chunk method: In the chunk method, a thread is assigned consecutive rows, columns or subgrids to validate. For example, if the number of rows is R and the number of threads for validating rows is K1, the chunk size p is calculated as p=R/K1. Thread 1 will validate rows 1 to p, Thread 2 will validate rows p+1 to 2p, Thread 3 will validate rows 2p+1 to 3p and so on. The same allocation method is applied for columns and subgrids.

Mixed method: Here each thread, instead of assigning consecutive rows, columns or subgrids, the workload is distributed evenly among the threads in a cyclic manner. For example, if the number of rows is R and the number of threads for validating rows is K_1, Thread 1 will validate rows 1, K_1+1, $2K_1$+1,… Thread 2 will validate rows 2, K_1+2, $2K_1$+2,… and so on. This pattern continues for all threads.

Input File:- The input to the program will be a file, named input.txt
The first line of the input file will contain two values **K** and **N,** where K is the number of threads and N is the dimensional value of N X N sudoku.

From the second line, the sudoku of dimension N X N will be present, with each row in a new line. This sudoku can be generated through a python script.

A sample input file will be like this :-

4 4
1 3 4 2
2 4 1 3
4 2 3 1
3 1 2 4

In the above example, K = 4 and N = 4.

Output File:- For ease of understanding, you need to generate an output file in which you will store the details of the execution of each thread and at last it should output whether the given sudoku is valid or invalid. A sample output would be something like this:-

Thread 1 checks row 1 and is valid.
Thread 2 checks column 2 and is valid.
Thread 3 checks grid 1 and is valid.
.
.
.
Thread i checks the grid m and is valid.
.
Thread j checks the row n and is invalid.

Note1: As mentioned in the class, having timestamps here will help the TAs in verifying the correctness of your program.

Sudoku is invalid.
The total time taken is 25.05 microseconds.

Report Details:- As a part of this assignment you have to perform two experiments **without considering early termination**. This means you must check all the rows, columns and subgrids completely and then decide if the Sudoku is valid or not. You must perform the following experiments:

1. Experiment 1: In this experiment, you have to keep the number of threads constant say 8 and compare the time taken to validate the sudoku by varying the size as follows: 9X9, 16X16, 25X25, 36X36, 49X49 and 64X64. Thus, in this experiment, the size of the sudoku will be on the x-axis as described above and the y-axis will show the time taken.

   Note2: Some students informed me that even for 64X64 size matrix, sequential algorithms seems to be doing better than parallel versions. So, if you wish, you can test with larger matrix sizes such that the parallel versions do better than serial version and

note your observations in the report.

2. In the second experiment, you have to keep the size of the sudoku constant 36X36 and compare the performance by varying the number of threads from 4 to 32 in multiples of 2: 4, 8, 16 and 32. In this experiment, the number of threads will be on the x-axis and the y-axis will show the time taken.

   Note3: Similar to the Note2 in point1, you can test with larger matrix size such that the parallel version does better than serial and note your observations in the report.

Note that, there will be three curves in the graph for each experiment: mixed, chunk and sequential. Finally, you must also give an analysis of the results while explaining any anomalies observed in the report.

Extra Credit - Early Termination: Please implement the logic for early termination among the threads for extra credit. The program terminates immediately if some thread finds that the sudoku is not valid. The program does not have to wait for all the threads to terminate naturally.

As mentioned in the class, please use signals for early termination and not use any synchornization tools.

Submission Format*:-* You have to upload: (1) The source code in the following format: Assgn1Src-<RollNo>.c (2) Input file as inp.txt (3) Readme: Assgn1Readme-<RollNo>.txt, which contains the instructions for executing the program (4) Report: Assgn1Report-<RollNo>.pdf. Name the zipped document as Assgn1-<RollNo>.zip

Please follow this naming convention. Otherwise, your assignment will not be graded.

Grading Policy:- The policy for grading this assignment will be -
(1) Design as described in the report and analysis of the results: 50% (2) Execution of the tasks based on the description in the readme: 40% (3) Code documentation and indentation: 10% (4) Extra Credit: 10%

Please note:

1. All assignments for this course have a late submission policy of a penalty of 10% each day after the deadline of six days. After that, it will not be evaluated.
2. *All submissions are subjected to plagiarism checks. Any case of plagiarism will result in F grade. You can refer to the department's website for more information regarding the anti-plagiarism policy.*