

Operating Systems–2

Programming Assignment 2: Validating Sudoku Solution

Submission Deadline: 3rd February 2023, 9:00 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, P-25 of the book). Please read the details given in the book carefully. Solve this problem using **OpenMP** and **Pthreads** to *compare their performance*.

For a given $N \times N$ 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 $N \times N$ sudoku, the validation condition is:

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

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

Input File: The input to the program will be a file, named input.tex.

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 $N \times N$ sudoku.

From the second line, the elements of the sudoku 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: To verify the correctness, you must 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.

Sudoku is invalid.

The total time taken is 25.05 microseconds.

Report Details:- As a part of this assignment you have to prepare a report in which you will compare the performance of OpenMP and Pthreads. You must run your program multiple times to compare the performance in terms of the time it takes and display the result in the form of a graph. You will have the following plots:

1. In the first experiment, the x-axis will be the size of the sudoku varying from 9*9 to 100 * 100 (8 points). The y-axis will show the time taken. Keep the total number of threads fixed at 16 for this experiment.
2. For the second experiment, the x-axis will consist of the number of threads varying from 2 to 64 in the powers of 2 (5 points). The y-axis will again show the time taken. Keep the sudoku size to be fixed at 25 * 25 for all the experiments.

You must average each point in the graph plot by 5 times. Note that, there will be two curves in each of these plots: one for OpenMP and the other for Pthreads. Finally, you must also give an analysis of the results while explaining any anomalies observed in the report.

Extra Credit: For extra credit, you can implement early termination under the following condition: if a thread discovers that the given input sudoku file is incorrect, it can inform other threads to terminate early.

Please note that early termination cases should not be considered for the graph plots for report generation.

Submission Format: You have to upload: (1) The source code in the following format: Assgn2Src-<RollNo>.c (2) Readme: Assgn2Readme-<RollNo>.txt, which contains the instructions for executing the program. (3) Report: Assgn2Report-<RollNo>.pdf. Name the zipped document as: Assgn2-<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%.

Please note:

- 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.
- *All submissions are subjected to plagiarism checks. Any case of plagiarism will be dealt with severely. You can refer to Institute's website for more information regarding the anti-plagiarism policy.*