# Operating System

# COM301P

## Mid-Sem

By :

Sai Kaushik S

CED18I044

**Develop a SUDOKU solution validator that checks if in a 9 * 9 matrix (i) each column contains the digits 1 thru 9 using one thread (ii) each row contains the digits 1 thru 9 using another thread and (iii)9 threads to check the 9 possible 3*3 subgrids to contain the digits 1 thru 9. Compare the efficiency of the threaded version over its equivalent serial version.**

## Code:

```
// Include the required libraries
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <pthread.h>

// 2D pointer for the matrix
int** matrix;

bool isValid = true;

// Structure for the data
struct data{
   int row;
   int column;
};

// Function declarations
void* rowCheck(void* params);
void* columnCheck(void* params);
void* matrixCheck(void* params);

// Main driver function
int main(){
   // Initialize the array
   matrix = (int**)malloc(sizeof(int*)*9);
   for(int i = 0; i < 9; i++)
       matrix[i] = (int*)malloc(sizeof(int)*9);
   // Scan for the matrix
   for(int i = 0; i < 9; i++){
       for(int j = 0; j < 9; j++){
           a:
           // If the input is greater than 9 or less than 1, take the input again
           printf("Enter the number [%d][%d]: ", i + 1, j + 1);
```

```
            scanf("%d", &matrix[i][j]);
            if(matrix[i][j] < 1 || matrix[i][j] > 9){
                printf("The number should be in the range [1, 9].\n");
                goto a;
            }
        }
    }
    pthread_t tid[27];
    int x = 0;
    for(int i = 0; i < 9; i++){
        for(int j = 0; j < 9; j++){
            // Get the current index
            struct data params;
            params.row = i;
            params.column = j;
            // If the index mod 3 is 2, ie there are end index of a 3X3 matrix
            if (i % 3 == 2 && j % 3 == 2)
                // Check if the condition is satisfied
                pthread_create(&tid[x++], NULL, matrixCheck, (void*)&params);
            if(j == 0){
                // Check the column condition and row condition
                pthread_create(&tid[x++], NULL, rowCheck, (void*)&params);
                pthread_create(&tid[x++], NULL, columnCheck, (void*)&params);
            }
        }
    }
    // Wait for the threads to complete
    for(int i = 0; i < 27; i++)
        pthread_join(tid[i], NULL);
    // If the sudoku is not valid
    if(!isValid)
        printf("The given sudoku matrix is invalid.\n");
    // If the sudoku is valid
    else
        printf("The given sudoku matrix is valid.\n");
    exit(EXIT_SUCCESS);
}

// A function for row condition check
void* rowCheck(void* params){
    struct data* temp = (struct data* )params;
    int* sum = (int* )malloc(sizeof(int));
    // Initialize sum to 0
    *sum = 0;
    // Add the elements in the row
```

```
    for(int i = 0; i < 9; i++)
        *sum += matrix[temp->row][i];
    // If the sum is not 45, sudoku is invalid
    if (*sum != 45)
        isValid = false;
    // Exit the threads
    pthread_exit(NULL);
}

// A function for column condition check
void* columnCheck(void* params){
    struct data* temp = (struct data* )params;
    int* sum = (int* )malloc(sizeof(int));
    // Initialize sum to 0
    *sum = 0;
    // Add the elements in the column
    for(int i = 0; i < 9; i++)
        *sum += matrix[i][temp->column];
    // If the sum is not 45, sudoku is invalid
    if (*sum != 45)
        isValid = false;
    // Exit the threads
    pthread_exit(NULL);
}

// A function for 3X3 matrix condition check
void* matrixCheck(void* params){
    struct data* temp = (struct data* )params;
    int* sum = (int* )malloc(sizeof(int));
    // Initialize sum to 0
    *sum = 0;
    // Add the elements in the matrix
    for(int i = temp->row; i > temp->row - 3; i--)
        for(int j = temp->column; j > temp->column - 3; j--)
            *sum += matrix[i][j];
    // If the sum is not 45, sudoku is invalid
    if (*sum != 45)
        isValid = false;
    // Exit the threads
    pthread_exit(NULL);
}
```

**Output:**

```
thegamingbot@pop-os:~/Downloads/sem-5/OS/MidSem$ gcc sudokuChecker.c -o sudokuChecker -pthread
thegamingbot@pop-os:~/Downloads/sem-5/OS/MidSem$ ./sudokuChecker
Enter the number [1][1]: 6
Enter the number [1][2]: 2
Enter the number [1][3]: 4
Enter the number [1][4]: 5
Enter the number [1][5]: 3
Enter the number [1][6]: 9
Enter the number [1][7]: 1
Enter the number [1][8]: 8
Enter the number [1][9]: 7
Enter the number [2][1]: 5
Enter the number [2][2]: 1
Enter the number [2][3]: 9
Enter the number [2][4]: 7
Enter the number [2][5]: 2
Enter the number [2][6]: 8
Enter the number [2][7]: 6
Enter the number [2][8]: 3
Enter the number [2][9]: 4
Enter the number [3][1]: 8
Enter the number [3][2]: 3
Enter the number [3][3]: 7
Enter the number [3][4]: 6
Enter the number [3][5]: 1
Enter the number [3][6]: 4
Enter the number [3][7]: 2
Enter the number [3][8]: 9
```

**Enter the number [3][9]: 5**
**Enter the number [4][1]: 1**
**Enter the number [4][2]: 4**
**Enter the number [4][3]: 3**
**Enter the number [4][4]: 8**
**Enter the number [4][5]: 6**
**Enter the number [4][6]: 5**
**Enter the number [4][7]: 7**
**Enter the number [4][8]: 2**
**Enter the number [4][9]: 9**
**Enter the number [5][1]: 9**
**Enter the number [5][2]: 5**
**Enter the number [5][3]: 8**
**Enter the number [5][4]: 2**
**Enter the number [5][5]: 4**
**Enter the number [5][6]: 7**
**Enter the number [5][7]: 3**
**Enter the number [5][8]: 6**
**Enter the number [5][9]: 1**
**Enter the number [6][1]: 7**
**Enter the number [6][2]: 6**
**Enter the number [6][3]: 2**
**Enter the number [6][4]: 3**
**Enter the number [6][5]: 9**
**Enter the number [6][6]: 1**
**Enter the number [6][7]: 4**
**Enter the number [6][8]: 5**
**Enter the number [6][9]: 8**
**Enter the number [7][1]: 3**
**Enter the number [7][2]: 7**
**Enter the number [7][3]: 1**

**Enter the number [7][4]: 9**
**Enter the number [7][5]: 5**
**Enter the number [7][6]: 6**
**Enter the number [7][7]: 8**
**Enter the number [7][8]: 4**
**Enter the number [7][9]: 2**
**Enter the number [8][1]: 4**
**Enter the number [8][2]: 9**
**Enter the number [8][3]: 6**
**Enter the number [8][4]: 1**
**Enter the number [8][5]: 8**
**Enter the number [8][6]: 2**
**Enter the number [8][7]: 5**
**Enter the number [8][8]: 7**
**Enter the number [8][9]: 3**
**Enter the number [9][1]: 2**
**Enter the number [9][2]: 8**
**Enter the number [9][3]: 5**
**Enter the number [9][4]: 4**
**Enter the number [9][5]: 7**
**Enter the number [9][6]: 3**
**Enter the number [9][7]: 9**
**Enter the number [9][8]: 1**
**Enter the number [9][9]: 6**
**The given sudoku matrix is valid.**
**thegamingbot@pop-os:~/Downloads/sem-5/OS/MidSem$ ./sudokuChecker**
**Enter the number [1][1]: 1**
**Enter the number [1][2]: 2**
**Enter the number [1][3]: 3**
**Enter the number [1][4]: 4**

**Enter the number [1][5]: 5**
**Enter the number [1][6]: 6**
**Enter the number [1][7]: 7**
**Enter the number [1][8]: 8**
**Enter the number [1][9]: 9**
**Enter the number [2][1]: 1**
**Enter the number [2][2]: 2**
**Enter the number [2][3]: 3**
**Enter the number [2][4]: 4**
**Enter the number [2][5]: 5**
**Enter the number [2][6]: 6**
**Enter the number [2][7]: 7**
**Enter the number [2][8]: 8**
**Enter the number [2][9]: 9**
**Enter the number [3][1]: 1**
**Enter the number [3][2]: 2**
**Enter the number [3][3]: 3**
**Enter the number [3][4]: 4**
**Enter the number [3][5]: 5**
**Enter the number [3][6]: 6**
**Enter the number [3][7]: 7**
**Enter the number [3][8]: 8**
**Enter the number [3][9]: 9**
**Enter the number [4][1]: 1**
**Enter the number [4][2]: 2**
**Enter the number [4][3]: 3**
**Enter the number [4][4]: 4**
**Enter the number [4][5]: 5**
**Enter the number [4][6]: 6**
**Enter the number [4][7]: 7**
**Enter the number [4][8]: 8**

**Enter the number [4][9]: 9**
**Enter the number [5][1]: 1**
**Enter the number [5][2]: 2**
**Enter the number [5][3]: 3**
**Enter the number [5][4]: 4**
**Enter the number [5][5]: 5**
**Enter the number [5][6]: 6**
**Enter the number [5][7]: 7**
**Enter the number [5][8]: 8**
**Enter the number [5][9]: 9**
**Enter the number [6][1]: 1**
**Enter the number [6][2]: 2**
**Enter the number [6][3]: 3**
**Enter the number [6][4]: 4**
**Enter the number [6][5]: 5**
**Enter the number [6][6]: 6**
**Enter the number [6][7]: 7**
**Enter the number [6][8]: 8**
**Enter the number [6][9]: 9**
**Enter the number [7][1]: 1**
**Enter the number [7][2]: 2**
**Enter the number [7][3]: 3**
**Enter the number [7][4]: 4**
**Enter the number [7][5]: 5**
**Enter the number [7][6]: 6**
**Enter the number [7][7]: 7**
**Enter the number [7][8]: 8**
**Enter the number [7][9]: 9**
**Enter the number [8][1]: 1**
**Enter the number [8][2]: 2**
**Enter the number [8][3]: 3**

**Enter the number [8][4]: 4**
**Enter the number [8][5]: 5**
**Enter the number [8][6]: 6**
**Enter the number [8][7]: 7**
**Enter the number [8][8]: 8**
**Enter the number [8][9]: 9**
**Enter the number [9][1]: 1**
**Enter the number [9][2]: 2**
**Enter the number [9][3]: 3**
**Enter the number [9][4]: 4**
**Enter the number [9][5]: 5**
**Enter the number [9][6]: 6**
**Enter the number [9][7]: 7**
**Enter the number [9][8]: 8**
**Enter the number [9][9]: 9**
**The given sudoku matrix is invalid.**
**thegamingbot@pop-os:~/Downloads/sem-5/OS/MidSem$**