Project 3 - Sudoku Validator

Jordan Lu | jjl4mb

CS 4414 - Operating Systems

09/29/2016

The first step to solving this problem was to determine how to validate a sudoku puzzle. From the instructions, we first check every row to make sure that the numbers 1 - 9 are in every row. We then do the same for every column and 3x3 subsection. The input file we are given contains 81 numbers, all separated by spaces. We first parse this code, splitting the string by spaces, and storing them in a 9x9 integer array in row column fashion. After that, we write 3 different methods: one to check every row, one to check every column, and one to check subsections. These functions must return void pointers, and also have them as parameters in order to create a thread with these functions. We first make a thread for the two independent functions, checking the rows and checking the columns. Next, we make 9 threads in a for loop to cover every 3x3 subsection in sudoku. We use pthread_create to create the threads, and pthread_join to wait for these to finish. The functions of each thread will output the missing number in the row, column, or subsection. Thus, after every thread is finished running, we will have outputted all the missing numbers if there were any. We also have a global int variable that gets changed to 0 whenever a thread finds a number that is missing. Thus, after all the threads have run, we can check this variable, and print out whether or not the sudoku is valid or not.