HACETTEPE UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

BBM203

SOFTWARE LABORATORY I

ASSIGNMENT – 2

Student's Name: Onur Tolga Kesemen

Student's Number: 21228511

Due Date: 12.11.2017

Advisor: R.A Burçak Asal, Dr. Burcu Can, Dr. Sevil Şen, Dr. Adnan Özsoy

Software Usage

This program takes two input files and one output file name from command line as arguments. First input file specifies the number of clients and both clients and server's queue and stack sizes. Second input file includes the number of commands and commands itself. Program creates clients and server depends on first input and operates the commands in second input file than writes histories of all clients and server to the output file.

Sample Input And Output

Input File 1	Input File 2	Output File
5 2 1 4 2 2 2 1 2 1 2	20 A 3 I I 3 q S 3 G A 4 s A 2 p S 3 G S 3 G A 1 x O G G A 3 m S 4 G I 4 x A 1 x S 3 G S 2 G O G G O G G A 4 I A 2 o I 1 g	/* EMPTY */ p q I 3 m s 1 q 1 1 s 3

Software Design Notes

Problem

In this experiment we are expected to implement a basic client-server architecture with usage of stack and queue structures. There is four type of commands; A, I, S, O. These commands stands for operations on queue and stack structures of clients and server. Detailed description of these commands are in the experiment sheet. And there is three type of errors. ‘1’ stands for queue is full, ‘2’ means stack is full and ‘3’ means both stack and queue are empty. The program should be able to create clients and server upon first input file and then operate the commands in second input file. After all it should print the history of all clients and server to the output file.

Solution

In my algorithm, i wrote the stack and queue functions in seperate c files and created .h files for them. Because of it provides the simplicity of code. I did not seperated the main fucntion to multiple functions because it makes problem much bigger for me.

Program keeps the second input file in an array of ‘Command’ structure to gain more speed. But it loses memory a little more. It controls the commands with nested if clauses. And makes operations on clients and server.

Finally, the commands are finished, program prints histories of all clients and server to the output file to the given directory and name.