

HACETTEPE UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

BBM 203



Name: Emre

Surname: Tunç

Number: 21527401

E-mail: emrettunc@hotmail.com

Programming Language: C

Assignment 2

Aim

In this assignment, we are expected to design a basic client-server architecture. We will perform operations on server and clients according to input files. We are expected to use stack and queue structures.

Problem

We have two different input files. First input file gives us number of the clients, the stack and queue sizes of server and clients. Second input file gives use commands to implement. We will look at the character in the first column to understand what the command is. If first character is 'A', we add the character of the 3rd column to specific client's queue structure. The second column represents the client's number. If specific client's queue is full, we will give '1' error message. If first character is 'I', we add the character of the 3rd column to specific client's or server's stack structure. The second column represent the client's number or server. If specific client's or server's queue is full, we give '2' error message. If first character is 'S', firstly we will look specific client's stack structures. If there is an element in the stack, it is deleted from the client's stack and added to server's queue. If there is no element in the stack, we will look client's queue and queue is empty, we will give '3' error message. If there is an element in the queue , it is deleted form the client's queue

and added server's queue. During these additions if server's queue is full, we will ignore characters and give error message. In the send operations, the entire submitted character is added to the story of clients. If first character is 'O', first deleted the server's stack's elements and added them to server's history, Then deleted the server's queue's elements and added them to server's history. If no element in the server's queue and stack, we will give '3' error message.

Solution

Firstly, I read the given first input file and find number of clients, the stack and queue size of server and clients. Then I read the given second input file and find command operations. I create a struct for server and clients. Struct include stack and queue structure, size of stack and queue, history, size of history, information to specify the locations of elements in the stack and queue. I used to functions to adding and removing from stack and queue, control stack and queue fullness and emptiness, create server and clients information.

Functions

main():Read input files and stores information about commands and sizes. Selection of the necessary command is provided by if else statement. The results are printed in the output file.

makeArray():Creation of server' or client's stack and queue,filling information about server and clients.

stackisFull ():Control server's or client's stack. If the last element is equal to the size of the stack, the stack is full.

stackisEmpty():Control server's or client's stack. If the struct information top equal to -1, the stack is empty.

push():Incrementing the index of the last element in the stack and adding the new element to the stack.

pop():Decrement the index of the last element in the stack and delete the last element.

quequeisEmpty():Control server's or client's queue.If the struct information rear equal -1 and front equal -1, the queue is empty.

quequeisFull():Control server's or client's queue.If the struct information rear equal first element index and front equal last element index or if the element in the front comes after the element in the rear, queue is full.

equeue():Firstly,control queue's emptiness. If queue is empty, increase rear and front, add element in the queue.

If queue have element, increase last element index and adding new element in the queue.

dequeue():If there is only one element in the queue,it deletes the element and takes the index at the beginning. If there is more than one element,it deletes the first element.

appendArray():Append elements or error message in the server's or client's history.

Software Usage

I used Code::Blocks 16.01 for coding. I used different six I1.txt and I2.txt as input files. I used command line argument for output file.

Personal Comment

In this experiment, I developed stack and queue structure knowledge. I use stack and queue functions very well. This assignment helped me improve myself for coding.