Erdem Şimşek
2305423

EE444 HW1

Client_process

For client, a socket is created for IP address 127.1.1.1 and port number 6007. It is the user interface of the homework, and the prompt window gets the instruction. After getting instruction it sends all instruction to proxy and wait the respond of the proxy. After getting respond from proxy it displays on the screen with “Operation is done” message.

Proxy_process

In the proxy, two sockets created for client and server connections. Server socket is created over 127.2.2.2 and port number 6009.

Proxy gets the instruction from client and parse it to dataSplit array. dataSplit array is shown as follows

OP=PUT;IND=2,3;DATA=11,12;

dataSplit[0]	OP
dataSplit[1]	PUT
dataSplit[2]	IND
dataSplit[3]	2,3
dataSplit[4]	DATA
dataSplit[5]	11,12

The desired function is determined by dataSplit[1]. In every function, indices are also parsed by using the dataSplit[3]. For PUT function, integer data is obtained by parsing dataSplit[5].

In the proxy, there is a 2D array indicating index, data and counter for determining oldest value. The sample cache is given below. Counter is initialized as follows. The smallest value indicates the oldest updated value in cache. When it is updated from server or by placing PUT function, counter of this element is set as 5 and other counters are decremented by one.

	0th element	**1st element**	**2nd element**	**3rd element**	**4th element**
Index	7	8	2	3	5
Data	11	24	8	12	16
Counter	1	2	3	4	5

For all functions, firstly cache table is checked if there is a match with index the operation is executed and flag_found is updated. If it is not found in cache in other words, flag_found is 0, a simplified instruction sent to server. Server gets the simplified instruction and returns required data or update the indicated index. The message format between proxy and server is given below. “W”, “R” and “C” indicates the write, read and clear, respectively. The first value indicates the index, and the second value indicates the data itself.

W;1;13

R;8;0

C;

Get

Clear instruction clear cache and server table by using the simplified instructions.

BONUS

In addition to ADD instruction, AVG (average) instruction is added to homework. In addition to ADD function it keeps a counter that, indicate the number of data and by dividing the sum to counter, average can be obtained.

Server_process

In server, it gets simplified instruction from the proxy and takes action according to it. It keeps the main data table.

The server parses the simplified function at first. If the simplified instruction starts with R, it finds indicated index in main data table and return data. If the simplified instruction starts with W, it finds indicated index in main data table and update it with given data. If the simplified instruction starts with C, it clears whole table.

NOTE:

In the homework, server_process.m file doesn’t work properly, to demonstrate rest of of the homework, server_process.py is created. Both server_process.m and server_process.py files can be found in .zip file.