

HOW TO EXCECUTE FROM cmd:

First initialize cmd and make sure you are in the project directory. Once there, you write the following line: `java -classpath ./bin theSystem/QueueSimulation`.

Program information:

This software simulates different scenarios of serving policies and calculates the time it takes to finish, average waiting time, and the amount of people who took advantage time related of the line changing monitor.

The serving policies are the following:

Single Line Multiple Servers (SLMS)

Under this policy, there is only one waiting line and one or more service posts. Whenever a post is available, the first person in line, if any, will start to be served by the service person at the post. In the case in which there are more than one server available at a moment, then the first person in line will go to the available post having min index value among those available.

Multiple Lines Multiple Servers (MLMS)

Under this policy, each service post has its own waiting line (one line per server). Once a person enters a waiting line, that person cannot transfer to another line, even if one becomes empty. When a new person arrives looking for service, the person will choose the first line that has minimum number of persons waiting, as per the indexes identifying the corresponding service posts. If a line becomes empty, then the server there remains idle until a new customer arrives and selects that particular line. (This happens often in lines for cars to pay at the toll gate on a highway; once you are in a line, it is hard to change to another, even if one becomes empty.)

Multiple Lines Multiple Servers and Balanced Line Lengths (MLMSBLL)

This case is similar to the previous policy but a customer is not stuck to the line he/she initially chooses as in that one. This scheme includes a monitor (a person or a specialized device) that decides when, and to what line, a person already in one of the lines can transfer to. The initial selection is automatically done as in the previous policy. However, the monitor is there to control that persons move from one line to another in a controlled manner by implementing the following policy.

A person in a particular line can be transferred to a different line whenever the monitor allows. The monitor has the goal to always keep lines as *balanced in length* (number of persons waiting) as possible. The monitor can immediately determine when a particular line becomes shorter than the others, and that at least one person waiting in another line can benefit from being transferred to that shorter line. In that case, among all those that would benefit from the transfer, the monitor always selects the one which arrived first. The person being selected cannot reject the transfer. Whenever a transfer is possible, the transfer has priority over new arrivals. If a destination line for a transfer has more than one option (at least two lines have become shorter and at least one customer can benefit from being transferred to any one of those lines), the monitor chooses the one closer to the right (modulo the number of service posts from the line where the person to be transferred currently is). More than one transfer can happen at the same time; in that case, the monitor considers the candidates in the same order as they were in the input file.

Multiple Lines Multiple Servers and Balanced Waiting Times (MLMSBWT)

This scheme is similar to the previous two. Here, there is a monitor too, many servers and one waiting line per server. No line crossing is allowed. But in this case, the monitor decides which line the new arriving customer has to go to. The decision is based on the total expected time on each line. The new customer will be assigned to the first line having minimum total waiting time at that moment. In case of ties, the line with minimum index wins. To determine the expected time, the monitor always keeps, for each line, the sum of the service times of all those persons in the line, as well as the remaining time for service of the person who is being served at the moment, if any.

THE INPUT:

This software reads an initial file containing a collection of files to be analyzed, these files to be analyzed are the test cases. These test cases names are then accessed and inside each is found a collection of number tuples, recognized as “personas”. Each case is run in 3 different situations: 1, 3, and 5 clerks. These 3 situations are tested for each of the service policies.

THE OUTPUT:

A new file is created for each of the input cases and is deposited in the same inputFile directory. Each of these files contains all the results of every case, for every policy, for the specified set of “personas”.

Example output:

Amount of clients: 3

SLMS 1:74 24.66 0

SLMS 3:51 0.0 0

SLMS 5:51 0.0 0

MLMS 1:74 72.66 0.0

MLMS 3:51 72.66 0.0

MLMS 5:51 72.66 0.0

