Technical University of Cluj-Napoca

Fundamental Programming Techniques

Laboratory – Assignment 1

Queues simulator

Logo, company name

Description automatically generated

Teacher: prof. Ioan Salomie

Teacher Assistant: Ciprian Adrian Stan

Student: Itu Anca

Group: 30422

1. Assignment objective

Design and implement a simulation application aiming to analyze queuing based systems for determining and minimizing clients’ waiting time.

Queues are very often used in real world domains. They have the main objective to place items one after another and process them one by one in the order of their arrival. In our case, the purpose is to provide a place for a client to wait before receiving a service. Queue management is a set of principles aiming to control the customer flow by organizing the queues. The waiting time of each client has to be minimized and this can be done either by adding more servers or by achieving an optimal waiting time. The first solution would imply undesired and unnecessary costs for the service suppliers.

The application should simulate the following process: a series of clients have to wait in line before they receive a service. They take their place one by one in queues where the wait, then they are served and finally they leave.

Problem analysis, modeling, scenarios, use cases

Problem analysis

Queues are used when things have to be processed in First In First Out order. Clients are entering the queues according to their arrival time. This means that they have to be placed in queues at the moment they arrive at the queues. A client enters one queue, waits until all the clients preceding him are served, receives the service and then leaves. In this application, the clients are characterized by their ID, arrival time and service time. The clients could be placed in the queues according to one of the following 2 criterion: they go where the waiting time is the shortest or where there are the fewest clients already waiting.

Modelling

In the interface there are 3 text fields where the user need to fill in the number of clients who have to enter the queues, the number of queues, and the simulation interval, 2 text fields for the minimum and the maximum arrival time – these 2 values represent the bounds of the interval where the random generated values are placed – and 2 text fields for the minimum and maximum service time – the bounds of the interval where the generated service time of each client has to be. The arrival time represents the moment when the client enters one queue and the service time is the time interval needed to serve the client. There is also a combo box used to select the preferred selection policy: shortest waiting time or smallest number of clients in queue.

After the user fills in all the text fields and chooses the selection policy, he has to press the “Start simulation” button. After that, a text area where the results of the simulation are progressively displayed will become visible.

Use cases

A use case is a definition of a specific objective that the system needs to accomplish. The two types of use-cases can be described at an abstract level (business use-case) or at an implementation-specific level (system use-case).

The **business use-cases** in my project are “fill in the input data”, “view simulation results”, “view log events”, “view simulation statistics” referring to the high-level business process being described, and the user, who is the primary actor and represents the external entity that takes part in the process. The business needs to perform a sequence of actions, which are defined by the business use-case, in order to generate a meaningful result to the external entity.

Diagram

Description automatically generated

Scenarios

The main success scenario would be the following one: the user inserts the appropriate data, starts the simulation by pressing the “Start simulation button” and then he can see the results on the screen. According to the desired simulation time, the contents of the text area will update at each unit of time until the whole interval is covered or until there are no more clients to be served.

Another possible scenario would be in case the user fills in invalid data. This can happen when the text fields are not completed, the selection policy not chosen, the time intervals not correctly specified (if the minimum time is larger than the maximum one) or if the introduced numbers are not positive integers.

1. Design (design decisions, UML diagrams, data structures, class design, interfaces, relationships, packages, algorithms, user interfaces)

After the data is introduced, the number of clients, the minimum and maximum arrival time and the minimum and maximum service time are used in order to create the array of clients. The arrivalTime and serviceTime are values generated randomly in the interval bounded by the specified minimum and maximum. The clients are sorted in the ascending order of their arrivalTime. After that, the queues are created. Each queue is considered as having an associated processor. The threads will be launched to process in parallel the clients. Another thread that corresponds to the manager of the simulation will be launched. It has the objective to hold the global time of the simulation, to manage the queues by adding clients to them and to do the statistics. After each of the threads corresponding to the queues terminate the execution, it means that all the clients have been processed and the simulation stops.

As data structures, I used arrays of type ArrayList or BlockingQueue. Arrays are collections containing elements of similar type and has contiguous memory location. The disadvantage of this data structure is that it can store only a fixed set of elements and it doesn’t grow its size at the runtime. To solve this problem, I preferred to use ArrayList, which is a resizable-array implementation of the List interface. In this assignment I had to use BlockingQueue type of array in order to represent the queues. A BlockingQueue is a queue that additionally supports operations that wait for the queue to become non-empty when retrieving an element and wait for space to become available in the queue when storing an element. One of the important advantages is that it is thread-safe because all the methods used for queuing achieve their effects atomically using some forms of concurrency control.

As data types, I used both **primitive** data types, like **int**, **double** or **boolean**, and **non-primitive** data types, such as **String**, **Integer**, **ArrayLists** and **Classes**.

I chose to use the interactive system pattern **Model View Controller.**

* **Model** is the central component of the pattern and it is the application’s dynamic data structure, independent of the user interface and receives user input from the controller.
* **View** is used for the UI logic of the application.
* **Controller** receives the input, inserted by the user and converts it to commands for the **Model** and **View**. It is like an interface between the other two components.

**Controller** updates the **Model** based on the events happening as a consequence of the inserted input in the **View** part.

Diagram

Description automatically generated

My project contains classes divided into 4 packages:

* application – contains the “Main” class;
* model – contains any business logic-related data. This package contains the following classes: “Client”, “RandomClientsGenerator”, “Scheduler”, “Server”, “SimulationManager”, “Statistics”, the enum “SelectionPolicy” and the package “strategy” containing the classes “ConcreteStrategyQueue”, “ConcreteStrategyTime” and the interface “Strategy” which is implemented by the previous two classes;
* view - contains the class “Interface” and the FXML file used to create the interface;
* controller – contains the class “QueuesSimulatorController”,

and a package containing the text files where the generated results of the simulation are displayed:

* logs – contains the text files “log.”, “test1”, “test2”, “test3”

application:

* Main - Here, I create the object controller of the class QueuesSimulatorController and I call the start() method from the QueuesSimulatorController.

model: - which contains:

* strategy
  + ConcreteStartegyQueue
  + ConcreteStrategyTime
  + Strategy
* Client
* RandomClientsGenerator
* Scheduler
* SelectionPolicy
* Server
* SimulationManager
* Statistics

view:

* Interface
* the FXML file sample.fxml

controller:

* QueuesSimulatorController

Diagram

Description automatically generated with medium confidence

1. Implementation

* Model
* Client

Constructors: *public* Client(Integer ID, Integer arrivalTime, Integer serviceTime)

Getters and setters:

*public* Integer getID()

public Integer getArrivalTime()

*public* Integer getServiceTime()

*public void* setServiceTime(Integer serviceTime)

*public* String getStatus()

*public void* setStatus(String status)

Other methods:

*public int* compareTo(Client compareClient)

* The clients need to be compared by their arrival time, so the class Client need to implements the class Comparable<Client> and the method compareTo overridden.
* RandomClientsGenerator

*public void* createClientsArray(ArrayList<Client> clients, Integer noOfClients, Integer minArrivalTime, Integer maxArrivalTime, Integer minServiceTime, Integer maxServiceTime)

* This method adds to the array each client generated in the generateRandomTuple method from below

*private* Client generateRandomTuple(*int* clientNo, *int* minArrivalTime, *int* maxArrivalTime, *int* minServiceTime, *int* maxServiceTime)

* This method creates the tuple (ID, arrivalTime, serviceTime) using the values generated randomly between the minimum and maximum specified values using the 2 methods below; the ID is considered the index in the loop that iterates through the integer values between 0 and the number of clients to be generated; this method returns a client being represented by the 3 parameters from the tuple;

*private int* generateRandomArrivalTime(*int* minArrivalTime, *int* maxArrivalTime)

*private int* generateRandomServiceTime(*int* minServiceTime, *int* maxServiceTime)

* Server

Constructors: *public* Server(*int* queueIndex)

Getters and setters:

*public BlockingQueue*<Client> getClients()

*public* AtomicInteger getWaitingPeriod()

*public int* getQueueIndex()

*public* AtomicInteger getNoOfClients()

*public* Thread getQueueThread()

*public* String getStatus()

Other methods:

*public void* addClientToQueue(Client client, Statistics statistics)

*public void* removeClientFromQueue(Client client)

*public void* serveClient(Client client)

*public void* stopServer()

*public void* run()

* An object of this class has an index, a queue of clients, a waiting period, the number of clients from that queue, a thread that processes in parallel with the ones corresponding to the other queues, status that can be “Open” or “Closed”, and a boolean that is false while the thread is not still running
* The waiting period is obtained by adding the service period of each client when added to the queue and it needs to be known in order to be compared to the waiting time of the other queues when the selection policy is to choose the shortest waiting time when a client is to enter a queue;
* The number of clients in queue needs to be known in case the selection policy is that the clients have to go to the queue with the shortest length
* The status of the server is set to be “Closed” when there are no clients in the queue; otherwise, it is “Open”
* The class “Server” implements the interface Runnable and the method run() which will include the code that will be executed in the thread; being synchronized with the thread from the SimulationManager, this run() method has to run at each unit of time and if the queue is not empty, then the first client in queue has to be served; when the client is served:
  + the service time of that client has to be decremented at each unit of time, also the waiting period of the queue is decremented as the client is being processed;
  + when the service time of the client is 0, he is removed from the queue
* Scheduler

Constructors: *public* Scheduler(*int* noOfServers, SelectionPolicy policy)

Getters and setters: *public* ArrayList<Server> getServers()

Other methods:

*public void* changeStrategy(SelectionPolicy policy)

*public void* stopServers()

*public int* computeNoOfClientsCurrentlyInQueues()

*public boolean* checkIfAllTheServersAreEmpty()

*public void* dispatchClient(Client client, Statistics statistics)

* This class manages the array of servers; in the constructor the servers are instantiated and the strategy is established according to the policy, such that the method addClient declared in the Strategy interface could be called from the class implementing the Strategy that corresponds to the selected policy;
* Strategy
* This is an interface where the method addClient is declared; an object of this type is created in the class Scheduler, where it is decided which one of the classes that implements this interface is used according to the selection policy preferred by the user
* ConcreteStartegyQueue
* This is one of the classes that implemnets the Strategy interface; it implements the addClient method in such a way that the client will be added to that queue where the number of clients waiting is the smallest
* ConcreteStrategyTime
* This is one of the classes that implemnets the Strategy interface; it implements the addClient method in such a way that the client will be added to that queue where the waiting time is the shortest
* SelectionPolicy
* This is an enum and includes the constants SHORTEST\_QUEUE and SHORTEST\_TIME
* Statistics

Constructors: *public* Statistics()

Getters and setters: *public void* setPeakHour(*int* peakHour)

Other methods:

*public void* addWaitingTime(Server server)

*public void* addClient()

*public void* computeAverageWaitingTime()

*public void* addServiceTime(Client client)

*public void* computeAverageServiceTime()

*public* String changeToDecimalFormat(*double* numberToBeDisplayed)

* This class has methods used to make some computations needed to obtain the statistics results;
* The average waiting time represents the average of the waiting time that each client who entered the queues had; in other words, it represents the average of all the values of the waiting time computed at one moment in all the queues when a client entered the queue
* The average service time represents the average of the serviceTime parameters of all the clients
* The peak hour represents the simulation time when the sum of the clients from all the queues is the greatest
* SimulationManager

Constructor: *public* SimulationManager(*int* noOfClients, *int* noOfServers, *int* timeLimit, *int* minArrivalTime, *int* maxArrivalTime, *int* minServiceTime, *int* maxServiceTime, String selectionPolicyString, TextArea textArea)

Methods:

*public void* establishSelectionPolicy(String selectionPolicy)

*public void* createClientsArray()

*private void* sortClientsArray()

* Methods that write to text files
* Methods that write to text area in the interface

*public void* run()

* SimulationManager implements Runnable, so it implements the method run() that includes the tasks that this thread has to execute in one unit of time
* Here the simulation time is hold and each client is distributed to the queue when the arrival time of the client is equal to the simulation time; the simulation time increments until it reaches the time limit; here are done also the statistics and the writing to the text area from the interface;
* In this class are also implemented the methods used for displaying the log of events in the text files from the “logs” package
* View
* Interface
* sample.fxml
* Here, an object of type URL is created in order to realize the connection with the fxml file which contains the declaration and definition of all the elements that appear on the user interface. Inside the main stage is inserted a scene which represents the content displayed inside a window. When the JavaFX application starts up, it creates a root Stage object which is passed to the start method of the root class of the application.
* As a layout I used an AnchorPane and as UI controls I used labels, text fields, a combo box, a button and a text area.

Graphical user interface

Description automatically generated

* Controller
* QueuesSimulatorController
* This class includes the some getters for the textFields and comboBox, the method used to set an actionEvent on the button and one method to validate the input
* In the startSimulation method, which sets an event on the button, it is created an object of the class SimulationManager and a thread corresponding to this object is started.

1. Results

The UI presents the results of the simulation progressively. At each moment of time, the results obtained are displayed on the screen and when the simulation is done there are displayed also the simulation statistics results. The logs can be seen in the text files where they are generated during the simulation.

In the following, one can see the results obtained after running the program for the input data specified for each test:

Test 1

N = 4

Q = 2

𝑡𝑠𝑖𝑚𝑢𝑙𝑎𝑡𝑖𝑜𝑛 𝑀𝐴𝑋 = 60 seconds

[𝑡𝑎𝑟𝑟𝑖𝑣𝑎𝑙 𝑀𝐼𝑁, 𝑡𝑎𝑟𝑟𝑖𝑣𝑎𝑙 𝑀𝐴𝑋 ] = [2, 30]

[𝑡𝑠𝑒𝑟𝑣𝑖𝑐𝑒 𝑀𝐼𝑁, 𝑡𝑠𝑒𝑟𝑣𝑖𝑐𝑒 𝑀𝐴𝑋 ] = [2, 4]

Results obtained after the simulation has finished:

N: 4  
Q: 2  
simulation time: 60  
selection policy: shortest waiting time  
Time 0  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 1  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 2  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 3  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 4  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 5  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 6  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 7  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 8  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 9  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 10  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 11  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 12  
Waiting clients:   
(3, 13, 4) (2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 13  
Waiting clients:   
(2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: (3, 13, 4)   
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 14  
Waiting clients:   
(2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: (3, 13, 3)   
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 15  
Waiting clients:   
(2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: (3, 13, 2)   
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 16  
Waiting clients:   
(2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: (3, 13, 1)   
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 17  
Waiting clients:   
(2, 18, 2) (1, 19, 2) (4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 18  
Waiting clients:   
(1, 19, 2) (4, 28, 3)   
Queue 1: (2, 18, 2)   
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 19  
Waiting clients:   
(4, 28, 3)   
Queue 1: (2, 18, 1)   
Queue 2: (1, 19, 2)   
-------------------------------------------------------------------------------------------------------------------  
Time 20  
Waiting clients:   
(4, 28, 3)   
Queue 1: Closed  
Queue 2: (1, 19, 1)   
-------------------------------------------------------------------------------------------------------------------  
Time 21  
Waiting clients:   
(4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 22  
Waiting clients:   
(4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 23  
Waiting clients:   
(4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 24  
Waiting clients:   
(4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 25  
Waiting clients:   
(4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 26  
Waiting clients:   
(4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 27  
Waiting clients:   
(4, 28, 3)   
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 28  
Queue 1: (4, 28, 3)   
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 29  
Queue 1: (4, 28, 2)   
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 30  
Queue 1: (4, 28, 1)   
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 31  
Queue 1: Closed  
Queue 2: Closed  
-------------------------------------------------------------------------------------------------------------------  
Statistics results:  
average waiting time: 2.75  
average service time: 2.75  
peak hour: 19

Test 2

N = 50

Q = 5

𝑡𝑠𝑖𝑚𝑢𝑙𝑎𝑡𝑖𝑜𝑛 𝑀𝐴𝑋 = 60 seconds

[𝑡𝑎𝑟𝑟𝑖𝑣𝑎𝑙 𝑀𝐼𝑁 , 𝑡𝑎𝑟𝑟𝑖𝑣𝑎𝑙 𝑀𝐴𝑋 ] = [2, 40]

[𝑡𝑠𝑒𝑟𝑣𝑖𝑐𝑒 𝑀𝐼𝑁 , 𝑡𝑠𝑒𝑟𝑣𝑖𝑐𝑒 𝑀𝐴𝑋 ] = [1, 7]

Results obtained after the simulation has finished:

N: 50  
Q: 5  
simulation time: 60  
selection policy: shortest waiting time  
Time 0  
Waiting clients:   
(30, 2, 2) (34, 2, 3) (7, 3, 7) (39, 3, 4) (43, 3, 6) (15, 4, 4) (33, 4, 4) (44, 4, 6) (6, 6, 4) (20, 6, 1)   
(29, 7, 4) (10, 8, 5) (11, 12, 5) (48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5)   
(21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7)   
(42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4)   
(12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: Closed  
Queue 2: Closed  
Queue 3: Closed  
Queue 4: Closed  
Queue 5: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 1  
Waiting clients:   
(30, 2, 2) (34, 2, 3) (7, 3, 7) (39, 3, 4) (43, 3, 6) (15, 4, 4) (33, 4, 4) (44, 4, 6) (6, 6, 4) (20, 6, 1)   
(29, 7, 4) (10, 8, 5) (11, 12, 5) (48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5)   
(21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7)   
(42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4)   
(12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: Closed  
Queue 2: Closed  
Queue 3: Closed  
Queue 4: Closed  
Queue 5: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 2  
Waiting clients:   
(7, 3, 7) (39, 3, 4) (43, 3, 6) (15, 4, 4) (33, 4, 4) (44, 4, 6) (6, 6, 4) (20, 6, 1) (29, 7, 4) (10, 8, 5)   
(11, 12, 5) (48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7)   
(2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6)   
(26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5)   
(47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (30, 2, 2)   
Queue 2: (34, 2, 3)   
Queue 3: Closed  
Queue 4: Closed  
Queue 5: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 3  
Waiting clients:   
(15, 4, 4) (33, 4, 4) (44, 4, 6) (6, 6, 4) (20, 6, 1) (29, 7, 4) (10, 8, 5) (11, 12, 5) (48, 14, 6) (25, 15, 3)   
(8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5)   
(5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1)   
(1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2)   
(13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (30, 2, 1)   
Queue 2: (34, 2, 2)   
Queue 3: (7, 3, 7)   
Queue 4: (39, 3, 4)   
Queue 5: (43, 3, 6)   
-------------------------------------------------------------------------------------------------------------------  
Time 4  
Waiting clients:   
(6, 6, 4) (20, 6, 1) (29, 7, 4) (10, 8, 5) (11, 12, 5) (48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6)   
(37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4)   
(36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2)   
(27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4)   
(19, 40, 2) (24, 40, 6)   
Queue 1: (15, 4, 4)   
Queue 2: (34, 2, 1) (33, 4, 4)   
Queue 3: (7, 3, 6)   
Queue 4: (39, 3, 3) (44, 4, 6)   
Queue 5: (43, 3, 5)   
-------------------------------------------------------------------------------------------------------------------  
Time 5  
Waiting clients:   
(6, 6, 4) (20, 6, 1) (29, 7, 4) (10, 8, 5) (11, 12, 5) (48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6)   
(37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4)   
(36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2)   
(27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4)   
(19, 40, 2) (24, 40, 6)   
Queue 1: (15, 4, 3)   
Queue 2: (33, 4, 4)   
Queue 3: (7, 3, 5)   
Queue 4: (39, 3, 2) (44, 4, 6)   
Queue 5: (43, 3, 4)   
-------------------------------------------------------------------------------------------------------------------  
Time 6  
Waiting clients:   
(29, 7, 4) (10, 8, 5) (11, 12, 5) (48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5)   
(21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7)   
(42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4)   
(12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (15, 4, 2) (6, 6, 4)   
Queue 2: (33, 4, 3) (20, 6, 1)   
Queue 3: (7, 3, 4)   
Queue 4: (39, 3, 1) (44, 4, 6)   
Queue 5: (43, 3, 3)   
-------------------------------------------------------------------------------------------------------------------  
Time 7  
Waiting clients:   
(10, 8, 5) (11, 12, 5) (48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5)   
(40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1)   
(50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2)   
(38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (15, 4, 1) (6, 6, 4)   
Queue 2: (33, 4, 2) (20, 6, 1)   
Queue 3: (7, 3, 3)   
Queue 4: (44, 4, 6)   
Queue 5: (43, 3, 2) (29, 7, 4)   
-------------------------------------------------------------------------------------------------------------------  
Time 8  
Waiting clients:   
(11, 12, 5) (48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7)   
(2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6)   
(26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5)   
(47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (6, 6, 4)   
Queue 2: (33, 4, 1) (20, 6, 1) (10, 8, 5)   
Queue 3: (7, 3, 2)   
Queue 4: (44, 4, 5)   
Queue 5: (43, 3, 1) (29, 7, 4)   
-------------------------------------------------------------------------------------------------------------------  
Time 9  
Waiting clients:   
(11, 12, 5) (48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7)   
(2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6)   
(26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5)   
(47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (6, 6, 3)   
Queue 2: (20, 6, 1) (10, 8, 5)   
Queue 3: (7, 3, 1)   
Queue 4: (44, 4, 4)   
Queue 5: (29, 7, 4)   
-------------------------------------------------------------------------------------------------------------------  
Time 10  
Waiting clients:   
(11, 12, 5) (48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7)   
(2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6)   
(26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5)   
(47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (6, 6, 2)   
Queue 2: (10, 8, 5)   
Queue 3: Closed  
Queue 4: (44, 4, 3)   
Queue 5: (29, 7, 3)   
-------------------------------------------------------------------------------------------------------------------  
Time 11  
Waiting clients:   
(11, 12, 5) (48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7)   
(2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6)   
(26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5)   
(47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (6, 6, 1)   
Queue 2: (10, 8, 4)   
Queue 3: Closed  
Queue 4: (44, 4, 2)   
Queue 5: (29, 7, 2)   
-------------------------------------------------------------------------------------------------------------------  
Time 12  
Waiting clients:   
(48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7) (2, 23, 1)   
(3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3)   
(32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7)   
(4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (11, 12, 5)   
Queue 2: (10, 8, 3)   
Queue 3: Closed  
Queue 4: (44, 4, 1)   
Queue 5: (29, 7, 1)   
-------------------------------------------------------------------------------------------------------------------  
Time 13  
Waiting clients:   
(48, 14, 6) (25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7) (2, 23, 1)   
(3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3)   
(32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7)   
(4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (11, 12, 4)   
Queue 2: (10, 8, 2)   
Queue 3: Closed  
Queue 4: Closed  
Queue 5: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 14  
Waiting clients:   
(25, 15, 3) (8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7)   
(31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1)   
(46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4)   
(17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (11, 12, 3)   
Queue 2: (10, 8, 1)   
Queue 3: (48, 14, 6)   
Queue 4: Closed  
Queue 5: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 15  
Waiting clients:   
(8, 16, 4) (16, 16, 2) (9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5)   
(5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1)   
(1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2)   
(13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (11, 12, 2)   
Queue 2: (25, 15, 3)   
Queue 3: (48, 14, 5)   
Queue 4: Closed  
Queue 5: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 16  
Waiting clients:   
(9, 17, 6) (37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7)   
(35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4)   
(23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4)   
(14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (11, 12, 1)   
Queue 2: (25, 15, 2)   
Queue 3: (48, 14, 4)   
Queue 4: (8, 16, 4)   
Queue 5: (16, 16, 2)   
-------------------------------------------------------------------------------------------------------------------  
Time 17  
Waiting clients:   
(37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4)   
(36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2)   
(27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4)   
(19, 40, 2) (24, 40, 6)   
Queue 1: (9, 17, 6)   
Queue 2: (25, 15, 1)   
Queue 3: (48, 14, 3)   
Queue 4: (8, 16, 3)   
Queue 5: (16, 16, 1)   
-------------------------------------------------------------------------------------------------------------------  
Time 18  
Waiting clients:   
(37, 19, 5) (49, 19, 5) (21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4)   
(36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2)   
(27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4)   
(19, 40, 2) (24, 40, 6)   
Queue 1: (9, 17, 5)   
Queue 2: Closed  
Queue 3: (48, 14, 2)   
Queue 4: (8, 16, 2)   
Queue 5: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 19  
Waiting clients:   
(21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7)   
(42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4)   
(12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (9, 17, 4)   
Queue 2: (37, 19, 5)   
Queue 3: (48, 14, 1)   
Queue 4: (8, 16, 1)   
Queue 5: (49, 19, 5)   
-------------------------------------------------------------------------------------------------------------------  
Time 20  
Waiting clients:   
(21, 21, 5) (40, 21, 7) (2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7)   
(42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4)   
(12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (9, 17, 3)   
Queue 2: (37, 19, 4)   
Queue 3: Closed  
Queue 4: Closed  
Queue 5: (49, 19, 4)   
-------------------------------------------------------------------------------------------------------------------  
Time 21  
Waiting clients:   
(2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6)   
(26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5)   
(47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (9, 17, 2)   
Queue 2: (37, 19, 3)   
Queue 3: (21, 21, 5)   
Queue 4: (40, 21, 7)   
Queue 5: (49, 19, 3)   
-------------------------------------------------------------------------------------------------------------------  
Time 22  
Waiting clients:   
(2, 23, 1) (3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6)   
(26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5)   
(47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (9, 17, 1)   
Queue 2: (37, 19, 2)   
Queue 3: (21, 21, 4)   
Queue 4: (40, 21, 6)   
Queue 5: (49, 19, 2)   
-------------------------------------------------------------------------------------------------------------------  
Time 23  
Waiting clients:   
(3, 24, 7) (31, 24, 5) (5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3)   
(32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7)   
(4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (2, 23, 1)   
Queue 2: (37, 19, 1)   
Queue 3: (21, 21, 3)   
Queue 4: (40, 21, 5)   
Queue 5: (49, 19, 1)   
-------------------------------------------------------------------------------------------------------------------  
Time 24  
Waiting clients:   
(5, 25, 7) (22, 25, 7) (35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1)   
(1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2)   
(13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (3, 24, 7)   
Queue 2: (31, 24, 5)   
Queue 3: (21, 21, 2)   
Queue 4: (40, 21, 4)   
Queue 5: Closed  
-------------------------------------------------------------------------------------------------------------------  
Time 25  
Waiting clients:   
(35, 26, 4) (36, 26, 2) (28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4)   
(23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4)   
(14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (3, 24, 6)   
Queue 2: (31, 24, 4)   
Queue 3: (21, 21, 1) (22, 25, 7)   
Queue 4: (40, 21, 3)   
Queue 5: (5, 25, 7)   
-------------------------------------------------------------------------------------------------------------------  
Time 26  
Waiting clients:   
(28, 27, 7) (42, 27, 1) (50, 27, 6) (26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5)   
(45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2)   
(24, 40, 6)   
Queue 1: (3, 24, 5)   
Queue 2: (31, 24, 3) (36, 26, 2)   
Queue 3: (22, 25, 7)   
Queue 4: (40, 21, 2) (35, 26, 4)   
Queue 5: (5, 25, 6)   
-------------------------------------------------------------------------------------------------------------------  
Time 27  
Waiting clients:   
(26, 28, 3) (32, 28, 1) (46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5)   
(47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (3, 24, 4) (28, 27, 7)   
Queue 2: (31, 24, 2) (36, 26, 2) (42, 27, 1) (50, 27, 6)   
Queue 3: (22, 25, 6)   
Queue 4: (40, 21, 1) (35, 26, 4)   
Queue 5: (5, 25, 5)   
-------------------------------------------------------------------------------------------------------------------  
Time 28  
Waiting clients:   
(46, 29, 1) (1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4)   
(17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (3, 24, 3) (28, 27, 7)   
Queue 2: (31, 24, 1) (36, 26, 2) (42, 27, 1) (50, 27, 6)   
Queue 3: (22, 25, 5)   
Queue 4: (35, 26, 4) (26, 28, 3)   
Queue 5: (5, 25, 4) (32, 28, 1)   
-------------------------------------------------------------------------------------------------------------------  
Time 29  
Waiting clients:   
(1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2)   
(13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (3, 24, 2) (28, 27, 7)   
Queue 2: (36, 26, 2) (42, 27, 1) (50, 27, 6)   
Queue 3: (22, 25, 4) (46, 29, 1)   
Queue 4: (35, 26, 3) (26, 28, 3)   
Queue 5: (5, 25, 3) (32, 28, 1)   
-------------------------------------------------------------------------------------------------------------------  
Time 30  
Waiting clients:   
(1, 31, 7) (18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2)   
(13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (3, 24, 1) (28, 27, 7)   
Queue 2: (36, 26, 1) (42, 27, 1) (50, 27, 6)   
Queue 3: (22, 25, 3) (46, 29, 1)   
Queue 4: (35, 26, 2) (26, 28, 3)   
Queue 5: (5, 25, 2) (32, 28, 1)   
-------------------------------------------------------------------------------------------------------------------  
Time 31  
Waiting clients:   
(18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7)   
(41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (28, 27, 7)   
Queue 2: (42, 27, 1) (50, 27, 6)   
Queue 3: (22, 25, 2) (46, 29, 1)   
Queue 4: (35, 26, 1) (26, 28, 3)   
Queue 5: (5, 25, 1) (32, 28, 1) (1, 31, 7)   
-------------------------------------------------------------------------------------------------------------------  
Time 32  
Waiting clients:   
(18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7)   
(41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (28, 27, 6)   
Queue 2: (50, 27, 6)   
Queue 3: (22, 25, 1) (46, 29, 1)   
Queue 4: (26, 28, 3)   
Queue 5: (32, 28, 1) (1, 31, 7)   
-------------------------------------------------------------------------------------------------------------------  
Time 33  
Waiting clients:   
(18, 34, 4) (23, 34, 2) (27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7)   
(41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (28, 27, 5)   
Queue 2: (50, 27, 5)   
Queue 3: (46, 29, 1)   
Queue 4: (26, 28, 2)   
Queue 5: (1, 31, 7)   
-------------------------------------------------------------------------------------------------------------------  
Time 34  
Waiting clients:   
(27, 35, 5) (45, 35, 4) (12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4)   
(19, 40, 2) (24, 40, 6)   
Queue 1: (28, 27, 4)   
Queue 2: (50, 27, 4)   
Queue 3: (18, 34, 4)   
Queue 4: (26, 28, 1) (23, 34, 2)   
Queue 5: (1, 31, 6)   
-------------------------------------------------------------------------------------------------------------------  
Time 35  
Waiting clients:   
(12, 36, 2) (38, 36, 5) (47, 36, 7) (4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (28, 27, 3) (45, 35, 4)   
Queue 2: (50, 27, 3)   
Queue 3: (18, 34, 3)   
Queue 4: (23, 34, 2) (27, 35, 5)   
Queue 5: (1, 31, 5)   
-------------------------------------------------------------------------------------------------------------------  
Time 36  
Waiting clients:   
(4, 37, 4) (17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (28, 27, 2) (45, 35, 4)   
Queue 2: (50, 27, 2) (12, 36, 2) (47, 36, 7)   
Queue 3: (18, 34, 2) (38, 36, 5)   
Queue 4: (23, 34, 1) (27, 35, 5)   
Queue 5: (1, 31, 4)   
-------------------------------------------------------------------------------------------------------------------  
Time 37  
Waiting clients:   
(17, 38, 2) (13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (28, 27, 1) (45, 35, 4)   
Queue 2: (50, 27, 1) (12, 36, 2) (47, 36, 7)   
Queue 3: (18, 34, 1) (38, 36, 5)   
Queue 4: (27, 35, 5)   
Queue 5: (1, 31, 3) (4, 37, 4)   
-------------------------------------------------------------------------------------------------------------------  
Time 38  
Waiting clients:   
(13, 39, 7) (41, 39, 4) (14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (45, 35, 4) (17, 38, 2)   
Queue 2: (12, 36, 2) (47, 36, 7)   
Queue 3: (38, 36, 5)   
Queue 4: (27, 35, 4)   
Queue 5: (1, 31, 2) (4, 37, 4)   
-------------------------------------------------------------------------------------------------------------------  
Time 39  
Waiting clients:   
(14, 40, 4) (19, 40, 2) (24, 40, 6)   
Queue 1: (45, 35, 3) (17, 38, 2)   
Queue 2: (12, 36, 1) (47, 36, 7)   
Queue 3: (38, 36, 4) (41, 39, 4)   
Queue 4: (27, 35, 3) (13, 39, 7)   
Queue 5: (1, 31, 1) (4, 37, 4)   
-------------------------------------------------------------------------------------------------------------------  
Time 40  
Queue 1: (45, 35, 2) (17, 38, 2) (14, 40, 4)   
Queue 2: (47, 36, 7)   
Queue 3: (38, 36, 3) (41, 39, 4)   
Queue 4: (27, 35, 2) (13, 39, 7)   
Queue 5: (4, 37, 4) (19, 40, 2) (24, 40, 6)   
-------------------------------------------------------------------------------------------------------------------  
Time 41  
Queue 1: (45, 35, 1) (17, 38, 2) (14, 40, 4)   
Queue 2: (47, 36, 6)   
Queue 3: (38, 36, 2) (41, 39, 4)   
Queue 4: (27, 35, 1) (13, 39, 7)   
Queue 5: (4, 37, 3) (19, 40, 2) (24, 40, 6)   
-------------------------------------------------------------------------------------------------------------------  
Time 42  
Queue 1: (17, 38, 2) (14, 40, 4)   
Queue 2: (47, 36, 5)   
Queue 3: (38, 36, 1) (41, 39, 4)   
Queue 4: (13, 39, 7)   
Queue 5: (4, 37, 2) (19, 40, 2) (24, 40, 6)   
-------------------------------------------------------------------------------------------------------------------  
Time 43  
Queue 1: (17, 38, 1) (14, 40, 4)   
Queue 2: (47, 36, 4)   
Queue 3: (41, 39, 4)   
Queue 4: (13, 39, 6)   
Queue 5: (4, 37, 1) (19, 40, 2) (24, 40, 6)   
-------------------------------------------------------------------------------------------------------------------  
Time 44  
Queue 1: (14, 40, 4)   
Queue 2: (47, 36, 3)   
Queue 3: (41, 39, 3)   
Queue 4: (13, 39, 5)   
Queue 5: (19, 40, 2) (24, 40, 6)   
-------------------------------------------------------------------------------------------------------------------  
Time 45  
Queue 1: (14, 40, 3)   
Queue 2: (47, 36, 2)   
Queue 3: (41, 39, 2)   
Queue 4: (13, 39, 4)   
Queue 5: (19, 40, 1) (24, 40, 6)   
-------------------------------------------------------------------------------------------------------------------  
Time 46  
Queue 1: (14, 40, 2)   
Queue 2: (47, 36, 1)   
Queue 3: (41, 39, 1)   
Queue 4: (13, 39, 3)   
Queue 5: (24, 40, 6)   
-------------------------------------------------------------------------------------------------------------------  
Time 47  
Queue 1: (14, 40, 1)   
Queue 2: Closed  
Queue 3: Closed  
Queue 4: (13, 39, 2)   
Queue 5: (24, 40, 5)   
-------------------------------------------------------------------------------------------------------------------  
Time 48  
Queue 1: Closed  
Queue 2: Closed  
Queue 3: Closed  
Queue 4: (13, 39, 1)   
Queue 5: (24, 40, 4)   
-------------------------------------------------------------------------------------------------------------------  
Time 49  
Queue 1: Closed  
Queue 2: Closed  
Queue 3: Closed  
Queue 4: Closed  
Queue 5: (24, 40, 3)   
-------------------------------------------------------------------------------------------------------------------  
Time 50  
Queue 1: Closed  
Queue 2: Closed  
Queue 3: Closed  
Queue 4: Closed  
Queue 5: (24, 40, 2)   
-------------------------------------------------------------------------------------------------------------------  
Time 51  
Queue 1: Closed  
Queue 2: Closed  
Queue 3: Closed  
Queue 4: Closed  
Queue 5: (24, 40, 1)   
-------------------------------------------------------------------------------------------------------------------  
Time 52  
Queue 1: Closed  
Queue 2: Closed  
Queue 3: Closed  
Queue 4: Closed  
Queue 5: Closed  
-------------------------------------------------------------------------------------------------------------------  
Statistics results:  
average waiting time: 6.06  
average service time: 4.3  
peak hour: 28

1. Conclusions

In conclusion, as the main purpose of this assignment was to learn to work with threads, I understood that there are many advantages in having this skill because there are times when a process has multiple tasks that perform independently of the others and they have to be managed in such a way that they do not have the possibility to affect negatively one another. Some of the advantages of threads are that they minimize the context switching time, the concurrency within a process can be provided, the communication is efficient and they allow the usage of multiprocessor architectures to a larger scale.

1. Bibliography

https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/BlockingQueue.html

<https://www.w3schools.com/java/java_enums.asp>

<https://www.techopedia.com/>

<https://www.wikipedia.org/>

<https://stackoverflow.com/threads>

https://www.tutorialspoint.com/operating\_system/os\_multi\_threading.htm