

CSE 344 System Programming Project

Pide Shop

In this project you are expected to simulate a food production and delivery system. The overall system is composed of three main components: (i) the pide house containing a special pide oven (ii) delivery personnel waiting on their motorcycles in front of the store and (iii) lots of hungry customers.

The manager of the pide house gets the orders via phone and assigns it to one of the currently available cooks in the shop. The cook takes the order and prepares it around the same time as it takes your computer to calculate the pseudo-inverse of a 30 by 40 matrix having complex elements. Then collects the necessary apparatus (i.e. fırıncı küreği..) to place it in the oven and cooks for around half the time as the meal's preparation. When all these are completed the prepared food is given back to the manager who then hands it over to one of the available delivery personnel that makes the delivery to the customer. The velocity of the delivery personnel is fixed, so the delivery time is dependent on the address information provided by the customer.

The shop has one manager, n cooking personnels, m delivery personnels, but only 3 oven apparatus (i.e. fırıncı küreği..). The oven on the other hand can hold up to 6 meals inside and has 2 openings to place and remove the meals from. In order to optimize the production, cooks use these openings only to place and remove the meals and do not block the entries. They go back to their tables and prepare meals for other customers and return only when the meal is ready. The manager hands the fully prepared and cooked meal to one of the available delivery personnel with the address information of the customer. Each delivery person has three places on his/her carrying bag. And when his/her bag is full he leaves the shop to make the deliveries. It is the delivery personnel's duty to calculate the amount of time to place the deliveries and return back to the shop. However at the end of the day, the manager checks the amount of deliveries of each personnel and promotes the most efficient one. The Customers are informed when their orders are placed, prepared, cooked and handed out for delivery.

Assume that the town in which this fake shop is placed, is a p km by q km rectangle and the pide shop is in the middle of it. Each customer when placing an order is expected to give their location relative to the major office south entrance (0, 0) of the town.

In case of an order cancellation, regardless of the place or stage of the order, (in preparation, inside the oven or even in delivery) the shop discards the product (stops preparation, removes it from the oven, discards its

delivery). Removes the cooks waiting for the oven queue, removes the prepared meals from the delivery persons bags and discards them.

Your implementation would require a multithread internet server with at least two thread pools (one for cooks and one for the delivery personnel), the server should be able to handle signals properly, create a log file to keep the timely shops activities. An example call for the server could be of the form

```
> PideShop [portnumber] [CookthreadPoolSize] [DeliveryPoolSize] [k]
```

where `portnumber` is the port for connection, and the rest of the entries are the corresponding thread pool size of the threads for the cooks and delivery personnel and `k` being the speed in m/min of the delivery personnel. The server is expected to indicate when a connection is made, specifying the number of clients. Shop server also keeps an log file in order to keep track of the state and delivery time of the orders.

Use a client generator of the form

```
> HungryVeryMuch [portnumber] [numberOfClients] [p] [q]
```

where `portnumber` is the connection port of the server, `numberOfClients` generated, `p` and `q` being the size of the town in kms. Note that each client is generated with the corresponding position inside the town (use some sort of random number generator) and the client should return with a proper message when server is down and server should prompt a message when a client connection is accepted (with the address of connection) to the screen. `^C` and `^D` on client side corresponds for order cancellation. `^C` and `^D` on server side means the shop has been burned down, don't wait for the orders.

And example scenario is given below

Server Side	Client Side
> PideShop 192.168.10.45 4 6 1	> HungryVeryMuch 192.168.10.45 50 10 20
> > PideShop active waiting for connection ...	> > PID 2345..
> > 50 new customers.. Serving	> > ...
> > done serving client @ XXX PID 2345	> > All customers served
> > Thanks Cook 2 and Moto 5	> > log file written ..
> > active waiting for connections	>
	> HungryVeryMuch 192.168.10.45 10 10 20
	> > PID 2453
	> > ^C signal .. cancelling orders.. editing log..
> > 10 new customers.. Serving	>
> > order cancelled @YYY PID 2453	
> > ^C.. Upps quitting.. writing log file	
>	

Test your code with multiple (50, 100, 200) clients, reconnect to see if the server updates the client information properly. Check what happens when a new order is added, edited or removed on the client side when the client server connection is still active. Write a report, explaining your design, algorithm with at least 5 different test cases.

Submit only 1 compressed file indicating your number and name, (that is the student Zühtü Dülek with student id 1201040001 is expected to submit his project in a file 1201040001_ZÜHTÜ_DÜLEK.zip) your project should include a report , your code and a makefile.

Best Luck