

The aim of this exercise is to become familiar with IPC-Inter-Process Communication and Internet sockets.

For this purpose, you are asked to use the system calls `socket()`, `bind()`, `connect()`, `fork()`, etc. and implement an "online store" that accepts orders from customers via "internet".

Two files will be delivered: Client.c, Server.c and one bash file. You can also create a bash file to run 5 clients together.

Description

The eshop will have a list of products for which there is a limited number of pieces available.

The "customers" will request the purchase of the products in the online store, and if the products are available, the store will inform the customer about the success of the order, the total charge and will bind the products. Otherwise, it will inform the customer that the requested product is not in stock.

The online store will have a specific order processing time (0.5 seconds). At this time, it will not accept any more orders until the service is over. Each customer will submit a series of orders with a difference of 1 second between orders.

Once all orders have been completed, the online store will issue a summary report which will describe the following information for each product:

- 1) Product description
- 2) Number of purchase requests
- 3) Number of pieces sold
- 4) List of users who were not served (did not find the product available)

A summary message will be printed at the end of the report containing:

- 1) Total number of orders
- 2) Total number of successful orders
- 3) Total number of failed orders
- 4) Total money earned

Implementation Instructions

You will implement the catalog through a table (named catalog) which will exist in the parent process. The table will have a size of 20 different products. Each product will be a structure that you will implement and will contain the following fields: description, price, item_count. For each product set the item_count value to 2 (there are 2 products available with this description).

The initial process (parent) will perform the work of the eshop. The parent process must initialize the catalog table and initialize a “named” socket to receive messages from customers via sockets. Through sockets it will be able to accept orders from customers (for an order, customers will write with write() in the socket, and the server will read with read() from the socket).

To be informed about the progress of the order, the socket you have already created will be used (this is a two-way communication), in which the server will write and from which the customer will read the message that will describe the result of the order (whether or not he found the requested product).

The processing time of the order will be 1 second. You can implement the processing time using the sleep() system call from the parent process.

For the server, you can run it through bash and run in the background (with the character “&” at the end of the command).

For clients, your program can create (through a bash script and using the background process with the “&” character) a number of processes that will act as "clients". Customers do not need to use the fork().

Customers will communicate with the eshop via internet sockets (use of af_inet). You will create 5 different clients through a special bash script. (optional: for running the script you can create a file allclients.sh, using 5 “client” commands one under each, instead of running 5 times the ./clients & command) Each customer will send 10 orders and then terminate its execution. He will wait 1 second between orders. Waiting can be accomplished using the sleep() system call. For each order, it will execute the rand() function to select a random number between 0 and

20 (where 20 are the different products). Once he chooses the product, he will put his order in the socket and wait for the result of his order.