CSE 344 System Programming
Spring 2023-24
FINAL PROJECT REPORT

Name: Atakan Kurt
Student #: 200104004044
E-Mail: akurt2020@gtu.edu.tr

Project Description:
This project is about creating a simulation of a pide shop, which is a type of food store, where we manage food orders, cooking, and delivery. The shop has a manager who takes orders, cooks who prepare the food, and delivery people who bring the food to customers. The manager assigns tasks to the cooks, who make the food and use a special oven. After cooking, the food is handed to the delivery people, who take it to the customers' homes. The project involves writing a program with threads to handle multiple cooks and delivery people at the same time. It also involves logging all activities and handling order cancellations.

For running instructions, you can read README.md in hw folder.

hungryVeryMuch

1. **Connecting to the Server:** The program connects to a PideShop server using a provided IP address and port number.
2. **Signal Handling:** The program listens for signals (Ctrl+C) to handle order cancellations gracefully.
3. **Sending Orders:** It sends multiple orders from clients to the server, simulating a group of customers placing orders.
4. **Receiving Responses:** The program listens for messages from the server to check the status of the orders.
5. **Exiting:** When all orders are completed or the server shuts down, the program exits cleanly.

The program starts by checking if the correct number of arguments is provided. It takes the server's IP address and port, the number of clients, and the dimensions of the town (p by q).
It extracts the port number from the IP address if specified. It sets up the socket and server address structures. It connects to the server. Then it sets up signal handlers for SIGINT (Ctrl+C) and SIGTERM to cancel orders and clean up properly. It uses a loop to generate random coordinates (x, y) for each client within the town's dimensions. For each client, it sends an order to the server with the client's number, location, and process ID (PID). It continuously reads messages from the server. If a message indicates **all orders are completed or the server is shutting down**, it breaks out of the loop. It prints out any other messages received from the server. When done, it prints a message indicating that the log file is written and closes the socket.

Example running
./HungryVeryMuch 192.168.10.45 10 10 20

PideShop
Cook: Represents a cook in the shop, tracking their ID, if they are busy, and how many orders they've handled.
Order: Represents an order from a client, including client number, position, socket connection, and completion status.
DeliveryPerson: Represents a delivery person, tracking their ID, if they are busy, orders they are delivering, delivery count, speed, and total orders delivered.
OvenQueue: Manages orders waiting to be cooked or already cooked, using a queue structure.
Uses multiple mutexes and condition variables to handle synchronization between different threads (cooks, delivery persons, and oven manager).
Globals:
Arrays of cooks, delivery persons, and orders.
Counters and queues for tracking orders and oven usage.
A log file to record activities.
Threads are initilized globally to easy join them when we need to clean.

Separate threads for the oven managament, cooks, delivery persons, and client handling. Each type of thread performs specific tasks to simulate the shop's operations.

Main function similar to customers architecture
Parse command-line arguments to get IP address, pool sizes for cooks and delivery persons, and delivery speed. Open a log file to record activities. Set up signal handlers for termination. Initialize arrays of cooks and delivery persons with their respective attributes (ID, busy status, etc.). Create threads for each cook and delivery person, assigning them their specific functions (cooking or delivering). Create a socket for the server. Set socket options to allow reuse of the address. Bind the server to an available port in the specified range. Listen for incoming connections. Accepts new client conntections in a loop. For each new connection, create a separate thread to handle the client, ensuring the main server loop continues to accept new connections. Each client thread handles incoming orders, places them in the order queue, and updates the client about the order status.
Oven Manager: Manages the oven queue, ensuring orders are placed and removed from the oven efficiently. I used this extra thread to enable cooks prepare while pideler cooking.
Cooks: Prepare orders and place them in the oven queue.
Delivery Persons: Collect cooked orders and deliver them to clients, updating the order status.
When a termination signal is received, all threads are notified to stop.
Active orders are canceled, clients are notified, and the log file is updated. All threads are joined to ensure they complete their tasks before the program exits.

oven_manager:

The function runs in a continuous loop until a stop signal is received.
It locks the oven queue mutex to access the orders waiting to be cooked.
If there are no orders in the queue and no stop signal, it waits on the oven queue condition variable.
If a stop signal is received during waiting, it unlocks the mutex and breaks out of the loop.

If an order is available in the oven queue, it dequeues the order and unlocks the oven queue mutex.
It checks if the order has been canceled (order->completed == -1) and skips processing if so.
It then locks the oven mutex to check if the oven has space to cook the order.
If the oven is full, it waits on the oven condition variable until space is available or a stop signal is received.
If a stop signal is received during waiting, it unlocks the mutex and breaks out of the loop.

Once space is available in the oven, it increments the oven count (number of orders being cooked).
It simulates cooking time using a function simulate_pseudo_inverse_time_half() (not shown) that represents the cooking duration.

After cooking, it logs the completion of the order and signals the client about the order's status.
It decrements the oven count, signals the oven condition variable for other orders to be cooked, and unlocks the oven mutex.

It locks the cooked order queue mutex to add the cooked order to the queue.
After adding the order, it signals the cooked queue condition variable for delivery persons to pick up cooked orders.
It then unlocks the cooked queue mutex.

The function returns once the stop signal is received, indicating the end of oven management.

cook_function:
The function runs in a continuous loop until a stop signal is received.
It locks the order mutex to access orders waiting to be prepared.
If there are no orders in the queue and no stop signal, it waits on the order condition variable.
If a stop signal is received during waiting, it unlocks the mutex and breaks out of the loop.

If an order is available in the order queue, it dequeues the order and unlocks the order mutex.
It checks if the order has been canceled (order->completed == -1) and skips processing if so.
It marks the cook as busy, increments the order count handled by the cook, and logs the start of order preparation.

It simulates preparation time using a function simulate_pseudo_inverse_time() that represents the preparation duration.

After preparation, it marks the cook as not busy, logs the completion of order preparation, and notifies the client about order status.
It then locks the oven queue mutex to place the prepared order in the oven queue for cooking.

The function returns once the stop signal is received, indicating the end of cook's work.

delivery_function
The function runs in a continuous loop until a stop signal is received.
It locks the cooked order queue mutex to access orders ready for delivery.
If the delivery person can accept more orders and there are orders in the queue, it dequeues orders and adds them to the delivery person's deliveries.

It calculates the total delivery distance and estimates delivery time based on the delivery person's speed.
For each order in deliveries, it calculates the distance traveled and updates the delivery time accordingly.
It then simulates delivery time by sleeping for that duration and signals the client about order completion.

After completing deliveries, it calculates the return distance and time to the shop based on the last delivery's location.

It simulates return time by sleeping and logs the return of the delivery person to the shop.
Checks if the clients all order’s are delivered in every order deliveration.

If a stop signal is received during any stage, it breaks out of the loop and exits the function.

client_handler:
The function takes the client socket as an argument and frees the memory allocated for the socket argument.
It locks the client mutex to update the client socket array with the new client socket.

The function reads messages sent by clients until either a stop signal is received or an error occurs during reading.
If the message indicates order cancellation, it cancels orders associated with the client and breaks out of the loop.
Otherwise, it parses the message to extract order details and adds them to the order queue if space is available.

When adding an order to the order queue, it checks if the queue is full. If not, it adds the order and broadcasts a signal to wake up waiting threads that may be processing orders.

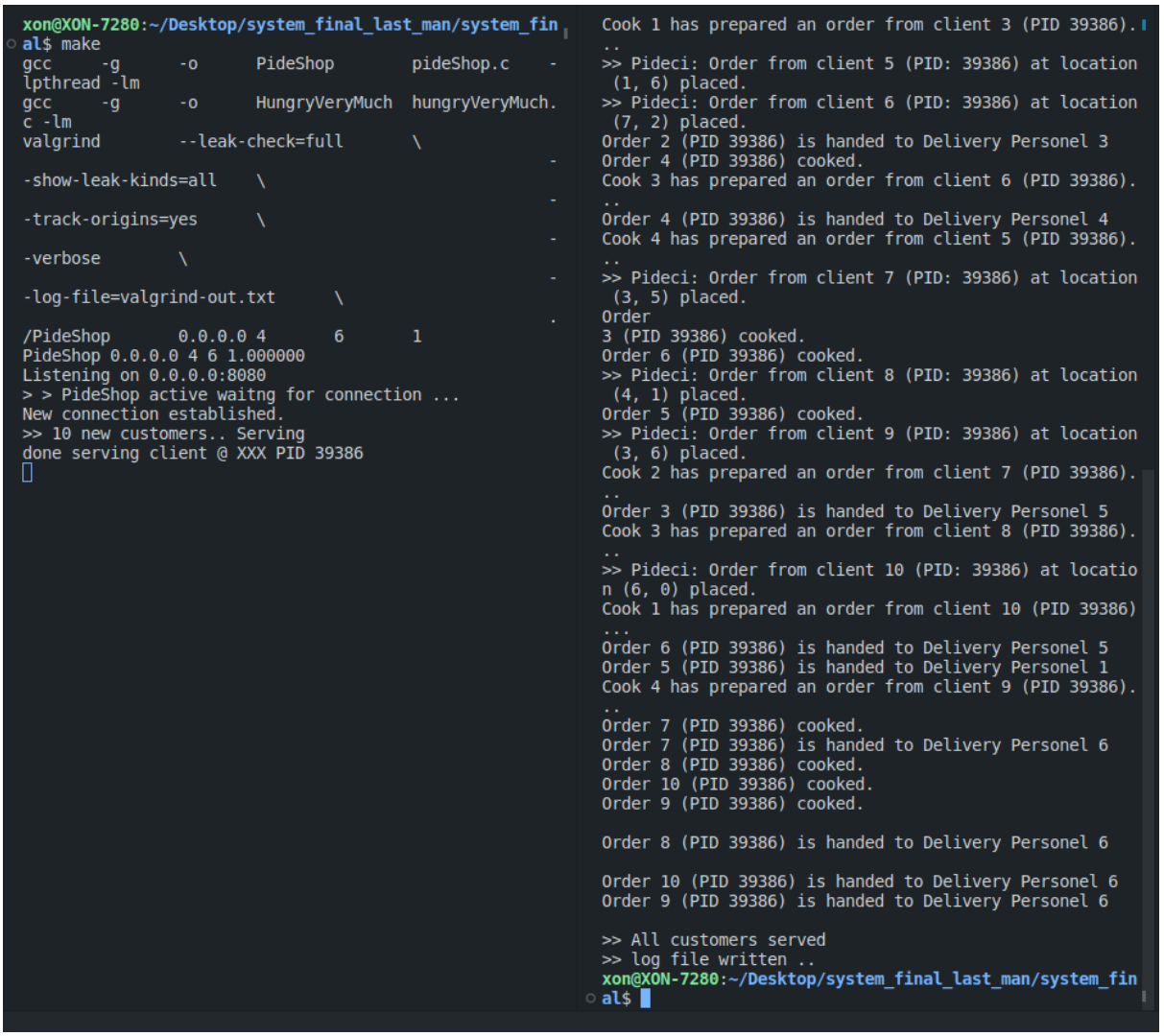
It responds to clients with messages confirming the placement of orders or notifying them of order cancellation.

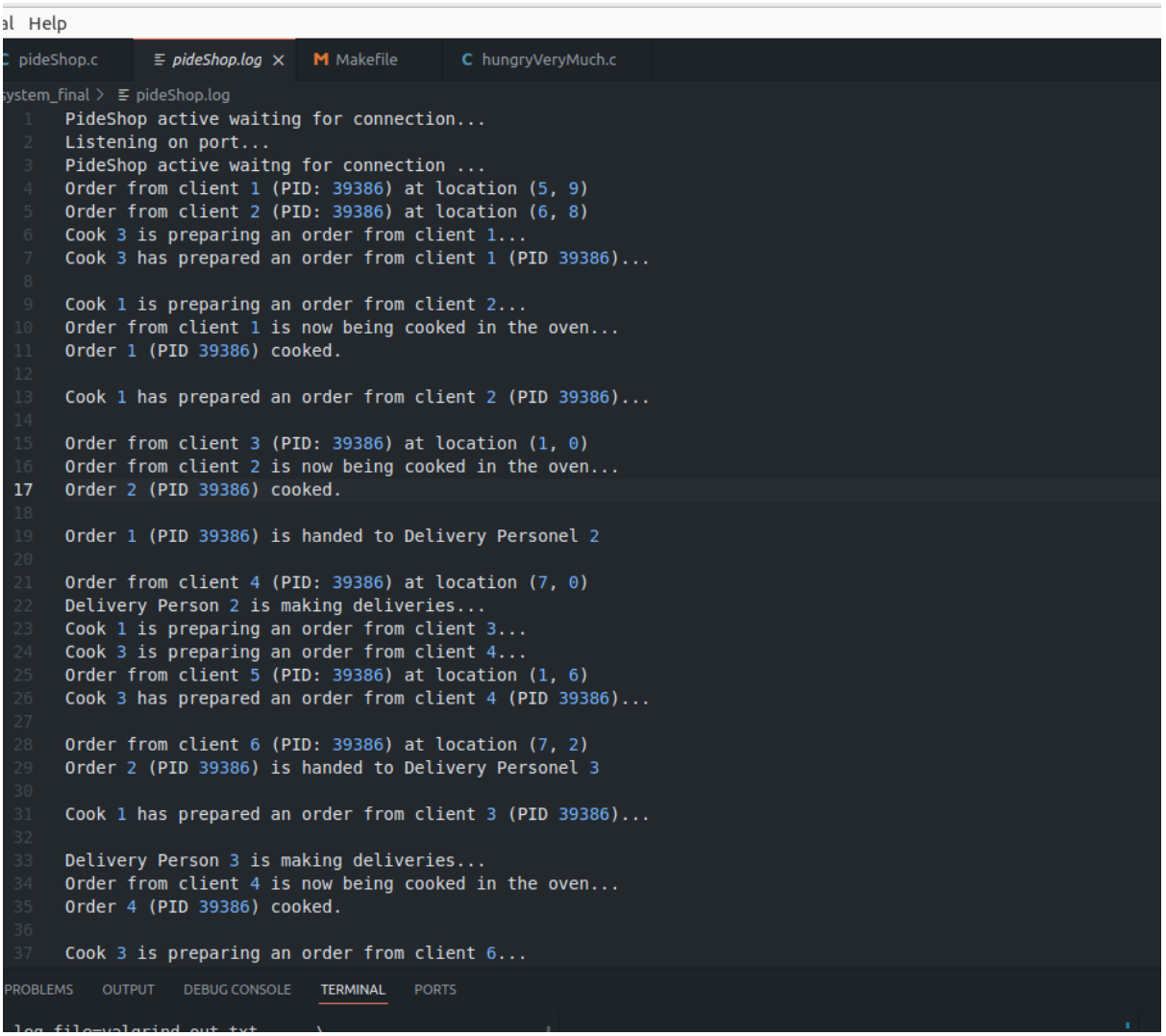
After processing messages or encountering errors, it closes the client socket and returns.

The hardest part of this project was signal handling and preventing memory leaks.

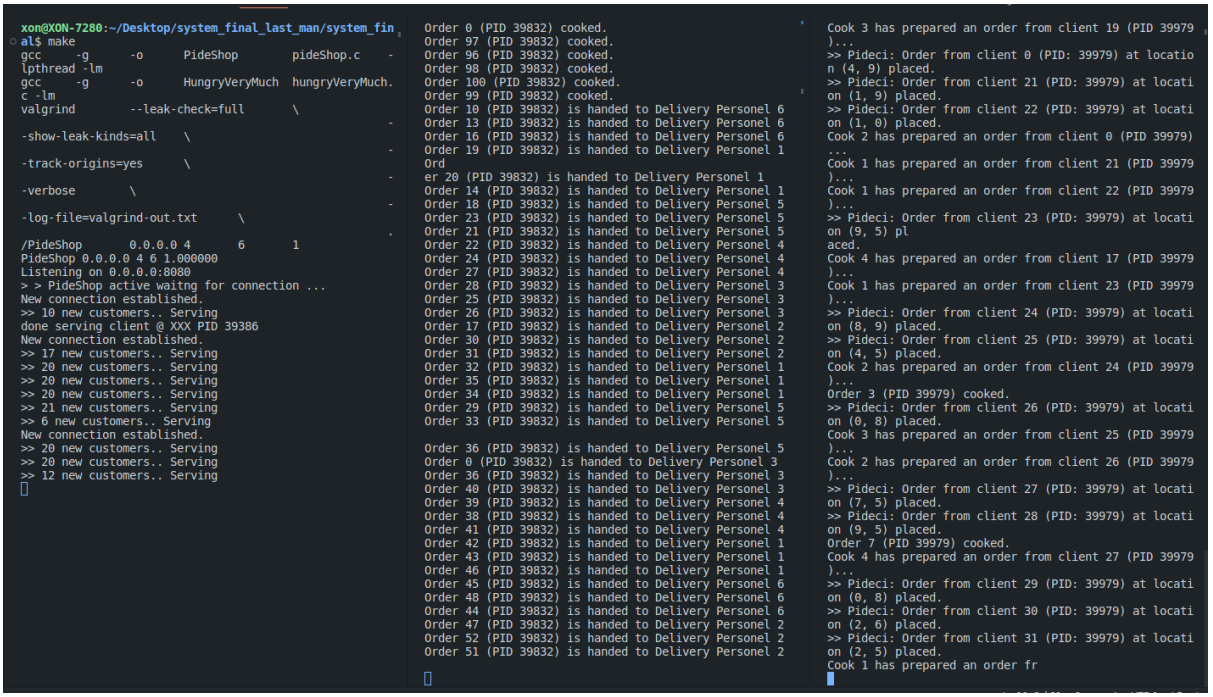
Tests:

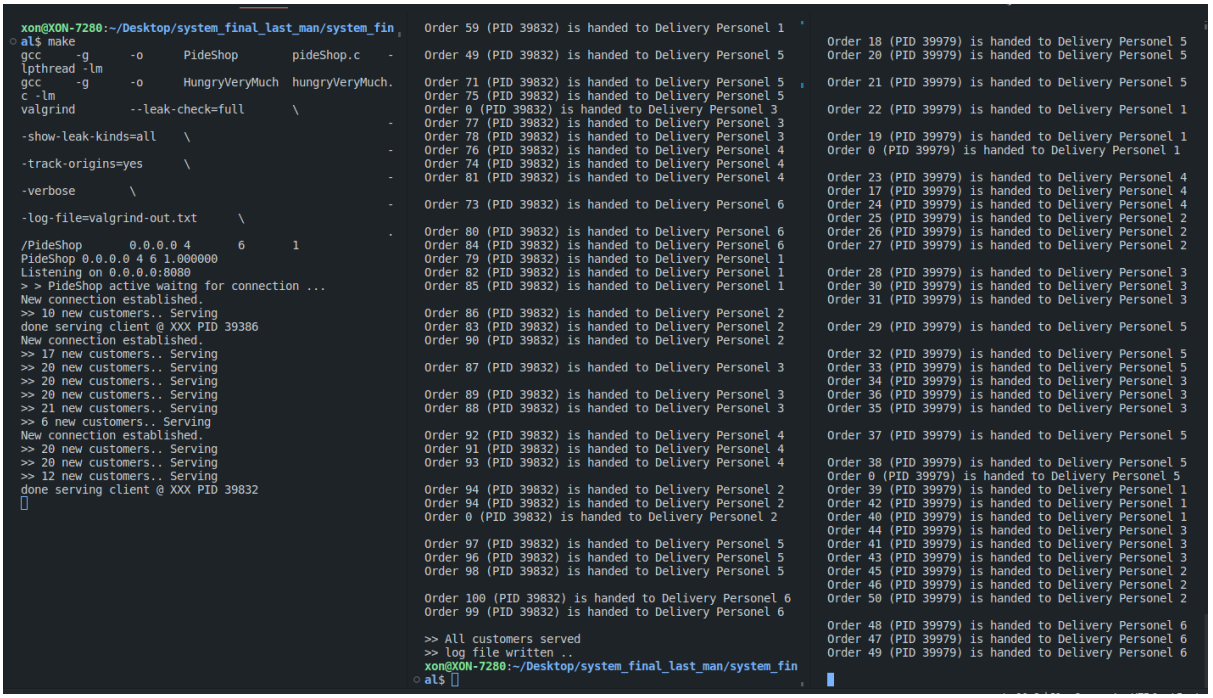
Simple one to one 10 customer

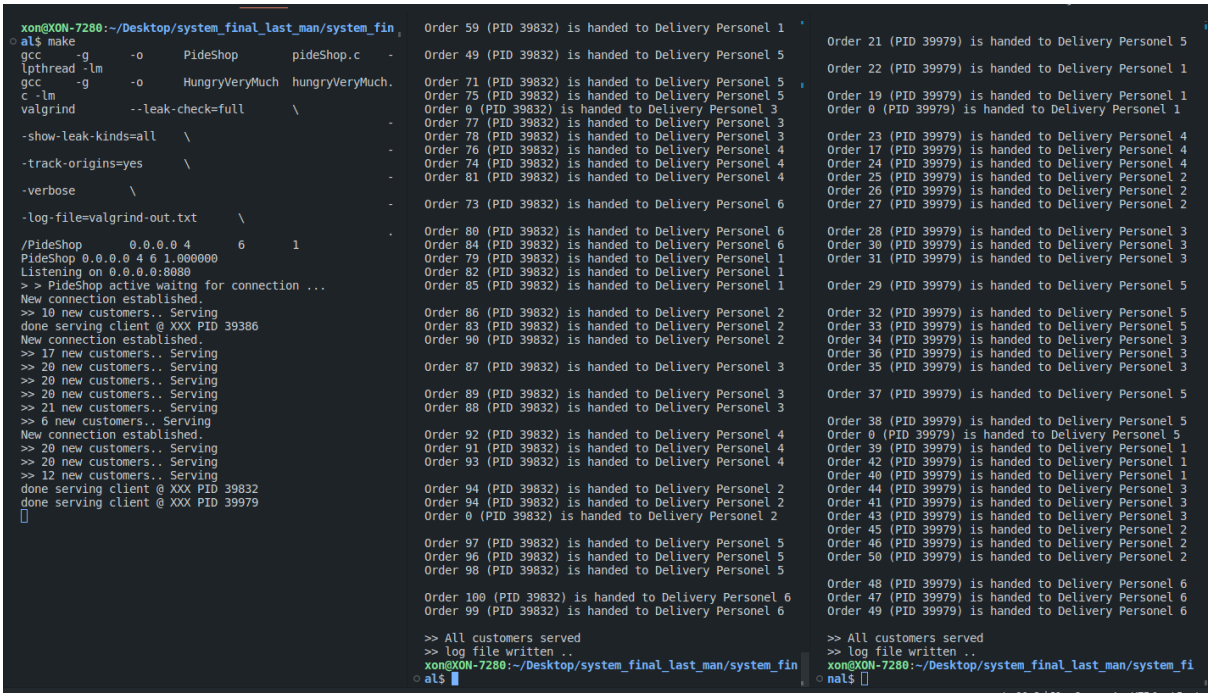




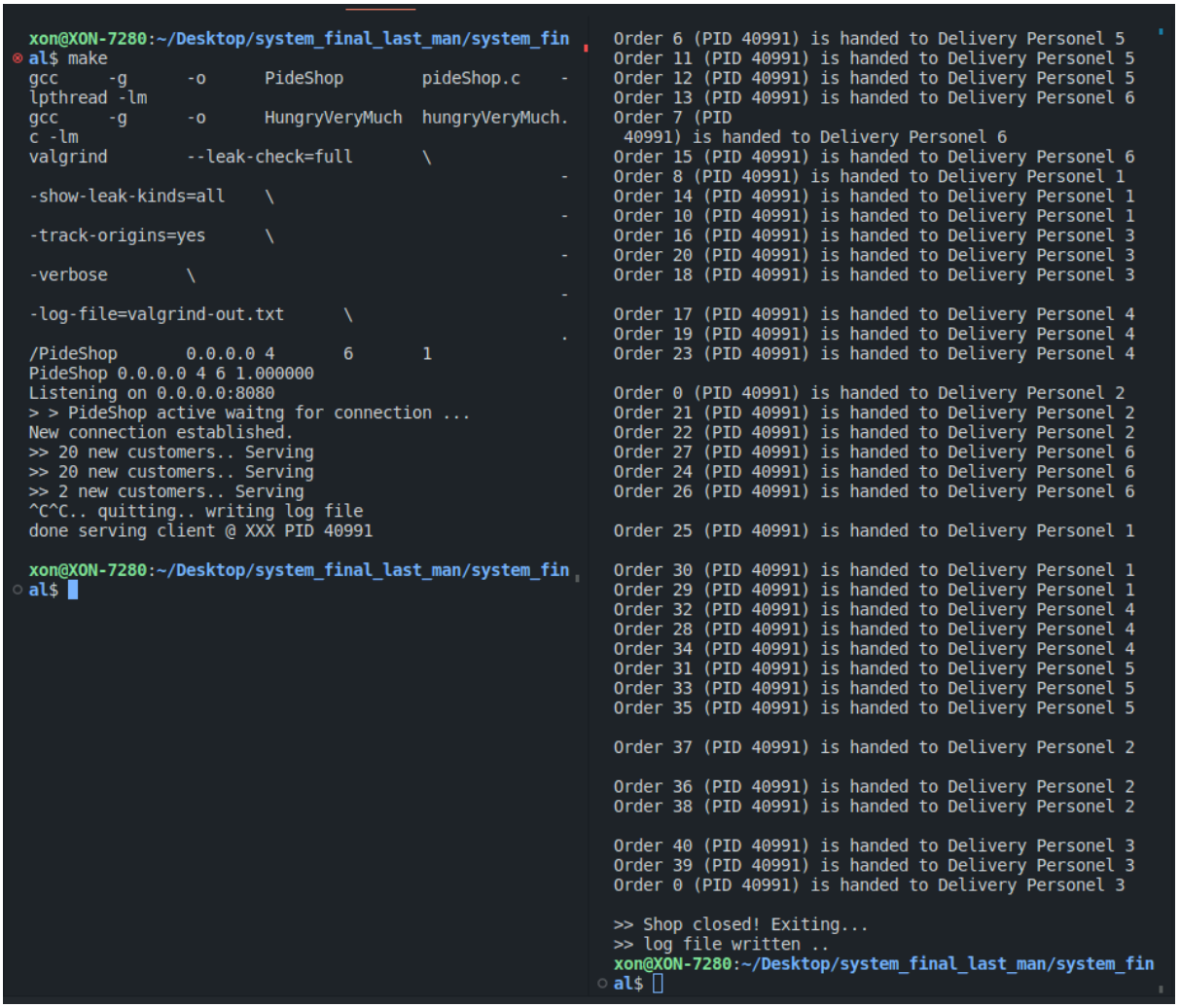
Without closing the previous server, sending multiple clients at the same time with different terminals and using large amount of clients


xon@XON-7280:~/Desktop/system_final_last_man/system_fin
al$ make
gcc -g -o PideShop pideShop.c -lpthread -lm
gcc -g -o HungryVeryMuch hungryVeryMuch.c -lm
valgrind --leak-check=full \
--show-leak-kinds=all \
--track-origins=yes \
--verbose \
--log-file=valgrind-out.txt \
/PideShop 0.0.0.0 4 6 1
PideShop 0.0.0.0 4 6 1.000000
Listening on 0.0.0.0:8080
> > PideShop active waitng for connection ...
New connection established.
>> 10 new customers.. Serving
done serving client @ XXX PID 39386
New connection established.
>> 17 new customers.. Serving
>> 20 new customers.. Serving
>> 20 new customers.. Serving
>> 20 new customers.. Serving
>> 21 new customers.. Serving
>> 6 new customers.. Serving
New connection established.
>> 20 new customers.. Serving
>> 20 new customers.. Serving
>> 12 new customers.. Serving
Order 0 (PID 39832) cooked.
Order 97 (PID 39832) cooked.
Order 96 (PID 39832) cooked.
Order 98 (PID 39832) cooked.
Order 100 (PID 39832) cooked.
Order 99 (PID 39832) cooked.
Order 10 (PID 39832) is handed to Delivery Personel 6
Order 13 (PID 39832) is handed to Delivery Personel 6
Order 16 (PID 39832) is handed to Delivery Personel 6
Order 19 (PID 39832) is handed to Delivery Personel 1
Order 20 (PID 39832) is handed to Delivery Personel 1
Order 14 (PID 39832) is handed to Delivery Personel 1
Order 18 (PID 39832) is handed to Delivery Personel 5
Order 23 (PID 39832) is handed to Delivery Personel 5
Order 21 (PID 39832) is handed to Delivery Personel 5
Order 22 (PID 39832) is handed to Delivery Personel 4
Order 24 (PID 39832) is handed to Delivery Personel 4
Order 27 (PID 39832) is handed to Delivery Personel 4
Order 28 (PID 39832) is handed to Delivery Personel 3
Order 25 (PID 39832) is handed to Delivery Personel 3
Order 26 (PID 39832) is handed to Delivery Personel 3
Order 17 (PID 39832) is handed to Delivery Personel 2
Order 30 (PID 39832) is handed to Delivery Personel 2
Order 31 (PID 39832) is handed to Delivery Personel 2
Order 32 (PID 39832) is handed to Delivery Personel 1
Order 35 (PID 39832) is handed to Delivery Personel 1
Order 34 (PID 39832) is handed to Delivery Personel 1
Order 29 (PID 39832) is handed to Delivery Personel 5
Order 33 (PID 39832) is handed to Delivery Personel 5
Order 36 (PID 39832) is handed to Delivery Personel 5
Order 0 (PID 39832) is handed to Delivery Personel 3
Order 36 (PID 39832) is handed to Delivery Personel 3
Order 40 (PID 39832) is handed to Delivery Personel 3
Order 39 (PID 39832) is handed to Delivery Personel 4
Order 38 (PID 39832) is handed to Delivery Personel 4
Order 41 (PID 39832) is handed to Delivery Personel 4
Order 42 (PID 39832) is handed to Delivery Personel 1
Order 43 (PID 39832) is handed to Delivery Personel 1
Order 46 (PID 39832) is handed to Delivery Personel 1
Order 45 (PID 39832) is handed to Delivery Personel 6
Order 48 (PID 39832) is handed to Delivery Personel 6
Order 44 (PID 39832) is handed to Delivery Personel 6
Order 47 (PID 39832) is handed to Delivery Personel 2
Order 52 (PID 39832) is handed to Delivery Personel 2
Order 51 (PID 39832) is handed to Delivery Personel 2
Cook 3 has prepared an order from client 19 (PID 39979)...
>> Pideci: Order from client 0 (PID: 39979) at location (4, 9) placed.
>> Pideci: Order from client 21 (PID: 39979) at location (1, 9) placed.
>> Pideci: Order from client 22 (PID: 39979) at location (1, 0) placed.
Cook 2 has prepared an order from client 0 (PID 39979)...
Cook 1 has prepared an order from client 21 (PID 39979)...
Cook 1 has prepared an order from client 22 (PID 39979)...
>> Pideci: Order from client 23 (PID: 39979) at location (9, 5) placed.
Cook 4 has prepared an order from client 17 (PID 39979)...
Cook 1 has prepared an order from client 23 (PID 39979)...
>> Pideci: Order from client 24 (PID: 39979) at location (8, 9) placed.
>> Pideci: Order from client 25 (PID: 39979) at location (4, 5) placed.
Cook 2 has prepared an order from client 24 (PID 39979)...
Order 3 (PID 39979) cooked.
>> Pideci: Order from client 26 (PID: 39979) at location (0, 8) placed.
Cook 3 has prepared an order from client 25 (PID 39979)...
Cook 2 has prepared an order from client 26 (PID 39979)...
>> Pideci: Order from client 27 (PID: 39979) at location (7, 5) placed.
>> Pideci: Order from client 28 (PID: 39979) at location (9, 5) placed.
Order 7 (PID 39979) cooked.
Cook 4 has prepared an order from client 27 (PID 39979)...
>> Pideci: Order from client 29 (PID: 39979) at location (0, 8) placed.
>> Pideci: Order from client 30 (PID: 39979) at location (2, 6) placed.
>> Pideci: Order from client 31 (PID: 39979) at location (2, 5) placed.
Cook 1 has prepared an order fr


xon@XON-7280:~/Desktop/system_final_last_man/system_fin
al$ make
gcc -g -o PideShop pideShop.c -lpthread -lm
gcc -g -o HungryVeryMuch hungryVeryMuch.c -lm
valgrind --leak-check=full \
--show-leak-kinds=all \
--track-origins=yes \
--verbose \
--log-file=valgrind-out.txt \
/PideShop 0.0.0.0 4 6 1
PideShop 0.0.0.0 4 6 1.000000
Listening on 0.0.0.0:8080
> > PideShop active waitng for connection ...
New connection established.
>> 10 new customers.. Serving
done serving client @ XXX PID 39386
New connection established.
>> 17 new customers.. Serving
>> 20 new customers.. Serving
>> 20 new customers.. Serving
>> 20 new customers.. Serving
>> 21 new customers.. Serving
>> 6 new customers.. Serving
New connection established.
>> 20 new customers.. Serving
>> 20 new customers.. Serving
>> 12 new customers.. Serving
done serving client @ XXX PID 39832
Order 59 (PID 39832) is handed to Delivery Personel 1
Order 49 (PID 39832) is handed to Delivery Personel 5
Order 71 (PID 39832) is handed to Delivery Personel 5
Order 75 (PID 39832) is handed to Delivery Personel 5
Order 0 (PID 39832) is handed to Delivery Personel 3
Order 77 (PID 39832) is handed to Delivery Personel 3
Order 78 (PID 39832) is handed to Delivery Personel 3
Order 76 (PID 39832) is handed to Delivery Personel 4
Order 74 (PID 39832) is handed to Delivery Personel 4
Order 81 (PID 39832) is handed to Delivery Personel 4
Order 73 (PID 39832) is handed to Delivery Personel 6
Order 80 (PID 39832) is handed to Delivery Personel 6
Order 84 (PID 39832) is handed to Delivery Personel 6
Order 79 (PID 39832) is handed to Delivery Personel 1
Order 82 (PID 39832) is handed to Delivery Personel 1
Order 85 (PID 39832) is handed to Delivery Personel 1
Order 86 (PID 39832) is handed to Delivery Personel 2
Order 83 (PID 39832) is handed to Delivery Personel 2
Order 90 (PID 39832) is handed to Delivery Personel 2
Order 87 (PID 39832) is handed to Delivery Personel 3
Order 89 (PID 39832) is handed to Delivery Personel 3
Order 88 (PID 39832) is handed to Delivery Personel 3
Order 92 (PID 39832) is handed to Delivery Personel 4
Order 91 (PID 39832) is handed to Delivery Personel 4
Order 93 (PID 39832) is handed to Delivery Personel 4
Order 94 (PID 39832) is handed to Delivery Personel 2
Order 94 (PID 39832) is handed to Delivery Personel 2
Order 0 (PID 39832) is handed to Delivery Personel 2
Order 97 (PID 39832) is handed to Delivery Personel 5
Order 96 (PID 39832) is handed to Delivery Personel 5
Order 98 (PID 39832) is handed to Delivery Personel 5
Order 100 (PID 39832) is handed to Delivery Personel 6
Order 99 (PID 39832) is handed to Delivery Personel 6
>> All customers served
>> log file written ..
xon@XON-7280:~/Desktop/system_final_last_man/system_fin
al$
Order 18 (PID 39979) is handed to Delivery Personel 5
Order 20 (PID 39979) is handed to Delivery Personel 5
Order 21 (PID 39979) is handed to Delivery Personel 5
Order 22 (PID 39979) is handed to Delivery Personel 1
Order 19 (PID 39979) is handed to Delivery Personel 1
Order 0 (PID 39979) is handed to Delivery Personel 1
Order 23 (PID 39979) is handed to Delivery Personel 4
Order 17 (PID 39979) is handed to Delivery Personel 4
Order 24 (PID 39979) is handed to Delivery Personel 4
Order 25 (PID 39979) is handed to Delivery Personel 2
Order 26 (PID 39979) is handed to Delivery Personel 2
Order 27 (PID 39979) is handed to Delivery Personel 2
Order 28 (PID 39979) is handed to Delivery Personel 3
Order 30 (PID 39979) is handed to Delivery Personel 3
Order 31 (PID 39979) is handed to Delivery Personel 3
Order 29 (PID 39979) is handed to Delivery Personel 5
Order 32 (PID 39979) is handed to Delivery Personel 5
Order 33 (PID 39979) is handed to Delivery Personel 5
Order 34 (PID 39979) is handed to Delivery Personel 3
Order 36 (PID 39979) is handed to Delivery Personel 3
Order 35 (PID 39979) is handed to Delivery Personel 3
Order 37 (PID 39979) is handed to Delivery Personel 5
Order 38 (PID 39979) is handed to Delivery Personel 5
Order 0 (PID 39979) is handed to Delivery Personel 5
Order 39 (PID 39979) is handed to Delivery Personel 1
Order 42 (PID 39979) is handed to Delivery Personel 1
Order 40 (PID 39979) is handed to Delivery Personel 1
Order 44 (PID 39979) is handed to Delivery Personel 3
Order 41 (PID 39979) is handed to Delivery Personel 3
Order 43 (PID 39979) is handed to Delivery Personel 3
Order 45 (PID 39979) is handed to Delivery Personel 2
Order 46 (PID 39979) is handed to Delivery Personel 2
Order 50 (PID 39979) is handed to Delivery Personel 2
Order 48 (PID 39979) is handed to Delivery Personel 6
Order 47 (PID 39979) is handed to Delivery Personel 6
Order 49 (PID 39979) is handed to Delivery Personel 6


xon@XON-7280:~/Desktop/system_final_last_man/system_fin
al$ make
gcc -g -o PideShop pideShop.c -lpthread -lm
gcc -g -o HungryVeryMuch hungryVeryMuch.c -lm
valgrind --leak-check=full \
--show-leak-kinds=all \
--track-origins=yes \
--verbose \
--log-file=valgrind-out.txt \
./PideShop 0.0.0.0 4 6 1
PideShop 0.0.0.0 4 6 1.000000
Listening on 0.0.0.0:8080
> > PideShop active waitng for connection ...
New connection established.
>> 10 new customers.. Serving
done serving client @ XXX PID 39386
New connection established.
>> 17 new customers.. Serving
>> 20 new customers.. Serving
>> 20 new customers.. Serving
>> 20 new customers.. Serving
>> 21 new customers.. Serving
>> 6 new customers.. Serving
New connection established.
>> 20 new customers.. Serving
>> 20 new customers.. Serving
>> 12 new customers.. Serving
done serving client @ XXX PID 39832
done serving client @ XXX PID 39979
Order 59 (PID 39832) is handed to Delivery Personel 1
Order 49 (PID 39832) is handed to Delivery Personel 5
Order 71 (PID 39832) is handed to Delivery Personel 5
Order 75 (PID 39832) is handed to Delivery Personel 5
Order 0 (PID 39832) is handed to Delivery Personel 3
Order 77 (PID 39832) is handed to Delivery Personel 3
Order 78 (PID 39832) is handed to Delivery Personel 3
Order 76 (PID 39832) is handed to Delivery Personel 4
Order 74 (PID 39832) is handed to Delivery Personel 4
Order 81 (PID 39832) is handed to Delivery Personel 4
Order 73 (PID 39832) is handed to Delivery Personel 6
Order 80 (PID 39832) is handed to Delivery Personel 6
Order 84 (PID 39832) is handed to Delivery Personel 6
Order 79 (PID 39832) is handed to Delivery Personel 1
Order 82 (PID 39832) is handed to Delivery Personel 1
Order 85 (PID 39832) is handed to Delivery Personel 1
Order 86 (PID 39832) is handed to Delivery Personel 2
Order 83 (PID 39832) is handed to Delivery Personel 2
Order 90 (PID 39832) is handed to Delivery Personel 2
Order 87 (PID 39832) is handed to Delivery Personel 3
Order 89 (PID 39832) is handed to Delivery Personel 3
Order 88 (PID 39832) is handed to Delivery Personel 3
Order 92 (PID 39832) is handed to Delivery Personel 4
Order 91 (PID 39832) is handed to Delivery Personel 4
Order 93 (PID 39832) is handed to Delivery Personel 4
Order 94 (PID 39832) is handed to Delivery Personel 2
Order 94 (PID 39832) is handed to Delivery Personel 2
Order 0 (PID 39832) is handed to Delivery Personel 2
Order 97 (PID 39832) is handed to Delivery Personel 5
Order 96 (PID 39832) is handed to Delivery Personel 5
Order 98 (PID 39832) is handed to Delivery Personel 5
Order 100 (PID 39832) is handed to Delivery Personel 6
Order 99 (PID 39832) is handed to Delivery Personel 6
>> All customers served
>> log file written ..
xon@XON-7280:~/Desktop/system_final_last_man/system_fin
al$
Order 21 (PID 39979) is handed to Delivery Personel 5
Order 22 (PID 39979) is handed to Delivery Personel 1
Order 19 (PID 39979) is handed to Delivery Personel 1
Order 0 (PID 39979) is handed to Delivery Personel 1
Order 23 (PID 39979) is handed to Delivery Personel 4
Order 17 (PID 39979) is handed to Delivery Personel 4
Order 24 (PID 39979) is handed to Delivery Personel 4
Order 25 (PID 39979) is handed to Delivery Personel 2
Order 26 (PID 39979) is handed to Delivery Personel 2
Order 27 (PID 39979) is handed to Delivery Personel 2
Order 28 (PID 39979) is handed to Delivery Personel 3
Order 30 (PID 39979) is handed to Delivery Personel 3
Order 31 (PID 39979) is handed to Delivery Personel 3
Order 29 (PID 39979) is handed to Delivery Personel 5
Order 32 (PID 39979) is handed to Delivery Personel 5
Order 33 (PID 39979) is handed to Delivery Personel 5
Order 34 (PID 39979) is handed to Delivery Personel 3
Order 36 (PID 39979) is handed to Delivery Personel 3
Order 35 (PID 39979) is handed to Delivery Personel 3
Order 37 (PID 39979) is handed to Delivery Personel 5
Order 38 (PID 39979) is handed to Delivery Personel 5
Order 0 (PID 39979) is handed to Delivery Personel 5
Order 39 (PID 39979) is handed to Delivery Personel 1
Order 42 (PID 39979) is handed to Delivery Personel 1
Order 40 (PID 39979) is handed to Delivery Personel 1
Order 44 (PID 39979) is handed to Delivery Personel 3
Order 41 (PID 39979) is handed to Delivery Personel 3
Order 43 (PID 39979) is handed to Delivery Personel 3
Order 45 (PID 39979) is handed to Delivery Personel 2
Order 46 (PID 39979) is handed to Delivery Personel 2
Order 50 (PID 39979) is handed to Delivery Personel 2
Order 48 (PID 39979) is handed to Delivery Personel 6
Order 47 (PID 39979) is handed to Delivery Personel 6
Order 49 (PID 39979) is handed to Delivery Personel 6
>> All customers served
>> log file written ..
xon@XON-7280:~/Desktop/system_final_last_man/system_fi
nal$

Sending kill to Server (it may take some time but it definitely finishes after cleaning)



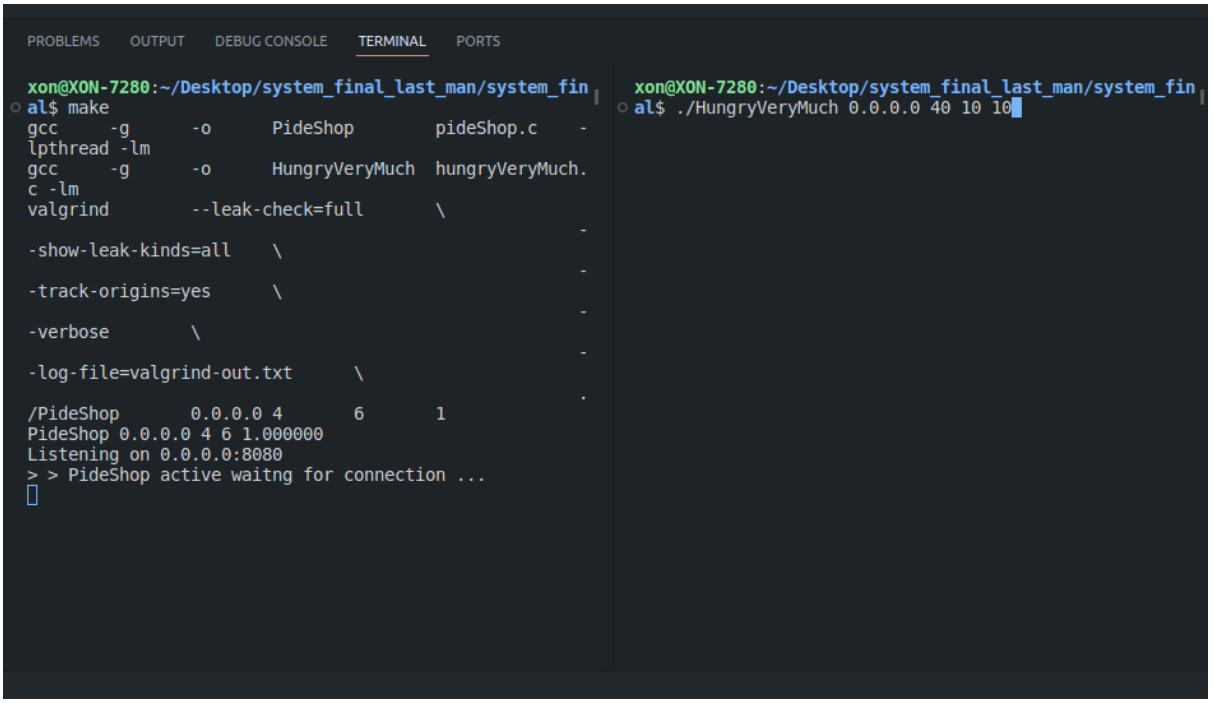
```
--40884-- REDIR: 0x49f93f0 (libc.so.6:memchr) redirected to 0x483f220 (_vgnU_ifunc_wrapper)
--40884-- REDIR: 0x49f8b00 (libc.so.6:strspn) redirected to 0x483f220 (_vgnU_ifunc_wrapper)
--40884-- REDIR: 0x49f97d0 (libc.so.6:mempcpy) redirected to 0x483f220 (_vgnU_ifunc_wrapper)
--40884-- REDIR: 0x49f9ac0 (libc.so.6:strncasecmp) redirected to 0x483f220 (_vgnU_ifunc_wrapper)
--40884-- REDIR: 0x49faf80 (libc.so.6:rawmemchr) redirected to 0x483f220 (_vgnU_ifunc_wrapper)
--40884-- REDIR: 0x49f9310 (libc.so.6:strstr) redirected to 0x483f220 (_vgnU_ifunc_wrapper)
--40884-- REDIR: 0x4aed610 (libc.so.6:__strrchr_avx2) redirected to 0x484e810 (rindex)
--40884-- REDIR: 0x4aed7e0 (libc.so.6:__strlen_avx2) redirected to 0x484ed10 (strlen)
--40884-- REDIR: 0x49f50a0 (libc.so.6:malloc) redirected to 0x4848820 (malloc)
--40884-- REDIR: 0x49f8fc0 (libc.so.6:__GI_strstr) redirected to 0x4853ae0 (__strstr_sse2)
--40884-- REDIR: 0x4aed400 (libc.so.6:__strchrnul_avx2) redirected to 0x4853330 (strchrnul)
--40884-- REDIR: 0x4af0780 (libc.so.6:__mempcpy_avx_unaligned_erms) redirected to 0x4853440 (mempcpy)
--40884-- REDIR: 0x49f6520 (libc.so.6:calloc) redirected to 0x484d9d0 (calloc)
--40884-- REDIR: 0x49f53e0 (libc.so.6:free) redirected to 0x484b210 (free)
--40884-- REDIR: 0x4a14180 (libc.so.6:__strstr_sse2_unaligned) redirected to 0x4853a50 (strstr)
--40884-- REDIR: 0x4ae8800 (libc.so.6:__strspn_sse42) redirected to 0x4853d50 (strspn)
--40884-- REDIR: 0x4ae85a0 (libc.so.6:__strcspn_sse42) redirected to 0x4853c70 (strcspn)
--40884-- REDIR: 0x4ae9590 (libc.so.6:__rawmemchr_avx2) redirected to 0x4853360 (rawmemchr)
==40884==
==40884== HEAP SUMMARY:
==40884==     in use at exit: 0 bytes in 0 blocks
==40884==   total heap usage: 16 allocs, 16 frees, 8,860 bytes allocated
==40884==
==40884== All heap blocks were freed -- no leaks are possible
==40884==
==40884== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

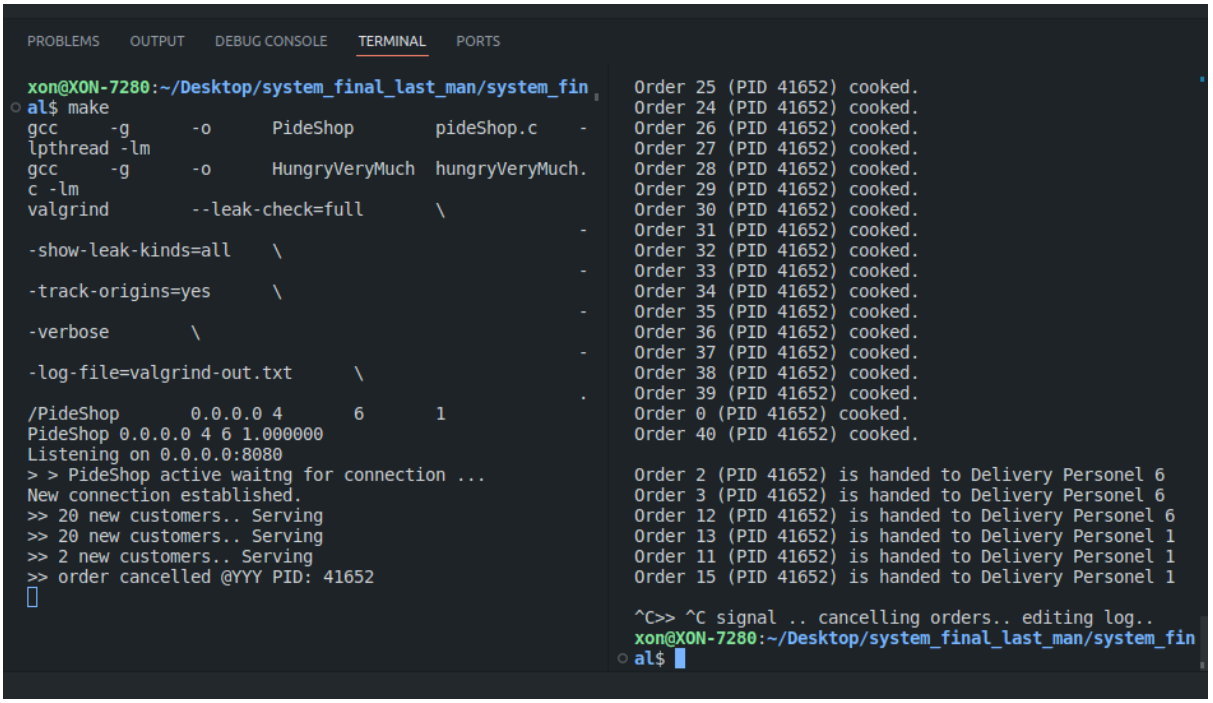
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
-verbose        \
-log-file=valgrind-out.txt      \
/PideShop       0.0.0.0 4       6       1
PideShop 0.0.0.0 4 6 1.000000
Listening on 0.0.0.0:8080
> > PideShop active waitng for connection ...
New connection established.
>> 20 new customers.. Serving
>> 20 new customers.. Serving
>> 2 new customers.. Serving
^C^C.. quitting.. writing log file
done serving client @ XXX PID 40991

xon@XON-7280:~/Desktop/system_final_last_man/system_final$
```

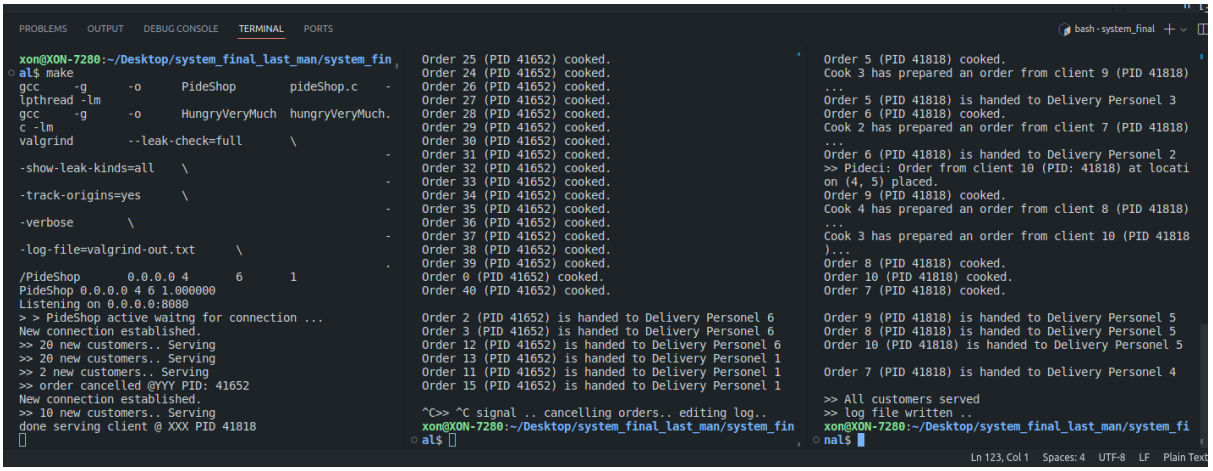
```
Order 34 (PID 40991) is handed to Delivery Personel 4
Order 31 (PID 40991) is handed to Delivery Personel 5
Order 33 (PID 40991) is handed to Delivery Personel 5
Order 35 (PID 40991) is handed to Delivery Personel 5

Order 37 (PID 40991) is handed to Delivery Personel 2

Order 36 (PID 40991) is handed to Delivery Personel 2
Order 38 (PID 40991) is handed to Delivery Personel 2

Order 40 (PID 40991) is handed to Delivery Personel 3
Order 39 (PID 40991) is handed to Delivery Personel 3
Order 0 (PID 40991) is handed to Delivery Personel 3

>> Shop closed! Exiting...
>> log file written ..
xon@XON-7280:~/Desktop/system_final_last_man/system_final$
```

Kill signal via customer:





after cancelation from one client, adding another



For further information you can contact me via email.