Barbershop Simulation in C: README

This program simulates a barbershop scenario with multiple barbers and customers, using threads to handle concurrent activities such as serving customers, waiting in line, and calculating waiting times. The program also uses mutexes and condition variables to synchronize access to shared resources like the queue and barbers' status.

Features:

- **Multiple Barbers**: Simulate up to 3 barbers serving customers.
- **Customer Queues**: Customers wait in a queue to be served. The program handles a chair queue, a sofa queue, and a standing queue.
- **Monitor Thread**: A monitor watches the queue and broadcasts a signal when customers are waiting, notifying barbers that they can start working.
- **Random Wait Time for Customers**: Customers are added to the queue after waiting for a random time.
- **Customer Thread**: Simulates a customer who waits for a random time before joining the queue.
- **Barber Thread**: Handles customer service by taking customers from the queue, serving them, and logging the total number of customers served by each barber.
- **Mutexes and Condition Variables**: Synchronize access to shared resources, such as the queue and barbers' status, ensuring thread safety.
- **Queue Management**: Handles three types of queues: chair queue (for customers in chairs), sofa queue, and standing queue. Customers are moved from the queues as barbers become available.

Compilation:

To compile and run the program, you can use gcc with the necessary libraries for threading:

gcc -o barbershop barbershop.c -lpthread

Or

gcc -o barbershop barbershop.c

Running the Program:

You will be prompted to enter the duration for which the barbershop will be open. The program will run until that time elapses, simulating the activities of customers and barbers.
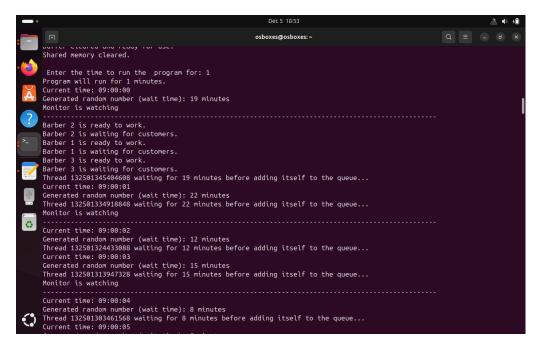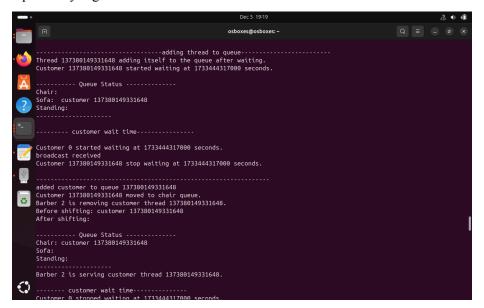
./barbershop

Key Functions:

- add_____to_____queue: Adds a customer to the queue after a random wait time.
- remove_____and_____shift_____queue: Removes and shifts customers in the queue after being served by a barber.
- register1: Simulates the checkout process for a customer.
- monitor_____function: Monitors the queue and broadcasts a signal to notify barbers when a customer is ready.
- barber_____function: Barber thread function that serves customers.
- print_____barbershop_____state: Prints the current state of the barbershop, including barbers' statuses and the current customer queues.
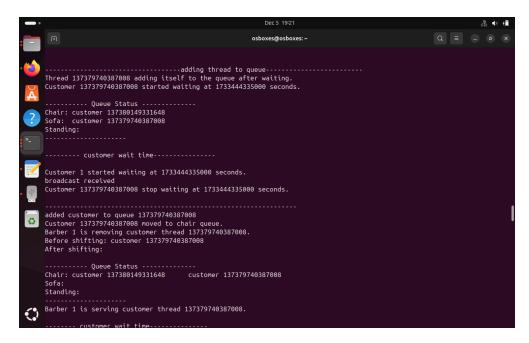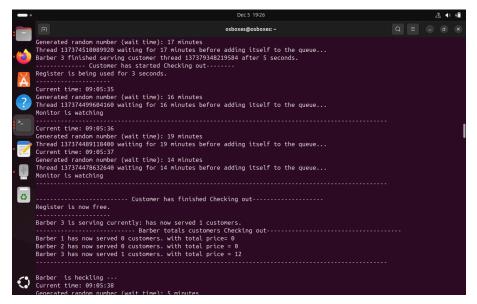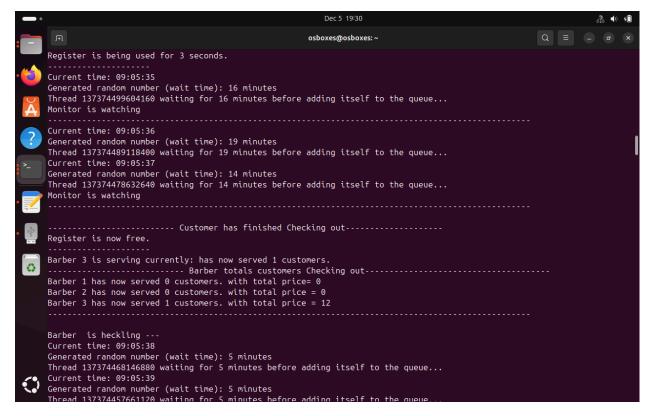
Output:
- Program start:

- Barbershop activity log

- Register

-

-

- Start and stop times : in the logs

- Total amount for each barber: price and customer

- - Average wait time

All customers have been served and the barbershop is closing: printing final register.
-------------- Customer has started Checking out--------
Register is being used for 0 seconds.

------------------------- Customer has finished Checking out--------------------
Register is now free.

Invalid barber ID: 0
---------------------------- Barber totals customers Checking out------------------------------------
Barber 1 has now served 0 customers. with total price= 0
Barber 2 has now served 0 customers. with total price = 0
Barber 3 has now served 1 customers. with total price = 12

Monitor is watching

Monitor is watching

Monitor is watching

Monitor is watching

Monitor is watching

Monitor is watching