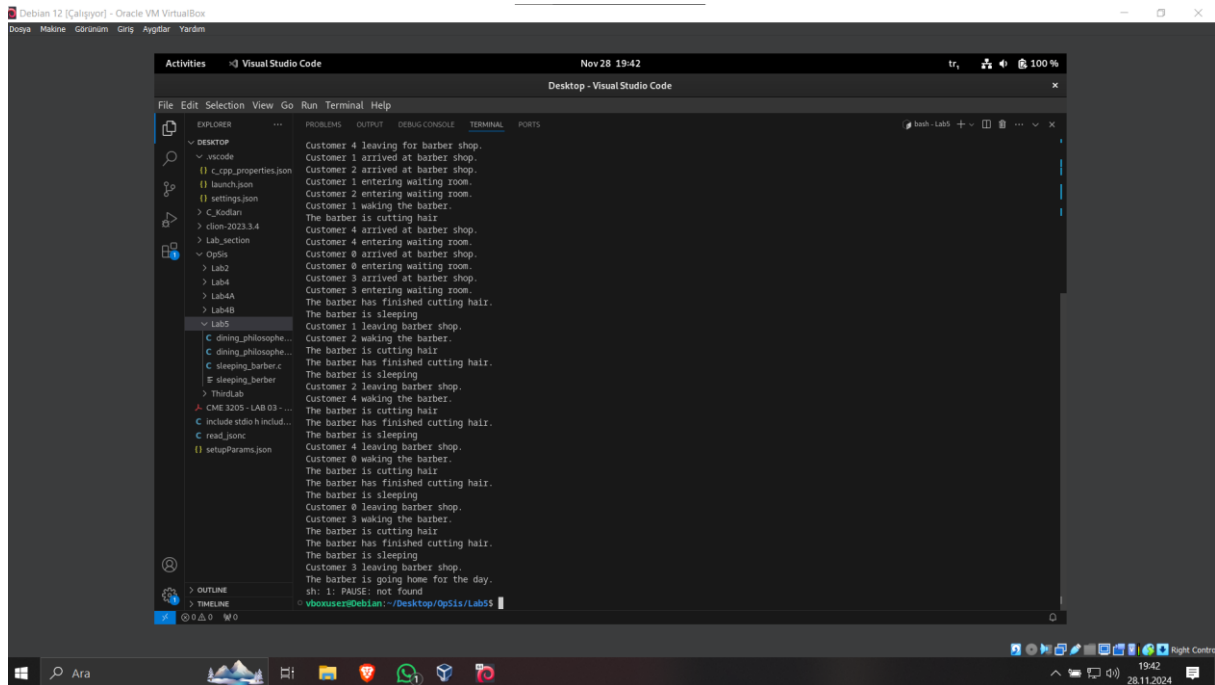


Çağrı AYDIN – 2021510010 – LAB 5

CODE 1:

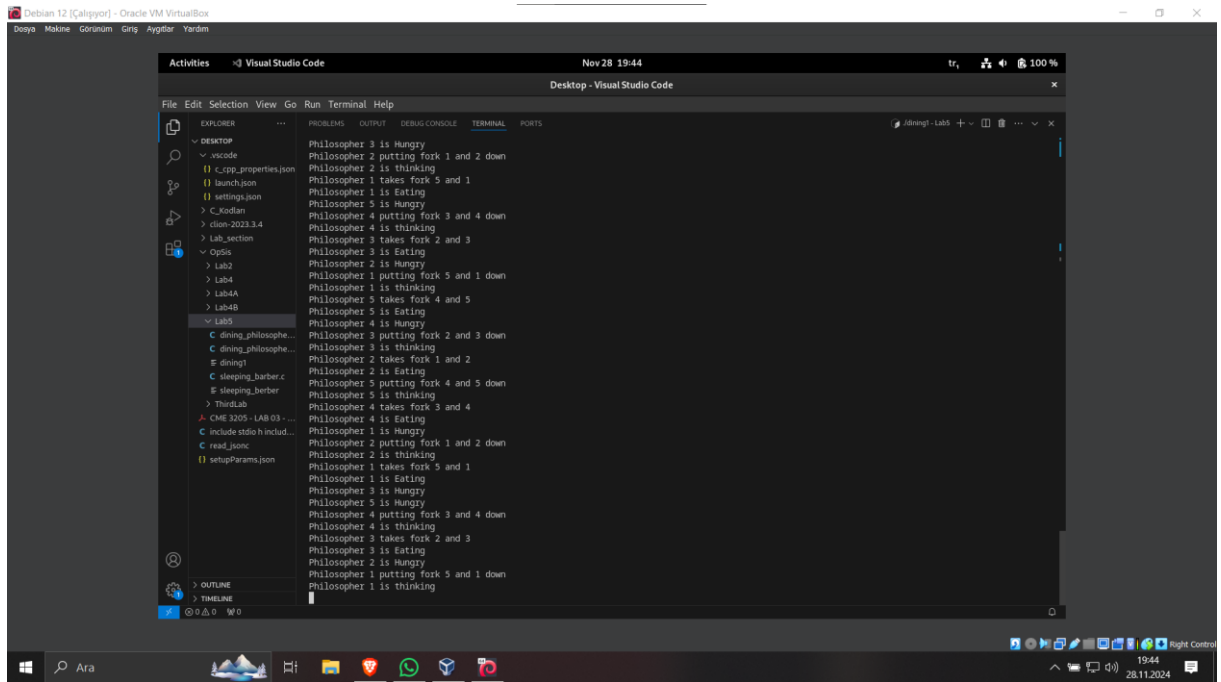


This part implements the sleeping barber problem using semaphores and threads. The barber sleeps when no customers are present, using barberPillow semaphore to wait for a customer. Customers use waitingRoom semaphore to ensure the waiting room has space and barberChair semaphore to occupy the barber's chair. The barber cuts hair and signals the customer with the seatBelt semaphore when finished. When all customers are served, the allDone flag ensures the barber stops working.

CODE 2 & 3:

In code 3 program solves the issue occurs in code 2 using semaphores and threads. Five philosophers sit at a table and share forks to eat. Each philosopher can think, get hungry, or eat. They can only eat if their left and right neighbors are not eating. Semaphores are used to control access to forks and avoid deadlocks. A mutex semaphore ensures that only one philosopher changes their state at a time. Each philosopher's thread runs in a loop, alternating between thinking and eating. This solution ensures fairness and prevents conflicts over shared resources.

CODE 2 SS:



CODE 3 SS:

