In the name of God

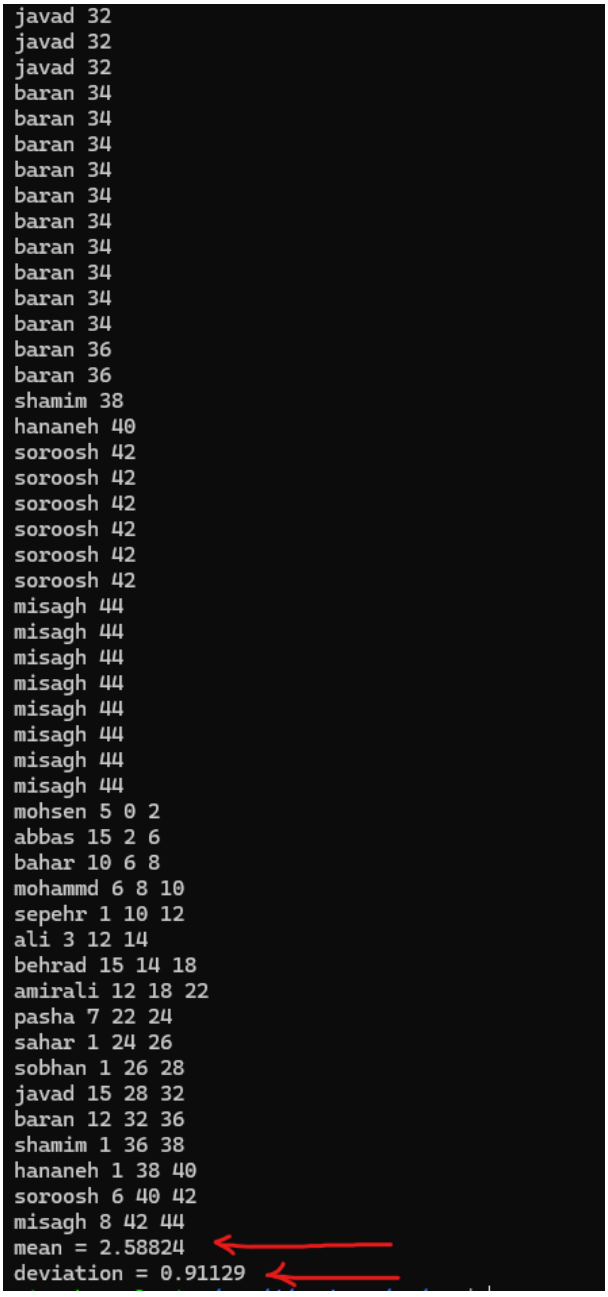
Project 3 — Parallel Programming Course

Instructor: **Dr. Safari**

Students:

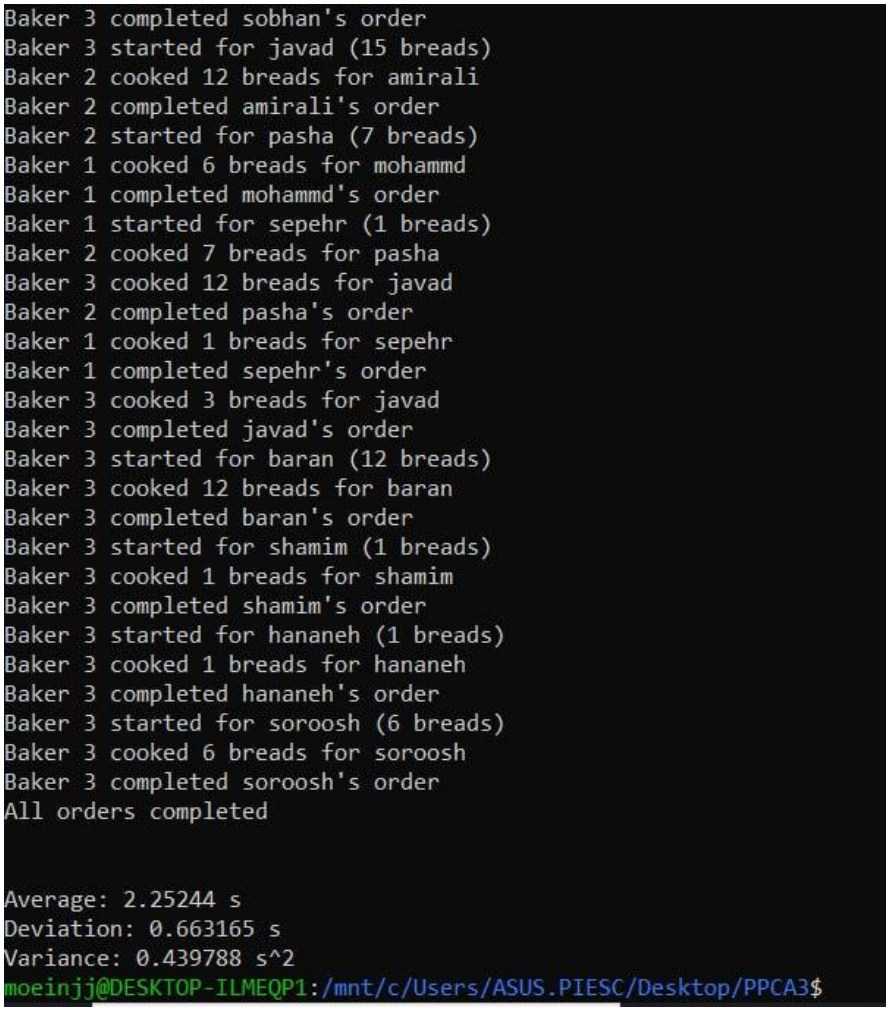
Mohamad Yahyapour

Mohammad Moien Joneidi Jaafari



Result of the **single-threaded baker**.

In the result of the **single-threaded baker** section, it shows how many breads each person ordered, when the order was received, and when it was delivered. By subtracting these two times, the **mean** and **standard deviation** can be calculated.



Result of the **multi-threaded baker without chaos**.

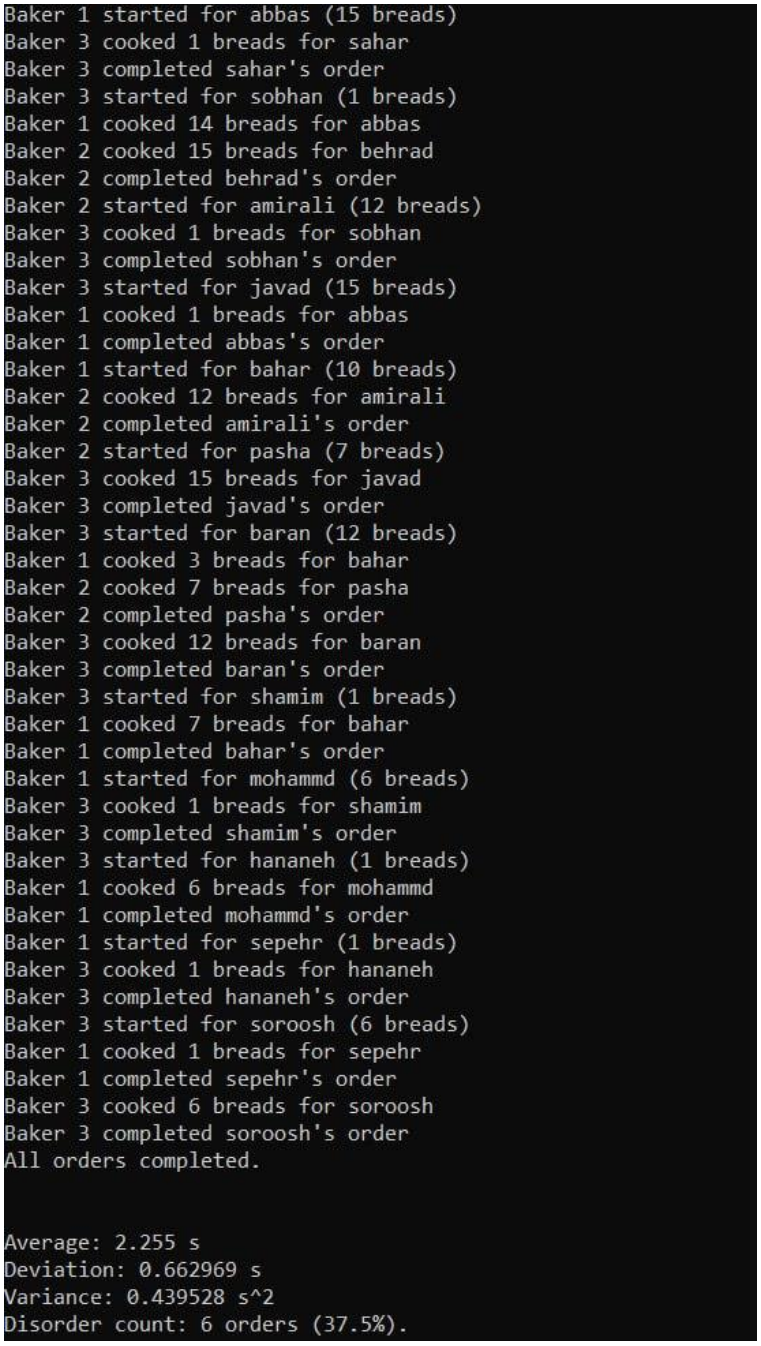
We wrote a **struct** for **order** that stores the customer's name, quantity, start time, and end time.
For time, functions from the **chrono** library are used, and for thread sleep, functions from the **thread** library are used.

A **mutex** is used for the queue, for printing output, for completion, and for checking available space and assigning bread. Each thread takes its own order from the queue and starts baking. When it is about to start baking, it **locks the oven mutex**, checks the remaining oven capacity, and chooses the smaller of the order quantity and the remaining capacity, subtracting that amount from

the oven capacity. Then it **releases the mutex** so other threads can access the oven.

Since multiple threads could print simultaneously, a **mutex for printing** is used: when a thread wants to print, it locks the mutex, prints, and then unlocks it. After a thread finishes baking, it **adds the number of baked breads back to the available space**, since the breads have been removed from the oven.

When a baker starts an order, the **start time** is recorded, and when the work is completed, the **end time** is recorded in the order struct of the customer. These times are then used to calculate the **mean**, **variance**, and **standard deviation**.



Multi-threaded baker with chaos.