Mohamed Magdy

Abstract

The project basically sorts the ready queue and calculates the turnaround and waiting times and their averages using these algorithms (number is the same as the input):

1. First-come, first-served.
2. Shortest-job-first (preemptive)
3. Shortest-job-first (non-preemptive)
4. Priority (preemptive)
5. Priority (non-preemptive)
6. Multilevel queue (first queue is SJF preemptive, second is FCFS)

The project is made in **Java**. Over **two** separate projects: A client project and another server project and they connect using a socket over *localhost:4444*.

How it works

- The server starts and accepts on "localhost" and an arbitrary port 4444.
- The client is started next and asks the user for all the details
    - Processes details are stored in a **2D ArrayList**.
- The 2D ArrayList is **Serialized** and sent over ObjectOutputStream to the server.
- Calculation is done on the server-side (each algorithm has its own class and calculation is mainly done in the constructors.)
- System.out of the server is set to output to the OutputStream (**so as to print the desired info on the client**)

Multilevel Queue

How I made it

Given that it's "multilevel queue", not "feedback", I used abstraction in order to use SJF preemptive and FCFS classes on each queue.

I split queues **according to priority**. A queue has processes with higher-than-average priorities, and the other less-priority ones are in the second queue.

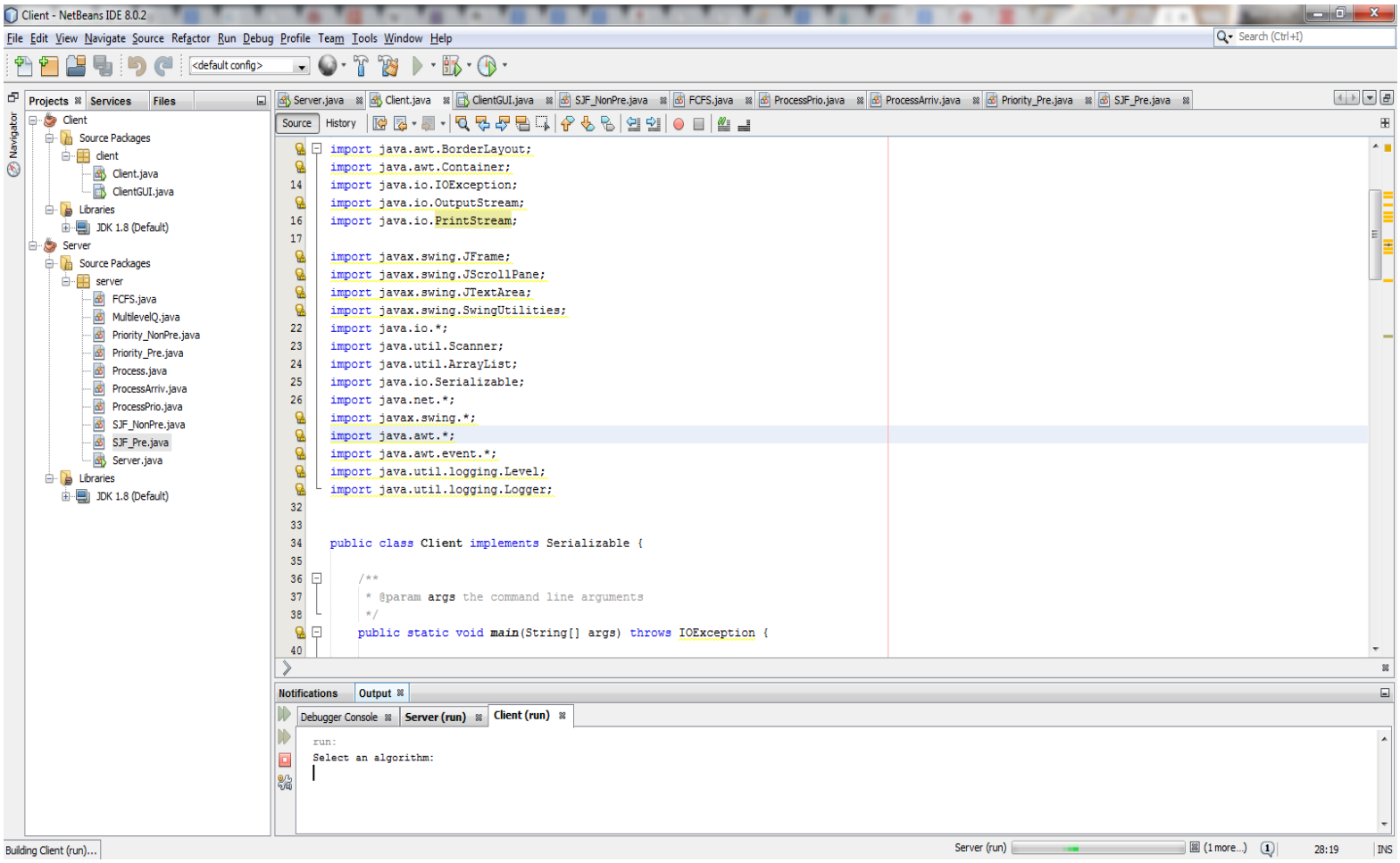
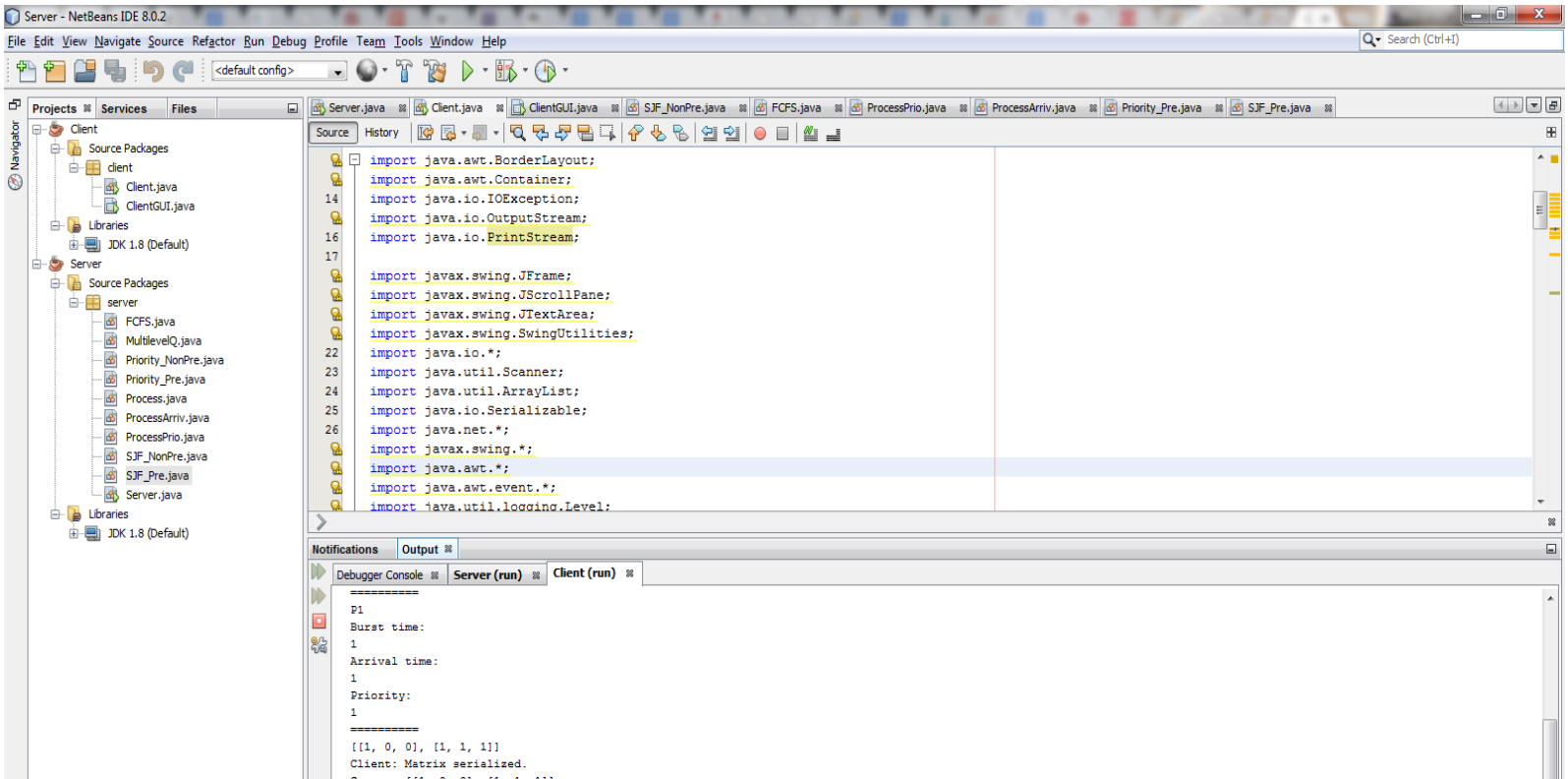


Figure 1 Input



Thank you.