**Program Description**

****

a) **Software used:** we used Windows 11, Eclipse as an IDE.

**Hardware used:** our personal laptops from MSI & HP & Huawei with i5 8th gen and ryzen5 4500u.

b) The project is divided into 3 threads:

1. **Reading thread**: Reads the file and fills the job queue and the list of the processes.

2. **Transporting thread**: Transporting the PCBs from job queue to ready queue.

3. **Processing Threads**: Takes processes from the ready queue and executes it.

4. **Other classes:** those classes are supportive and do not provide direct functionality to the user, e.g., Flag and Queue.

c) **Strength**: there is no race condition, easy to read and modify.

**Weakness**: we are moving the processes from the job to the ready queues based on the order, not the memory available.

**Test Runs**

صورة تحتوي على نص, الأحذية, شخص, تلبيس

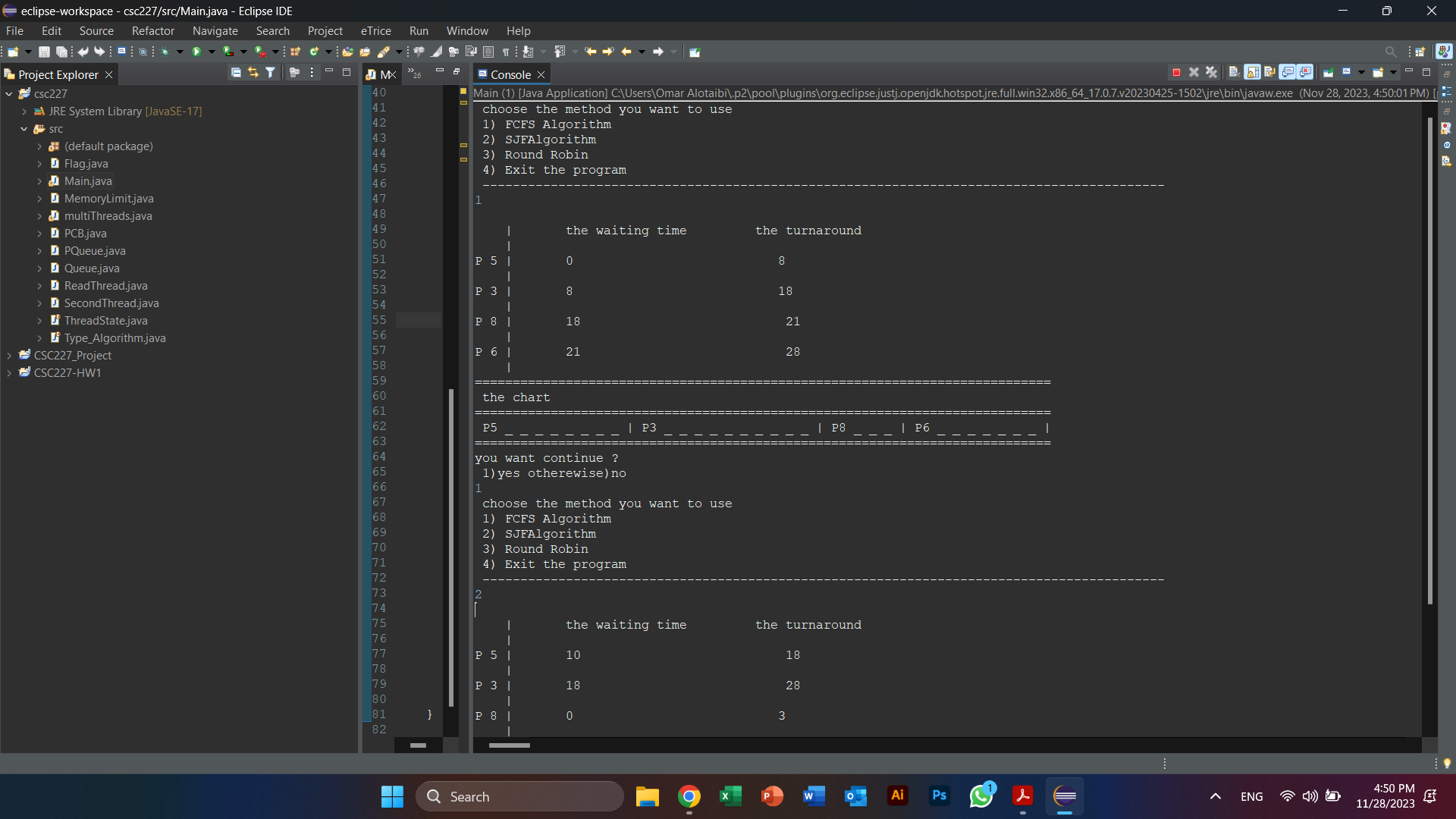
تم إنشاء الوصف تلقائياً

Test 1:

Job1: 5,8,800, Job2: 3,10,2000, Job3: 8,3,4000, Job4: 6,7,1000

Result:

FCFS

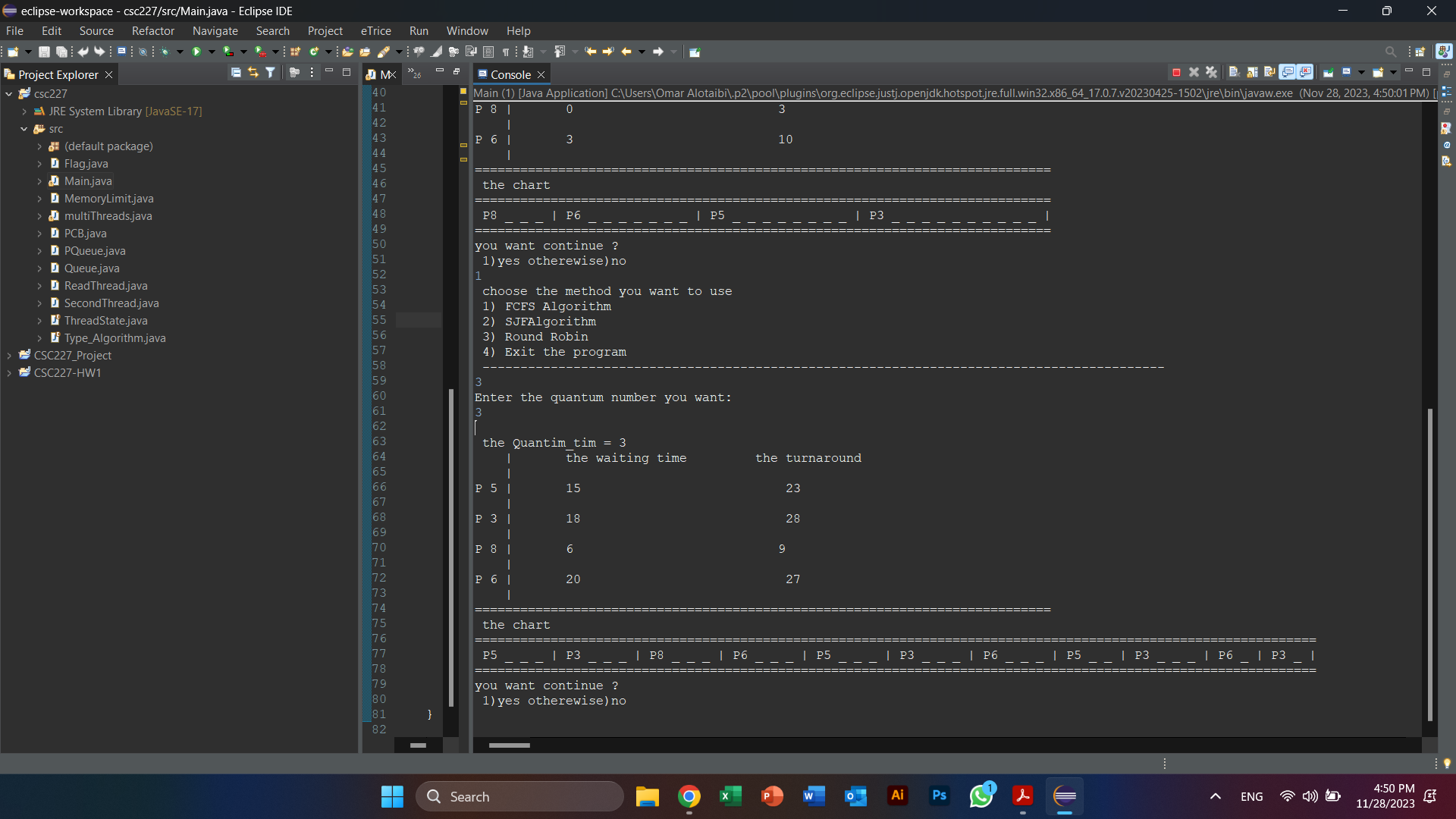


SJF

صورة تحتوي على نص, لقطة شاشة, برمجيات, برامج الوسائط المتعددة

تم إنشاء الوصف تلقائياً

RR(3)



RR(5)

صورة تحتوي على نص, لقطة شاشة, برمجيات, الحاسوب

تم إنشاء الوصف تلقائياً

TEST 2:

Job1: 7,5,100, Job2: 4,20,1000, Job3: 9,8,3000, Job4: 1,25,2000

Job5: 2,7,4000, Job6: 5,10,800

Results:

FCFS

Using FCFS Algorithm

| the waiting time the turnaround

|

P 7 | 0 5

|

P 4 | 5 25

|

P 9 | 25 33

|

P 1 | 33 58

|

P 2 | 58 65

|

P 5 | 65 75

|

============================================================================

the chart

====================================================================================================================================================================================

P7 \_ \_ \_ \_ \_ | P4 \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ | P9 \_ \_ \_ \_ \_ \_ \_ \_ | P1 \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ | P2 \_ \_ \_ \_ \_ \_ \_ | P5 \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ |

====================================================================================================================================================================================

you want continue ?

SJF

Using SJF Algorithm

| the waiting time the turnaround

|

P 7 | 0 5

|

P 4 | 30 50

|

P 9 | 5 13

|

P 1 | 50 75

|

P 2 | 13 20

|

P 5 | 20 30

|

============================================================================

the chart

====================================================================================================================================================================================

P7 \_ \_ \_ \_ \_ | P9 \_ \_ \_ \_ \_ \_ \_ \_ | P2 \_ \_ \_ \_ \_ \_ \_ | P5 \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ | P4 \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ | P1 \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ |

====================================================================================================================================================================================

RR(3)

Using RR Algorithm

the Quantim\_time = 3

| the waiting time the turnaround

|

P 7 | 9 14

|

P 4 | 48 68

|

P 9 | 26 34

|

P 1 | 50 75

|

P 2 | 40 47

|

P 5 | 47 57

|

============================================================================

the chart

==================================================================================================================================================================================================================================================================================================

P7 \_ \_ \_ | P4 \_ \_ \_ | P9 \_ \_ \_ | P1 \_ \_ \_ | P7 \_ \_ | P4 \_ \_ \_ | P9 \_ \_ \_ | P2 \_ \_ \_ | P5 \_ \_ \_ | P1 \_ \_ \_ | P4 \_ \_ \_ | P9 \_ \_ | P2 \_ \_ \_ | P5 \_ \_ \_ | P1 \_ \_ \_ | P4 \_ \_ \_ | P2 \_ | P5 \_ \_ \_ | P1 \_ \_ \_ | P4 \_ \_ \_ | P5 \_ | P1 \_ \_ \_ | P4 \_ \_ \_ | P1 \_ \_ \_ | P4 \_ \_ | P1 \_ \_ \_ | P1 \_ \_ \_ | P1 \_ |

==================================================================================================================================================================================================================================================================================================

RR(5)

Using RR Algorithm

the Quantim\_tim = 5

| the waiting time the turnaround

|

P 7 | 0 5

|

P 4 | 45 65

|

P 9 | 20 28

|

P 1 | 50 75

|

P 2 | 43 50

|

P 5 | 45 55

|

============================================================================

the chart

======================================================================================================================================================================================================================================

P7 \_ \_ \_ \_ \_ | P4 \_ \_ \_ \_ \_ | P9 \_ \_ \_ \_ \_ | P1 \_ \_ \_ \_ \_ | P4 \_ \_ \_ \_ \_ | P9 \_ \_ \_ | P2 \_ \_ \_ \_ \_ | P5 \_ \_ \_ \_ \_ | P1 \_ \_ \_ \_ \_ | P4 \_ \_ \_ \_ \_ | P2 \_ \_ | P5 \_ \_ \_ \_ \_ | P1 \_ \_ \_ \_ \_ | P4 \_ \_ \_ \_ \_ | P1 \_ \_ \_ \_ \_ | P1 \_ \_ \_ \_ \_ |

======================================================================================================================================================================================================================================

**Program Documentation**

**صورة تحتوي على نص, الحاسوب, حاسوب محمول, لقطة شاشة

تم إنشاء الوصف تلقائياً**

**Classes and Enums**

* **Flag:** A class with an object that works as a trigger and is used to communicate between classes.
* **MemoryLimit:** A class for the management of the limit of the memory and its 8192.
* **ProcessingThreads:** A class with a *thread* that takes processes from the ready queue and executes it.
* **PCB:** A class that contains all data of the processes.
* **PQueue:** A class for the priority queue data structure used in SJF.
* **Queue:** A class for the queue data structure used in FCFS & RR.
* **ReadThread:** Aclass that with a *thread* reads the file and fills the job queue and the list of the processes.
* **TransportingThread:** A class with a *thread* that transports the PCBs from job queue to ready queue.
* **ThreadState:** An Enum that displays the state of the PCBs from creation to termination.
* **Type\_algorithms:** AnEnum to choose which algorithm to apply.