

**G52OSC Coursework**

**20 Dec 2017**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of members** | | **Course Name** | **Student ID** |
| Oh Lean Kai | | Computer science with AI | 025344 |
| Lin Jiaqi | | Computer science with AI | 025443 |

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **No.** | **Content** | **Page** |
| **1.** | **First Come First Serve** | **2** |
| **2.** | **Shortest Job First (SJF) Preemptive Version** | **4** |
| **3.** | **Round Robin** | **6** |
| **4.** | **Priority, with higher priority jobs preempting a lower priority running job** | **8** |
| **5.** | **Sample Input/Output With Same Workload** | **10** |
| **7.** | **Conclusion** | **16** |

**1.First Come First Serve**

Implementation:

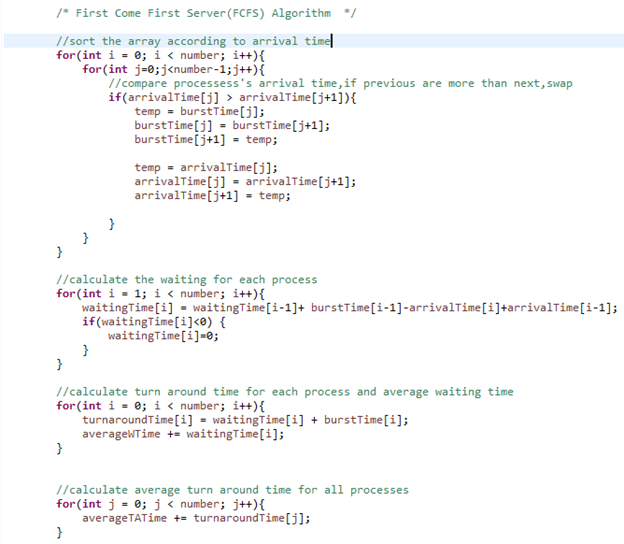
****

Figure1.1

Sample Input/Output:

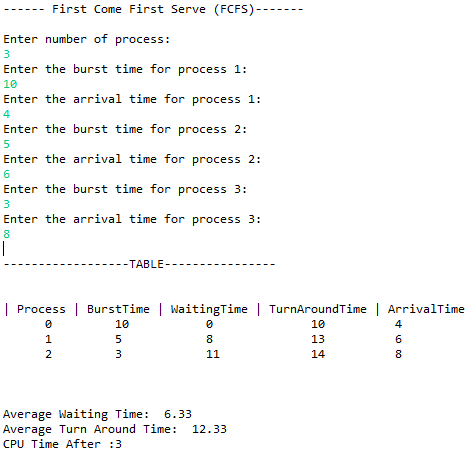
****

Figure1.2

Explanation:

First Come First Serve process scheduling is the simplest process scheduling algorithm.The implementation is based on First Come First Serve (FIFO) basis.By sorting processes according to their arrival time would be able to implement this algorithm. The advantage of FCFS algorithm is that doesn't include any complex logic, it just puts the process requests in a queue and executes it one by one but this scheduling method is non-preemptive. Because of non-preemptive scheduling, short processes which only arrived later on will have to wait till all the previous processes complete. After several testing with different sets of data, this algorithm gives a reasonably high waiting time.The advantage of FCFS algorithm is that doesn't include any complex logic, it just puts the process requests in a queue and executes it one by one

**2.Shortest Job First (SJF) Preemptive Version**

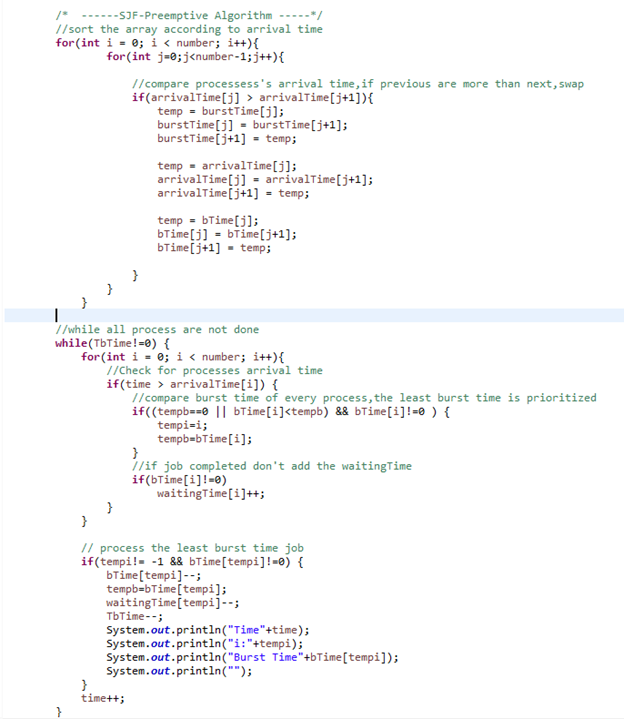
****

Figure2.1

Sample Input/Output:

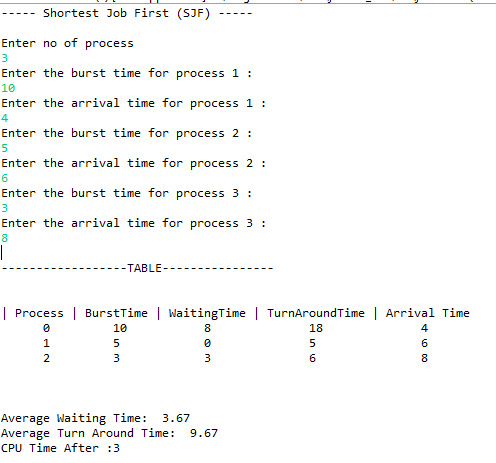
****

Figure2.2

Explanation:

Shortest job First process scheduling(SJF) also known as Shortest Job Next(SJN) always favor the waiting process with the smallest execution time to execute next.It is a Greedy Algorithm which will cause starvation (means the longer process will take a lot more waiting time to process) if shorter processes keep coming in but it outbeats with having the minimum average waiting time among all scheduling algorithms. The above implementation is the preemptive version of SJF. Every second, it will constantly check for the shortest job available to processes it.

**3.Round Robin**

Implementation:

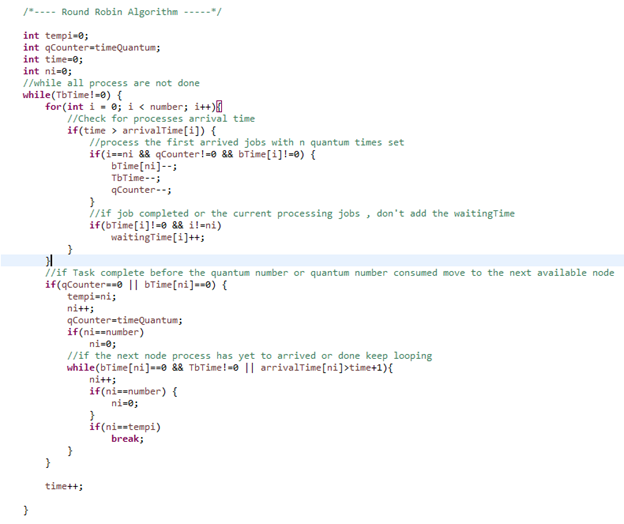


Figure3.1

Sample Input/Output:

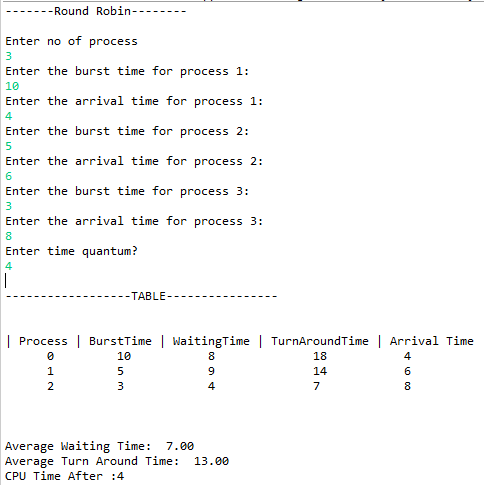


Figure3.2

Explanation:

Time slices also known as time quantum in round robin which assigned equal portion of time to all processes without priority in a circular order. It is a fair algorithm but the throughput in RR largely depends on the choice of the length of the time quantum. If time quantum is longer than needed, it tends to exhibit the same behavior as FCFS and if the time quantum is set too short, it will result in the least efficient algorithm. In short, the efficiency of this algorithm heavily rely on the time quantum preset by the user.

**4.Priority, with higher priority jobs preempting a lower priority running job**

Implementation:

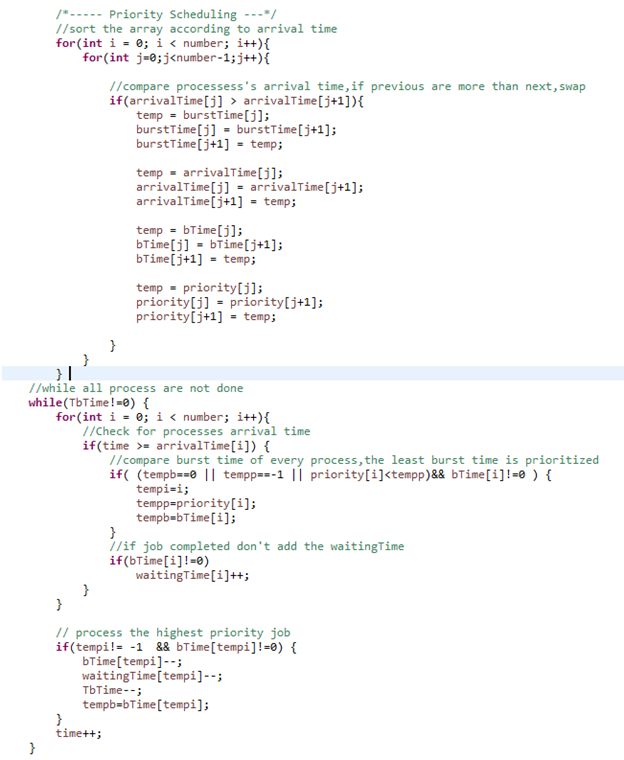
****

Figure4.1

Sample Input/Output:

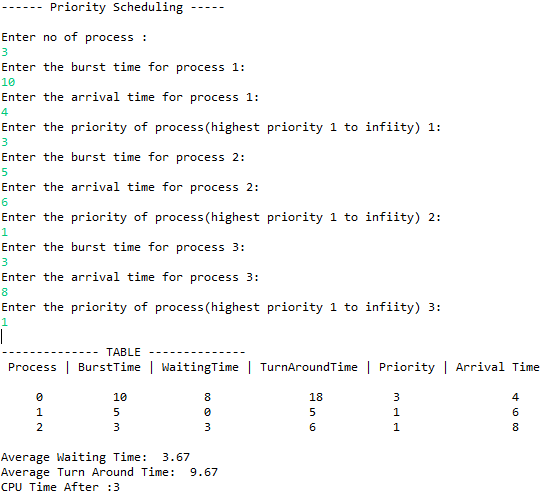
****

Figure4.2

Explanation:

Priority scheduling is one of the most common scheduling algorithms used in CPU process scheduling. Each process is assigned first according to their arrival time, if two processes have the same arrival time , then processes with highest priority first. If the processes with the same priority arrived while current process is not completed, then continue with the current process.

**5.Sample Input/Output with different type of Workloads**

**1.Input set 1**

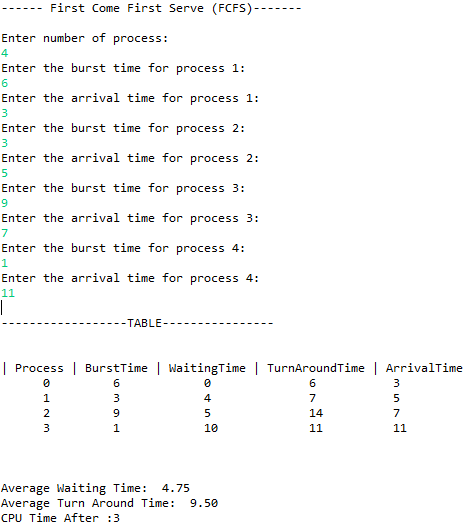
****

Figure5.1.1

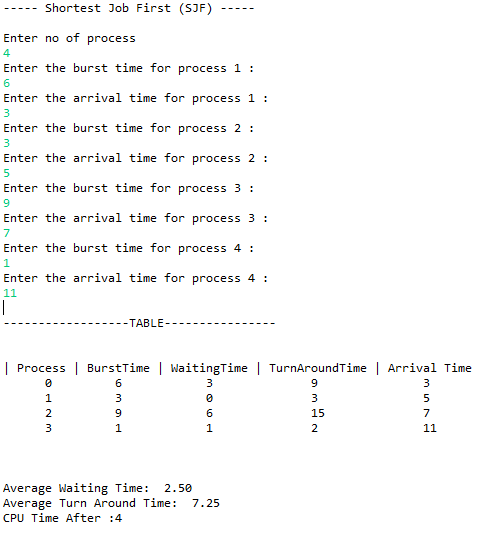
****

Figure5.1.2

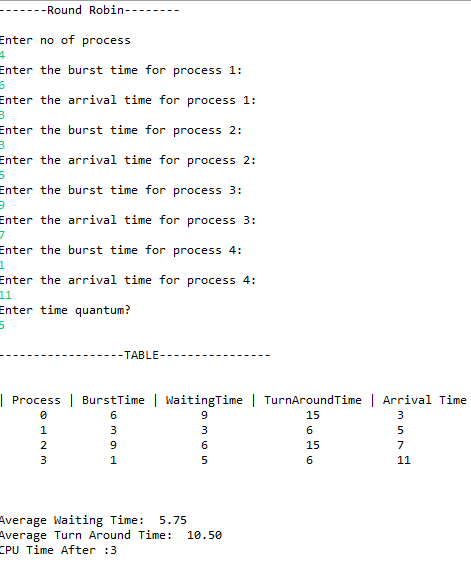


Figure5.1.3

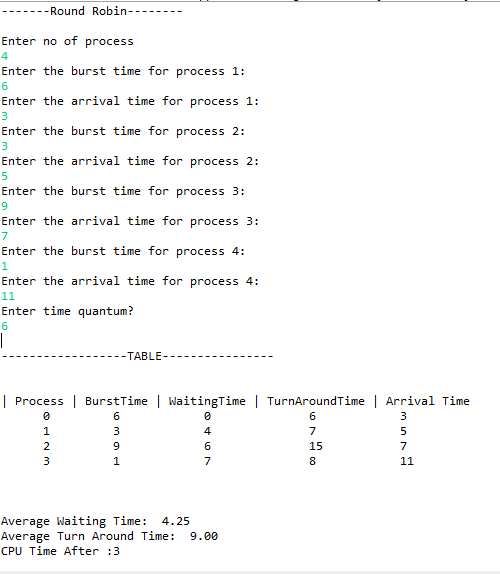
****

Figure5.1.4

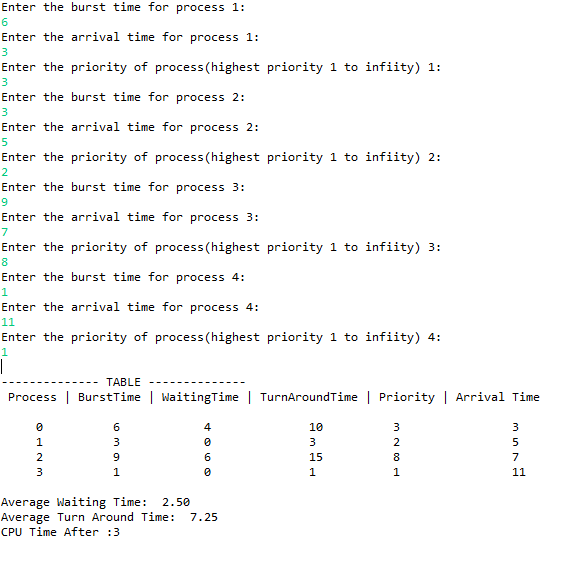
****

Figure5.1.5

**2.Input set 2**

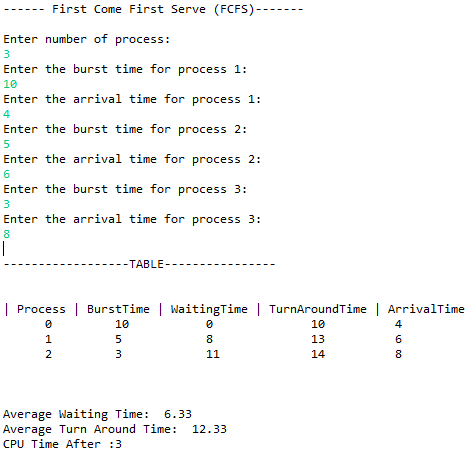
****

Figure5.2.1

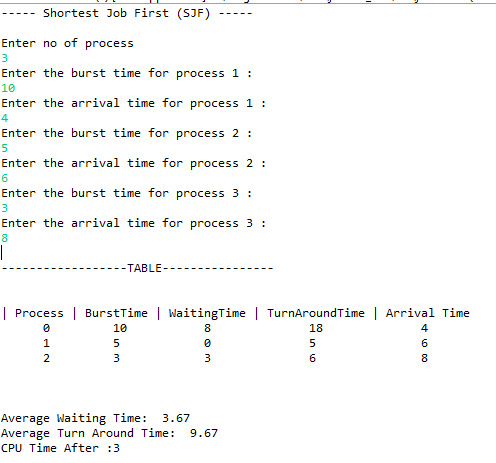
****

Figure5.2.2

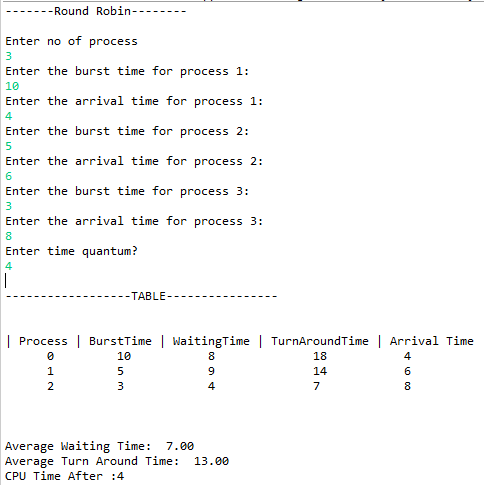


Figure5.2.3

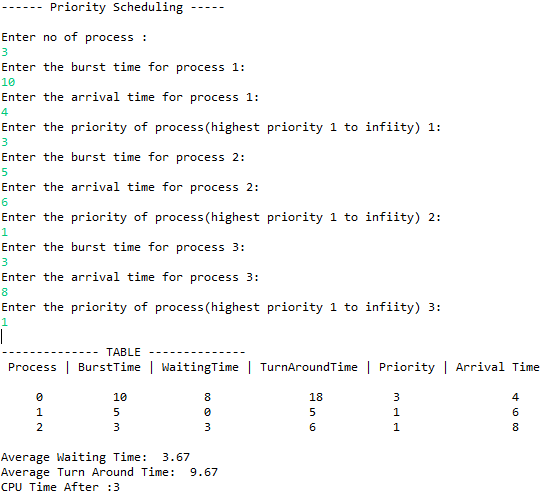
****

Figure5.2.4

Input set 1:

Processes={1,2,3,4}

Burst Time={6,3,9,1}

Arrival Time={3,5,7,11}

Input set 2:

Processes={1,2,3}

Burst Time={10,5,3}

Arrival Time={4,6,8}

With the above two sets of data given to all 4 algorithm , it clearly shows that SJF Preemptive Version and Priority Scheduling yield the shortest average waiting time with 2.5 and 3.67 respectively and turnaround time with 7.25 and 9.00 respectively. The CPU Usage for all algorithm are the same as.shown in the CPU Time,3.

In Figure 5.1.3 and Figure 5.1.4 are round robin algorithm with the same input set 1 but different quantum number. Round Robin can be a fair algorithm but the efficiency of the algorithm heavily rely on the time quantum. A right choice of time quantum is very important. If time quantum is longer than needed, it tends to exhibit the same behavior as FCFS. if the time quantum is set too short, it will yield a very high waiting and turnaround time.

**6.Conclusion**

In short, Shortest Job First ( SJF) Preemptive Version and Priority Scheduling can be equally suitable to executive any given set of processes with the shortest waiting and turnaround time. However,in SJF ,the longest burst time process will have to deal with the most waiting time if a shorter process keep coming and in real processor, certain processes will have higher priority over the others. Therefore, Priority Scheduling is the most suitable algorithm among all.

**MARKING SCHEME**

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name & No: | (1)Oh Lean Kai  (2)Lin Jiaqi | | |
| Group No: | NO.18 | | |
| Module Title & Code: | **Operating Systems and Concurrency (G52OSC)** | | |
| Attributes | | Marks Allotted | Marks Awarded |
| 1. Software Part: (30%) | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
| 2. Report Part: (20%) | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
|  | |  |  |
| **Total** | | 50 |  |
|  |  |  |  |

Note: Include this marking scheme page in the last page of your report.