Operating System I
Assignment #3

CPU Scheduler Simulator

Scheduling is a fundamental operating-system function. Almost all computer resources are scheduled before use. The CPU is, of course, one of the primary computer resources. Thus, its scheduling is central to operating-system design. CPU scheduling determines which processes run when there are multiple run-able processes. CPU scheduling is important because it can have a big effect on resource utilization and the overall performance of the system.

Submission Rules:

1. **Submitted file name should follow this format ID1_ID2_ID3_ID4_ID5_ID6_Group.zip**
2. **Submit a zip archive not a RAR**
3. **Groups will be a minimum of 4 and a maximum of 6 (Preferably from the same lab!)**
4. **Submissions will be through Blackboard**
5. **No late submissions are allowed this time**
6. **Cheating is prohibited and will be tolerated (You will be given Negative grades when caught)**

Write a java program to simulate the following schedulers:

1. **Non-Preemptive** Priority Scheduling with context switching (Must solve the starvation problem however any solution is acceptable)
2. **Non-Preemptive** Shortest- Job First (SJF) (Must solve the starvation problem however any solution is acceptable)
3. Shortest-Remaining Time First (SRTF) Scheduling with context switching (Must solve the starvation problem however any solution is acceptable)
4. AGAT Scheduling :
    a. The Round Robin (RR) CPU scheduling algorithm is a fair scheduling algorithm that gives equal time quantum to all processes So All processes are provided a static time to execute called quantum**, but in our AGAT scheduling each process will have a different Quantum**.
    b. A new factor is suggested to attach with each submitted process in our AGAT scheduling algorithm. This factor based on (priority, arrival time and remaining service time). The equation summarizes this relation is:

        - Set V1 as (if **last-arrival-time** > 10 then (**last-arrival-time(all processes) /10) else 1)**
        - Set V2 as (if **max-remaining burst time** > 10 then (**max-remaining burst time(all processes) /10) else 1)**

AGAT-Factor = (10-Priority) + ceiling(Arrival Time/v1) + ceiling(Remaining Burst Time/v2)

c. Once a process is executed for given time period, it's called **Non-preemptive AGAT** till the finishing of (round(40% of quantum time)), after that it's converted to **Pre-emptive AGAT after which it can be replaced with the process with the best (least) AGAT factor if any.**

d. We have 3 scenarios for a running process

   i. The running process used all its quantum time and it still has job to do (add this process to the end of the **queue**, then increases its quantum time by 2). Next process is picked from queue.

   ii. The running process didn't use all its quantum time because it was removed in favor of a process with better AGAT factor (add this process to the end of the **queue**, then increases its quantum time by the remaining quantum time for it).

   iii. The running process finished its job (set its quantum time to **zero and remove it from ready queue and add it to the dead list**).

Example of AGAT Schedule:

Processes	Burst time	Arrival time	Priority	Quantum
P1	17	0	4	4
P2	6	3	9	3
P3	10	4	3	5
P4	4	29	8	2

Answer:

P1	P2	P1	P2	P3	P1	P3	P1	P3	P4	P3	P4

0 3 6 8 11 13 20 23 28 32 33 34 3

Processes	**Burst time**	**Arrival time**	**Priority**	**Quantum**	**V1**	**Ceil(Arrival time/V1)**
P1	**17**	**0**	**4**	**4**	**2.9**	**0**
P2	**6**	**3**	**9**	**3**	**2.9**	**2**
P3	**10**	**4**	**3**	**5**	**2.9**	**2**
P4	**4**	**29**	**10**	**2**	**2.9**	**10**

- Quantum (4, 3, 5,2) -> round(40%) = (2,-,-,-) P1 Running
    - V2 = (17/10)=1.7
    - Factor = ((6+0+ceil(17/1.7))=**16** , ceil(1+2+ceil(6/1.7))=**7** , **15** , **13**)
- Quantum (4+1,3,5,2) -> round(40%) = (-,1,-,-) P2 Running
    - V2 = (14/10)=1.4
    - Factor = ((6+0+ceil(14/1.4))=**16** , ceil(1+2+ceil(6/1.4))=**8** , **17** , **13**)
- Quantum (5,3+2,5 ,2) -> round (40%) = (2,-,-,-) P1 Running
    - V2 = (14/10)=1.4
    - Factor = ((6+0+ceil(14/1.4))=**16** , ceil(1+2+ceil(3/1.4))=**6** , **17** , **13**)
- Quantum (5+2,5,5 ,2) -> round (40%) = (-,2,-,-) P2 Running
    - V2 = (12/10)=1.2
    - Factor = ((6+0+ceil(12/1.2))=**16** , **6** , **18** , **14**)
- Quantum (7,0,5 ,2) -> round (40%) = (-,-,2,-) P3 Running
    - V2 = (12/10)=1.2
    - Factor = ((6+0+ceil(12/1.2))=**16** , **-** , **18** , **14**)
- Quantum (7,0,5+3 ,2) -> round (40%) = (3,-,-,-) P1 Running
    - V2 = (12/10)=1.2
    - Factor = ((6+0+ceil(12/1.2))=**16** , **-** , **18** , **14**)

- Quantum (7+2,0,8 ,2) -> round (40%) = (-,-,3,-) P3 Running
    - V2 = 1 -> max burst time p3 = 8
    - Factor = ((6+0+ceil(5/1))=**11** , **-** , **17** , **14**)
- Quantum (9,0,8+2 ,2) -> round (40%) = (4,-,-,-) P1 Running
    - V2 = 1 -> max burst time p1,3 = 5
    - Factor = ((6+0+ceil(5/1))=**11** , **-** , **15** , **14**)
- Quantum (0,0,10 ,2) -> round (40%) = (-,-,4,-) P3 Running
    - V2 = 1 -> max burst time p3 = 5
    - Factor = (**-**, **-** , **15** , **14**)

- Quantum (0,0,10+6 ,2) -> round (40%) = (-,-,-,1) P4 Running
  - V2 = 1 -> max burst time p4 = 4
  - Factor = (-, **-** , **10** , **14**)
- Quantum (0,0,14 ,2+1) -> round (40%) = (-,-,6,-) P3 Running
  - V2 = 1 -> max burst time p4 = 3
  - Factor = (-, **-** , **10** , **13**)
- Quantum (0,0,0 ,3) -> round (40%) = (-,-,-,1) P4 Running
  - V2 = 1 -> max burst time p4 = 3
  - Factor = (-, **-** , **-** , **13**)

Program Input

- Number of processes
- Round Robin Time Quantum
- context switching

For Each Process you need to receive the following parameters from the user:

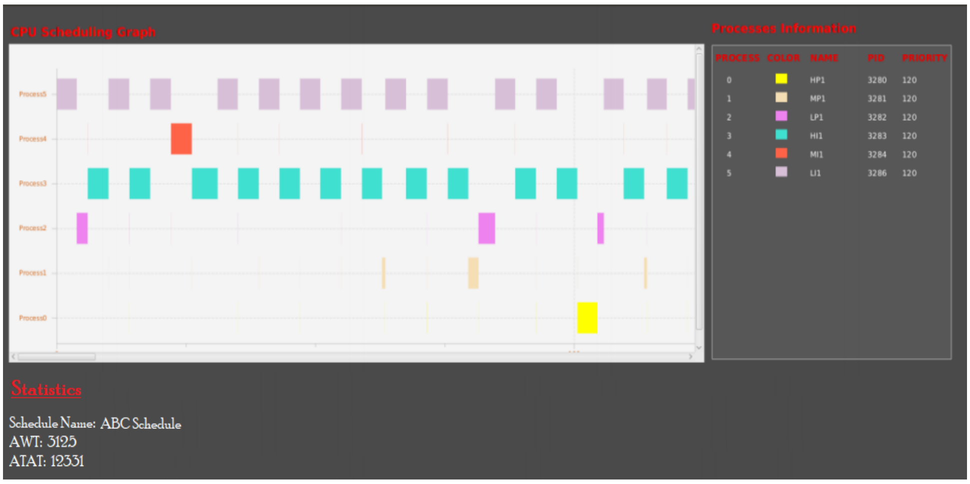
- Process Name
- Process Color(Graphical Representation)
- Process Arrival Time
- Process Burst Time
- Process Priority Number

Program Output

For each scheduler output the following:

- Processes execution order
- Waiting Time for each process
- Turnaround Time for each process
- Average Waiting Time
- Average Turnaround Time
- Print all history update of quantum time for each process (**AGAT Scheduling**)
- Print all history update of AGAT factor for each process (**AGAT Scheduling**)

<u>GUI Example</u>



Grading Criteria

Priority Scheduling	**1**
SJF Scheduling	**1**
SRTF Scheduling	**1**
AGAT Scheduling	**4**
GUI	**1**