SCHEDULING ALGORITHM

PRIORITY SCHEDULING

SCHEDULING ALGORITHM

ACTIVITY 4 PRIORITY ALGORITHM

PRIORITY ALGORITHM

Priority Scheduling is a method of scheduling processes that is based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority.

- The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis. Priority depends upon memory requirements, time requirements, etc.

Types of Priority Scheduling

Priority scheduling divided into two main types:

Preemptive Scheduling

- In Preemptive Scheduling, the tasks are mostly assigned with their priorities. Sometimes it is important to run a task with a higher priority before another lower priority task, even if the lower priority task is still running. The lower priority task holds for some time and resumes when the higher priority task finishes its execution.

PRIORITY ALGORITHM

Non-Preemptive Scheduling

In this type of scheduling method, the CPU has been allocated to a specific process. The process that keeps the CPU busy, will release the CPU either by switching context or terminating. It is the only method that can be used for various hardware platforms. That’s because it doesn’t need special hardware (for example, a timer) like preemptive scheduling.

Characteristics of Priority Scheduling

- A CPU algorithm that schedules processes based on priority.
- It used in Operating systems for performing batch processes.
- If two jobs having the same priority are READY, it works on a FIRST COME, FIRST SERVED basis.
- In priority scheduling, a number is assigned to each process that indicates its priority level.
- **Lower the number, higher is the priority.**
- In this type of scheduling algorithm, if a newer process arrives, that is having a higher priority than the currently running process, then the currently running process is preempted.

PRIORITY ALGORITHM

Consider following five processes P1 to P5. Each process has its unique priority, burst time, and arrival time.

PROCESS	AT	BT	PRIORITY
P1	0	11	2
P2	5	28	0
P3	12	2	3
P4	2	10	1
P5	9	16	4

Queueing:

Time 0: **P1** is executed for 11 sec.

Time 2: **P4** HP 1 | **P1** executed 2ms | **P1** BT is 11-2 = **9**ms

P1 BT new value **9ms**

Time 5: **P2** executed with HP 0 2 -> 5 = **3ms completed**

P4 BT 10-3 = 7ms (P4 PREEMPT)

P3 & **P5** arrived

P3 priority of 3 checking HP

P5 priority of 4 checking HP

P2 priority of 0 HP no preemption

P2 executed 28ms BT new value 0 finished

Time 33: **P4** executed 7ms BT new value 0 finished

Time 40: **P1** executed 9ms BT value 0 finished

Time 49: **P3** executed 2ms BT value 0 finished

Time 51: **P5** executed 16ms BT value **0 finished**

PRIORITY ALGORITHM

Gantt Chart

Time Units → **0** **2** **5** **33** **40** **49** **51** **67**

Process → P1 P4 P2 P4 P1 P3 P5

Waiting Time = Total waiting time – No. of ms process exec – AT

P1 = 40 – 2 – 0 = 38ms

P2 = 5 – 0 – 5 = 0ms

P3 = 49 – 0 – 12 = 37ms

P4 = 33 – 5 - 2 = 28ms

P5 = 51 – 0 – 9 = 42ms

ACTIVITY 4

PRIORITY

PRIORITY ALGORITHM

INSTRUCTIONS

1. Make a Java program that will compute for the following.

a. Turn Around Time = Completion Time – Arrival Time.

b. Waiting Time = Turn Around Time – Burst Time

c. Start Time = CT - BT

d. Average Turn Around Time.

e. Average Waiting Time.

f. Gannt Chart.

2. The user will have the choice to input 3 to 5 process, including Arrival Time and Burst time.

3. The program will ask the user if he/she wants to try again?

4. Upload the java file using the filename format: SNFN4.java

5. Upload a sample output of the program.

6. Java file and sample output will be uploaded at teams.

Criteria: 50 points

1. Complete and running program no errors. **50 points**.
2. Incomplete but running program no errors. **35 points**.
3. Running but no correct output **15 points**. (for the effort)
4. 2nd checking must be complete and working no errors **20 points**.
5. Programs to be checked should be error free. No checking of program if error persists.
6. Once I checked the program it is recorded. If you request for 2nd checking you be in the last priority.

PROGRAM OUTPUT

ACTIVITY 4: PRIORITY ALGORITHM

ACTIVITY 4: FCFS ALGORITHM

Title: **PRIORITY SCHEDULING ALGORITHM**

Enter no. of process: (IN)

Process	AT	BT	PRIORITY	CT	TAT	WT
P1	(IN)	(IN)	(IN)	(O)	(O)	(O)
P2	(IN)	(IN)	(IN)	(O)	(O)	(O)
P3	(IN)	(IN)	(IN)	(O)	(O)	(O)
P4	(IN)	(IN)	(IN)	(O)	(O)	(O)
P5	(IN)	(IN)	(IN)	(O)	(O)	(O)

Average TAT: (O)
Average WT: (O)

Gannt Chart:
(Example)

```
| P1 | P2 | P3 | P4 | P5 |
0    4    7    8    10   14
```

EOA

SAMPLE OUTPUT FOR CHECKING
PURPOSE ONLY

Activity 4: Priority Algorithm

Enter no. of process: (IN)

Process	AT	BT	PRIORITY	CT	TAT	WT
P1	0	11	2	49	49	38
P2	5	28	0	33	28	0
P3	12	2	3	51	39	37
P4	2	10	1	40	38	28
P5	9	16	4	67	58	42

AVERAGE TAT = 42ms
AVERAGE WT = 29ms

Gannt Chart: | P1 | P4 | P2 | P4 | P1 | P3 | P5 |
(Example) 0 2 5 33 40 49 51 67