

Operating System

Report Assignment Simulation Based

Programming Problem 11 Operating System By Galvin 9th Edition

11. Design a scheduler that uses a preemptive priority scheduling algorithm based on dynamically changing priority. Larger number for priority indicates higher priority.

Assume that the following processes with arrival time and service time wants to execute(for reference):

ProcessID	ArrivalTime	ServiceTime
P1	0	4
P2	1	1
P3	2	2
P4	3	1

When the process starts execution(i.e. CPU assigned),priority for that process changes at the rate of $m=1$.When the process waits for CPU in the ready queue(but not yet starts execution), its priority changes at rate $n=2$. All the processes are initially assigned priority value of 0 when they enter ready queue for the first time.The time slice for each process is $q=1$. When two processes want to join ready queue simulataneously, the process which has not executed recently is given priority. Calculate the average

waiting time for each process. The program must be generic i.e. number of processes, their burst time and arrival time must be entered by user.

Student Name: Sakshi Priya

Student ID: 11609970

Section No: K1655

Roll No.: A20

Email Address: sakshi.priya@outlook.com

GitHub Link: <https://github.com/SakshiPrRoy/OS-Assignment>

Priority Scheduling

Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithm. Each process is assigned first arrival time (less arrival time process first) if two processes have same arrival time, then compare to priorities (higher priority first). Also, if two processes have same priority then compare to process number (less process number first). This process is repeated while all process get executed.

Implementation--

1. First input the processes with their arrival time, burst time and priority.
2. Sort the processes, according to arrival time if two process arrival time is same then sort according process priority if two process priority are same then sort according to process number.
3. Now simply apply FCFS algorithm.

Compile and Run:

Use following command to compile program -
gcc os.c -o os

```
sakshi@mah: ~/Desktop/assignment
sakshi@mah:~/Desktop/assignment$ gcc os.c -o os
```

Use following command to compile program-
./os

```
sakshi@mah: ~/Desktop/assignment
sakshi@mah:~/Desktop/assignment$ ./os
```

Test Cases:

General ShortCuts used:

```
*****
                          ShortCuts Used In Table
*****
Process_name  -----> PName
~~~~~
Arrival_Time  -----> Atime
~~~~~
Execution_Time -----> ETime
~~~~~
Priority       -----> Priority
~~~~~
Waiting_Time  -----> WTime
~~~~~
Total_Turn_Around_Time -----> TTTime
*****
```

Case 1: When arrival time is zero but priority is different.

```
sakshi@mahi:~/Desktop/assignment$ gcc os.c -o os
sakshi@mahi:~/Desktop/assignment$ ./os
Enter the number of process:4
Please Enter Process Name[1]:P1
Please enter Arival Time:
0
Please Enter Service/Execution Time:
10
Please Enter Priority:
0
Please Enter Process Name[2]:P2
Please enter Arival Time:
0
Please Enter Service/Execution Time:
1
Please Enter Priority:
0
Please Enter Process Name[3]:P3
Please enter Arival Time:
0
Please Enter Service/Execution Time:
2
Please Enter Priority:
0
Please Enter Process Name[4]:P4
Please enter Arival Time:
0
Please Enter Service/Execution Time:
5
Please Enter Priority:
0
```

PName	ATime	ETime	Priority	WTime	TTTime
P2	0	1	1	0	1
P4	0	1	2	1	2
P1	0	10	3	2	12
P3	0	2	4	12	14

[illegible]

Average waiting time is:3.750000

~~~~~

Average turnaroundtime is:7.250000

A decorative horizontal separator consisting of approximately 60 small blue upward-pointing triangles.

**Case 2:** When arrival time is different and priority is also different.

```
sakshi@mahimahi:~/Desktop/assignment$ gcc os.c -o os
sakshi@mahimahi:~/Desktop/assignment$ ./os
Enter the number of process:4
Please Enter Process Name[1]:P1
Please enter Arival Time:
1
Please Enter Service/Execution Time:
5
Please Enter Priority:
1
Please Enter Process Name[2]:P2
Please enter Arival Time:
3
Please Enter Service/Execution Time:
10
Please Enter Priority:
2
Please Enter Process Name[3]:P3
Please enter Arival Time:
4
Please Enter Service/Execution Time:
7
Please Enter Priority:
4
Please Enter Process Name[4]:P4
Please enter Arival Time:
6
Please Enter Service/Execution Time:
8
Please Enter Priority:
3
```

```
*****
PName      ATime      ETime      Priority      WTime      TTTime
P1          1          5          1            0           5
P2          3         10          2            3          13
P4          6          8          3           10          18
P3          4          7          4           20          27
*****

^.....^

Average waiting time is:8.250000

^.....^

Average turnaroundtime is:15.750000

^.....^
```

### Case 3: When arrival and priority both are zero.

```
sakshi@mahi:~/Desktop/assignment$ gcc os.c -o os
sakshi@mahi:~/Desktop/assignment$ ./os
Enter the number of process:4
Please Enter Process Name[1]:P1
Please enter Arival Time:
0
Please Enter Service/Execution Time:
10
Please Enter Priority:
3
Please Enter Process Name[2]:P2
Please enter Arival Time:
0
Please Enter Service/Execution Time:
1
Please Enter Priority:
1
Please Enter Process Name[3]:P3
Please enter Arival Time:
0
Please Enter Service/Execution Time:
2
Please Enter Priority:
4
Please Enter Process Name[4]:P4
Please enter Arival Time:
0
Please Enter Service/Execution Time:
1
Please Enter Priority:
2
```

```
*****
PName      ATime      ETime      Priority      WTime      TTTime
P1          0          10          0             0           10
P2          0          1          0            10           11
P3          0          2          0            11           13
P4          0          5          0            13           18
*****
```

Average waiting time is:8.500000

Average turnaroundtime is:13.000000

**GitHub Link : <https://github.com/SakshiPrRoy/OS-Assignment>**