East West University

Assignment

Operating System

CSE-325

CPU Scheduling

Submitted by-

Name: Md Saiful

ID No: 2019-2-60-040

Section: 05

Submitted to-

Yeasir Rayhan

Lecturer

Department of CSE.

East West University

Submission Date: 2 - May -2021

Ans To the Question No :1

Starvation problem is actually a disadvantage. In case if an algorithm has a starvation problem, then that we say that algorithm has a disadvantage. Whereas in case if we say that an algorithm is free from starvation then we can be considered as an algorithm there as an advantage of algorithm.

Starvation: An algorithm suffers from starvation problem if and only if there is a chance for a process in the ready state to wait indefinitely to get the CPU.

First of all, priority-based scheduling algorithm will suffer from the problem of starvation.

Algorithms	Starvation
FCFS	Yes
SJF	Yes
SRTF	Yes
RR	No
Priority	Yes
Multi-level Queue	Yes
Multi-level FQ	No
O (n) Linux	Yes
O (1) Linux	No

Fast come first served scheduling Algorithms

1. The process which has the least arrival time will be scheduled first.
2. It is a non-permeative scheduling algorithm.
3. It is not a priority-based scheduling algorithm.

No: P	AT	BT
P1	0	50
P2	1	60
P3	2	10
P4	3	20
P5	4	30

Gantt chart:

P1	P2	P3	P4	P5
P1	P2	P3	P4	P5

P2	P3	P4	P5	
P3	P4	P5		
P4	P5			
P5				

0 50 110 120 140 170

From Gantt chart after P1 and P2 execute then P3 will be get CPU so here P3 and P4 and P5 starve to get CPU. So, if in the ready queue so many longest bust time processes come first then though other process like P5 and P4 come to the ready queue for run but still they are wait for get CPU it's like convey effect.

Shortest Job Fast

1. Among the arrival process, process with the least bust time (execution time) will be given preference
2. It is a non-primitive scheduling algorithm and also it is a priority-based algorithm.
3. The process with the shortest bust time will be given higher priority rate whereas the process with the longest bust time will be given less then priority.

No: P	AT	BT
P1	2	3
P2	3	2
P3	4	5
P4	6	1
P5	8	2

Gantt Chart:

P1	P2	P4	P5	P3
P1	P2	P3	P3	P3
	P3	P4	P5	

2 5 7 8 10 15

So here, low priority process is actually starving which means it is waiting indefinitely because some higher priority process is created.so it is always possibility to lower priority process to starve. Here P3 starve for long time.

Shortest Remaining Job Fast

1. Among the arrival process, process with the least bust time (execution time) will be given preference.
2. It is a primitive scheduling algorithm and also it is a priority-based algorithm.
3. The process with the shortest bust time will be given higher priority rate whereas the process with the longest bust time will be given less then priority.

No: P	AT	BT
P1	0	12
P2	2	4
P3	3	6
P4	8	5

Gantt Chart:

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

0 2 3 6 8 12 17 27

So here, low priority process is actually starving which means it is waiting indefinitely because some higher priority process is created at the same time. So, it is always possibility to lower priority process to starve. Here P1 starve for long time.

Round Robin

1. Its works on the basis of a particular time quantum.
2. Each process gets a small unit of CPU time quantum or time slice.

No: P	AT	BT
P1	0	4
P2	1	5
P3	2	2
P4	3	1
P5	4	6
P6	5	3

First of all, round robin in not priority-based scheduling algorithm so definitely it is not suffering from the problem of starvation.

Now see the Gantt chart for round robin time quantum 3

Gantt Chart:

P1	P2	P3	P4	P1	P5	P6	P2	P5
P1	P2	P3	P4	P1	P5	P6	P2	P5
	P3	P4	P1	P5	P6	P2	P5	
	P4	P1	P5	P6	P2	P5		
	P1	P5	P6	P2				
		P6	P2					
		P2						

0 3 6 8 9 10 13 16 18 21

So, here starvation or convoy effect doesn't occur because for each process is given a fixed time to execute.no process is left behind.

Round Robin With Priority

If we adding priority then round robin scheduling get starvation because of lower-level priority process will be starve. Normally round robin scheduling algorithm doesn’t concern about priority scheduling.

Time Quantum 3:

No: P	AT	BT	priority
P1	0	4	3
P2	1	5	3
P3	2	2	5
P4	3	1	4
P5	4	6	3
P6	5	3	6

🟨 P1	🟨 P1	🟨 P2	🟨 P2	🟨 P5	🟨 P5	🟨 P4	🟨 P3	🟨 P6
🟩 P1	🟩 P2	🟩 P2	🟥 P3	🟥 P3	🟥 P3	🟥 P3	🟥 P3	🟩 P6
	🟥 P3	🟥 P3	P4	P4	P4	P6	P6	
	P4	P4	P5	P5	P5	P4		
	P1	P5	P6	🟥 P6	P6			
		🟥 P6						

0 3 4 7 9 12 15 16 18 21

So, here lower priority process P6 and P3 starving.

Multi-level Queue

No: P	AT	BT
P1	0	4
P2	1	5
P3	2	2
P4	3	1
P5	4	6
P6	5	3

Here Queue one is higher priority and Queue two is lower priority.

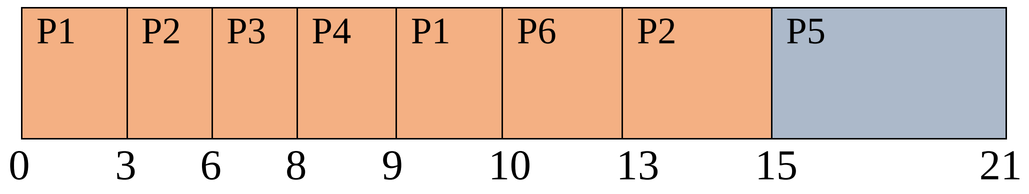
Queue one Round Robin time quantum 3:

P1	P2	P3	P4	P1	P6	P2
P1	P3	P4	P1	P6	P2	
	P4	P1	P6	P2		
	P1	P6	P2			
		P2				

Queue two FCFS:

P5			

Gantt Chart:



So, here we see that for lower priority Queue Process P5 did not get CPU because of Queue One not empty in the ready queue. So P5 get CPU after execution all process in the ready queue. So here P5 get starvation situation if Queue have to the process in the ready queue.

Multi-Level Feedback Queue

No: P	AT	BT
P1	0	4
P2	1	5
P3	2	2
P4	3	1
P5	4	6
P6	5	3

Here Queue one is higher priority and Queue two is lower priority.

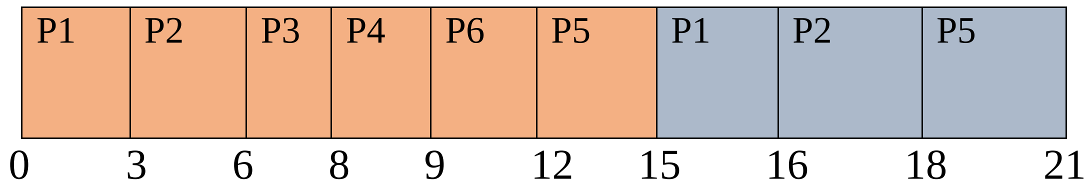
Queue one Round Robin time quantum 3:

P1	P2	P3	P4	P6	P5	
	P3	P4	P6	P5		
	P4	P6	P5			
		P5				

Queue two FCFS:

P1			
P2			
P5			

Gantt Chart:



So, here we see that lower priority process also get CPU. So, here starvation not occur because every process gets CPU deferent time slice and ramming process will be push to maintaining Queue FCFS and after Queue One empty then queue two process execute by first come first serve.

Linux (n) Scheduling Algorithm

Linux (n) Such starvation is exhibited by CPU-bound processes. The problem particularly starvation occurs when I/O-bound processes run in the system at the same time.

Linux (1) Scheduling Algorithms

Linux (1) scheduling use multi-level feedback Queue and every ready queue have different Queue so here starvation not occur. And all time slice interval for each process is set based on the dynamic priority. That’s way O(1) scheduler starvation not occur.

Ans To the Question No:2

Process	BT	AT	Priority
A	3	1	1
B	3	2	3
C	4	2	2
D	1	4	4
E	2	6	3
F	3	6	2

FCFS Gantt chart:

	A	C	B	D	F	E
	A	B	B	D	E	E
	B	C	D	E	F	
	C	D	E	F		
		E	F			
		F				

0 1 4 8 11 12 15 17

SJF Gantt chart:

	A	D	C	F	E	B
	A	B	B	B	B	B
	B	C	C	E	E	
	C	D	E	F		

0 1 4 5 9 12 14 17

Process	BT	AT	Priority
A	3	1	1
B	3	2	3
C	4	2	2
D	1	4	4
E	2	6	3
F	3	6	2

SRTF Gantt Chart:

	A	A	A	D	B	B	E	F	C
	A	B	B	B	B	B	C	C	C
		C	C	C	C	C	E	F	
		A	A	D		E	F		
						F			

0 1 2 3 4 5 6 8 10 13 17

Process	BT	AT	Priority
A	3	1	1
B	3	2	3
C	4	2	2
D	1	4	4
E	2	6	3
F	3	6	2

RR Gantt Chart:

	A	B	C	A	B	D	C	A	E	F	B	C	E	F	C	F
	A	B	C	A	B	D	C	A	E	F	B	C	E	F	C	
		C	A	B	D	C	A	E	F	B	C	E	F	C	F	
		A	B	D	C	A	E	F	B	C	E	F	C			
				C	A	E	F	B	C	E	F					
						F	B	C								
						B										

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

Priority With round robin

Process	BT	AT	Priority
A	3	1	1
B	3	2	3
C	4	2	2
D	1	4	4
E	2	6	3
F	3	6	2

Gantt Chart:

	A	A	A	C	C	E	E	C	C	F	F	F	B	B	B	D
	A	B	B	B	B	B	B	B	B	B	B	B	B	D	D	D
		C	C	C	D	D	D	D	D	D	D	D	D	B	B	
		A	A	D	C	E	F	F	F	F	F	F				
						F	C	C	C							
						C	E									

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

Multi-Level Queue

	Process	BT	AT	Priority
Q1	A	3	1	1
Q3	B	3	2	3
Q2	C	4	2	2
Q4	D	1	4	4
Q3	E	2	6	3
Q2	F	3	6	2

Q1:

A	A	A
A	A	A

Q2:

C	C	F	C	F	C	F
C	C	F	C	F	C	F
		C	F	C	F	

Q3:

B	E	B	E	B
B	E	B	E	B
	B	E	B	

Q4:

D	
D	

Gantt Chart:

A	A	A	C	C	F	C	F	C	F	B	E	B	E	B	D

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

Multi-Level Feedback Queue

Process	BT	AT	Priority
A	3	1	1
B	3	2	3
C	4	2	2
D	1	4	4
E	2	6	3
F	3	6	2

Q1 Highest priority timeslice 0.1:

A	B	C	D	E	F											

Q2 Middle priority timeslice 0.2:

A	B	C	D	E	F											

Q3 Lowest priority timeslice 1:

A	B	C	D	E	F	A	B	C	E	F	B	F	C
B	C	D	E	F	A	B	C	E	F	B	F	C	
C	D	E	F	A	B	C	E	F	B	C	C		
D	E	F	A	B	C	E	F	B	C				
E	F	A	B	C	E	F							
F	A	B	C										

Gantt Chart:

A	B	C	D	E	F	A	B	C	D	E	F	A	B	C	D	E	F	A	B	C	E	F	B	F	C

0 .1 .2 .3 .4 .5 .6 .8 1 1.2 1.4 1.6 1.8 2.8 3.8 4.8 5.5 6.7 7.8

Calculate the average wait and completion time for each scheduling algorithm:

Time	FCFS	SJF	SRTF	RR	Priority	MLQ
1	A	A	A	A	A	A
2	A	A	A	B	A	A
3	A	A	A	C	A	A
4	C	D	D	A	C	C
5	C	C	B	B	C	C
6	C	C	B	D	E	F
7	C	C	B	C	E	C
8	B	C	E	A	C	F
9	B	F	E	E	C	C
10	B	F	F	F	F	F
11	D	F	F	B	F	B
12	F	E	F	C	F	E
13	F	E	C	E	B	B
14	F	B	C	F	B	E
15	E	B	C	C	B	B
16	E	B	C	F	D	D
AWT	(0+6+2+ 6+9)/6 =5	(0+12+3+ 0+6+3)/6 =4	(0+3+11+ 0+2+4)/6 =3.33	(5+7+10+2 +6+10)/6 =6.67	(0+4+11+ 12+0+4)/6 =5.17	(0+4+12+ 11+7+2)/6 = 6
ACT	(3+10+7+ 11+16+14) /6 = 10.17	(3+4+8+ 11+13+16) /6 = 9.17	(3+7+16+ 4+9+12) /6 = 8.5	(8+11+15 +6+13+16) /6=11.5	(3+15+9+ 16+7+12) /6=10.33	(3+15+9 +16+7+ 12)/6=11.5