

# CSI3131 – Operating Systems

## Tutorial 4 – CPU Scheduling - Solution

1. Define the difference between preemptive and cooperative scheduling.

**Preemptive scheduling allows a process to be interrupted in the midst of its CPU burst, taking the CPU away and allocating it to another process. Cooperative scheduling ensures that a process keeps control of the CPU until it finishes with its current CPU burst.**

2. What advantage is there in having different time-quantum sizes on different levels of a multilevel queuing system?

**Processes that need more frequent servicing (and faster response time), for instance, interactive processes such as editors, can be in a queue with a small time quantum. Processes with no need for frequent servicing can be in a queue with a larger quantum, requiring fewer context switches to complete the processing, and thus making more efficient use of the computer.**

3. Many CPU-scheduling algorithms are parameterized. For example, the RR algorithm requires a parameter to indicate the time slice. Multilevel feedback queues require parameters to define the number of queues, the scheduling algorithms for each queue, the criteria used to move processes between queues, and so on. These algorithms are thus really sets of algorithms (for example, the set of RR algorithms for all time slices, and so on). One set of algorithms may include another (for example, the RR algorithm is the FCFS algorithm with an infinite time slice). What (if any) relation holds between the following pairs of sets of algorithms?

- |                                  |  |
|----------------------------------|--|
| ⇒ Priority and SJF               | <b>The shortest job has the highest priority.</b>  |
| ⇒ Multi feedback queues and FCFS | <b>The lowest level of MLFQ is usually FCFS.</b>   |
| ⇒ Priority and FCFS              | <b>FCFS assigns priority in terms of the longest time of existence in the Ready queue.</b> |
| ⇒ RR and SJF                     | <b>None.</b>   |

4. Suppose that a scheduling algorithm (at the level of short-term CPU scheduling) favors those processes that have used the least processor time in the recent past. Why will this algorithm favor I/O-bound processes and yet not permanently starve CPU-bound programs? Is it possible that the CPU-bound process be starved and under what conditions?

**It will favor the I/O-bound programs because of the relatively short CPU burst request by them; however, the CPU-bound programs will not starve because the I/O-bound programs will relinquish the CPU relatively often to do their I/O. If the number of I/O processes is sufficiently large to keep at least one I/O bound process in the ready queue, the CPU-bound process will be starved.**

5. Assume an operating system maps user-level threads to the kernel using the many-to-many model where the mapping is done through the use of LWPs (light weight processes). Furthermore, the system allows program developers to create real-time threads. Is it necessary to bind a real-time thread to an LWP?

**Answer: Yes, otherwise a user thread may have to compete for an available LWP prior to being actually scheduled. By binding the user thread to an LWP, there is no latency while waiting for an available LWP; the real-time user thread can be scheduled immediately. Also the OS can assign a high priority to such a thread so that it can receive real-time servicing.**

6. Suppose that the following processes arrive for execution at the times indicated. Each process will run the listed amount of time.

Process	Arrival Time	Burst Time
P1	0.0	8
P2	0.4	4
P3	1.0	1

- a) What is the average turnaround time for these processes with the FCFS scheduling algorithm?  
**P1 arrives and starts executing at 0.0 and terminates at 8, thus turnaround time is  $(8-0.0) = 8$**   
**P2 arrives at 0.4, starts executing at time 8 and terminates at 12, thus turnaround time is  $(12-0.4) = 11.6$**   
**P3 arrives at 1.0, starts executing at 12 and terminates at 13, and thus has a turnaround time is  $(13-1)=12$ .**  
**Average turnaround time =  $(8 + 11.6 + 12)/3 = 10.53$**
- b) What is the average turnaround time for these processes with the SJF (no preemption) scheduling algorithm?  
**P1 arrives and starts executing at time 0.0 and terminates at 8, thus the turnaround time is  $(8-0) = 8$ .**  
**P2 arrives at 0.4, starts executing at 9 (after P3 has executed), and terminates at 13, thus the turnaround time is  $(13-0.4) = 12.6$ .**  
**P3 arrives at 1.0, starts executing at 8 and terminates at 9, thus the turnaround time is  $(9-1) = 8$ .**  
**Average turnaround time =  $(8+12.6+8)/3 = 9.53$**
- c) The SJF algorithm is supposed to improve performance, but notice that we chose to run process P1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average turnaround time will be if the CPU is left idle for the first 1 time unit and then SJF scheduling is used. Remember that processes P1 and P2 are waiting during this idle time, so their waiting time may increase. This algorithm could be known as future-knowledge scheduling.  
**Remember that the CPU is left idle for the first 1 time unit.**  
**P1 arrives time 0.0, starts executing at 6 (after P2 and P3 have executed) and terminates at 14, thus the turnaround time is  $(14-0) = 14$ .**  
**P2 arrives at 0.4, starts executing at 2.0 (after P3 has executed), and terminates at 6, thus the turnaround time is  $(6-0.4) = 5.6$ .**  
**P3 arrives at 1.0, starts executing at 1 (it has the shortest burst and starts executing first) and terminates at 2, thus the turnaround time is  $(2-1) = 1$ .**  
**Average turnaround time =  $(14+5.6+1)/3 = 6.87$**
- d) What is the average turnaround time for these processes with the preemptive SJF (SRTF – shortest remaining time first) scheduling algorithm?  
**P1 arrives time 0.0,**  
**starts executing at 0,**  
**is preempted by P2 at 0.4 with 7.6 t.u. to execute,**  
**resumes only after P2 and P3 have executed at time 5.4,**  
**and terminates at time  $(5.4+7.6) 13$ ,**  
**thus the turnaround time =  $13-0 = 13$ .**  
**P2 arrives at 0.4,**  
**starts executing at 0.4,**  
**is preempted by P3 at 1 with 3.4 t.u. to execute,**  
**resumes after P3 is done at time 2,**  
**and terminates at  $(2+3.4) = 5.4$ ,**  
**thus the turnaround time is  $(5.4 - 0.4) = 5$ .**  
**P3 arrives at 1.0, preempts P2 that is executing and starts executing at 1 and terminates at 2, thus the turnaround time is  $(2-1) = 1$ .**  
**Average turnaround time =  $(13+5+1)/3 = 6.33$**   
**To be fair to the other algorithms, the number of context switches should be taken into account and added to the above turnaround times. The preemptive SJF will generate the most context switches.**

7. Consider three processes P1, P2, and P3 with the following CPU burst times:

Burst times for P1: 14, 12, 17

Burst times for P2: 2, 2, 2, 3, 2, 2, 2, 3

Burst times for P3: 6, 3, 8, 2, 1, 3, 4, 1, 9, 7

All three arrive at time 0, in order P1, P2, P3. Each CPU burst is followed by an I/O operation taking 6 time units, except for the last burst after which the process terminates. Using the provided tables, simulate the execution of these processes using the following scheduling algorithms. Assume that overhead time for process switching and scheduling functions are negligible (assumed to be 0).

- a) FCFS
- b) SJF (no pre-emption)
- c) Preemptive SJF
- d) Round robin with a quantum of 5 time units.
- e) Round robin with a quantum of 5 time units with priority:  $P2=P3 > P1$ .
- f) Multi-level feedback queuing using three queues with the following parameters:
  - i. Queue 0 – quantum of 2 time units (after which process is moved to Queue 1)
  - ii. Queue 1 – quantum of 4 time units (after which process is moved to Queue 2)
  - iii. Queue 2 – FCFS
  - iv. All process that becomes ready is added to Queue 0.
  - v. A process that arrives in Queue 0 preempts any executing process that belongs to either Queue 1 or Queue 2.
  - vi. Processes in Queue 1 are allocated to the CPU only when Queue 0 is empty.
  - vii. Processes in Queue 2 are allocated to the CPU only with both Queue 0 and Queue 1 are empty.

Table for algorithms a to d:

[illegible]

[illegible]

FCFS				
Time	CPU	Ready Queue	Blocked (waiting) Queue	Terminated
0	P1(0)	P2(0), P3(0)		
14	P2(0)	P3(0)	P1(14)	
16	P3(0)		P1(14), P2(2)	
20	P3(4)	P1(14)	P2(2)	
22	P3(6)	P1(14), P2(2)		
22	P1(14)	P2(2)	P3(6)	
28	P1(20)	P2(2), P3(6)		
34	P2(2)	P3(6)	P1(26)	
36	P3(6)		P1(26), P2(4)	
39			P1(26), P2(4), P3(9)	
40	P1(26)		P2(4), P3(9)	
42	P1(28)	P2(4)	P3(9)	
45	P1(31)	P2(4), P3(9)		
57	P2(4)	P3(9)		P1(43)
59	P3(9)		P2(6)	P1(43)
65	P3(15)	P2(6)		P1(43)
67	P2(6)		P3(17)	P1(43)
70			P3(17), P2(9)	P1(43)
73	P3(17)		P2(9)	P1(43)
75			P2(9), P3(19)	P1(43)
76	P2(9)		P3(19)	P1(43)
78			P3(19), P2(11)	P1(43)
81	P3(19)		P2(11)	P1(43)
82			P2(11), P3(20)	P1(43)
84	P2(11)		P3(20)	P1(43)
86			P3(20), P2(13)	P1(43)
88	P3(20)		P2(13)	P1(43)
91			P2(13), P3(23)	P1(43)
92	P2(13)		P3(23)	P1(43)
94			P3(23), P2(15)	P1(43)
97	P3(23)		P2(15)	P1(43)
100	P3(26)	P2(15)		P1(43)
101	P2(15)		P3(27)	P1(43)
104			P3(27), P2(18)	P1(43)
107	P3(27)		P2(18)	P1(43)
108			P2(18), P3(28)	P1(43)
110	P2(18)		P3(28)	P1(43)
112			P3(28), P2(20)	P1(43)
114	P3(28)		P2(20)	P1(43)
118	P3(32)	P2(20)		P1(43)
123	P2(20)		P3(37)	P1(43)
125			P3(37), P2(22)	P1(43)
129	P3(37)		P2(22)	P1(43)
131	P3(39)	P2(22)		P1(43)
136	P2(22)			P1(43), P3(44)
138			P2(24)	P1(43), P3(44)
144	P2(24)			P1(43), P3(44)
147				P1(43), P3(44), P2(27)

Round Robin (quantum = 5)				
Time	CPU	Ready Queue	Blocked (waiting) Queue	Terminated
0	P1(0)	P2(0), P3(0)		
5	P2(0)	P3(0), P1(5)		
7	P3(0)	P1(5)	P2(2)	
12	P1(5)	P3(5)	P2(2)	
13	P1(6)	P3(5), P2(2)		
17	P3(5)	P2(2), P1(10)		
18	P2(2)	P1(10)	P3(6)	
20	P1(10)		P3(6), P2(4)	
24	P1(14)	P3(6)	P2(4)	
24	P3(6)		P2(4), P1(14)	
26	P3(8)	P2(4)	P1(14)	
27	P2(4)		P1(14), P3(9)	
29			P1(14), P3(9), P2(6)	
30	P1(14)		P3(9), P2(6)	
33	P1(17)	P3(9)	P2(6)	
35	P1(19)	P3(9), P2(6)		
35	P3(9)	P2(6), P1(19)		
40	P2(6)	P1(19), P3(14)		
43	P1(19)	P3(14)	P2(9)	
48	P3(14)	P1(24)	P2(9)	
49	P3(15)	P1(24), P2(9)		
51	P1(24)	P2(9)	P3(17)	
53	P2(9)		P3(17), P1(26)	
55			P3(17), P1(26), P2(11)	
57	P3(17)		P1(26), P2(11)	
59	P3(19)	P1(26)	P2(11)	
59	P1(26)		P2(11), P3(19)	
61	P1(28)	P2(11)	P3(19)	
64	P2(11)	P1(31)	P3(19)	
65	P2(12)	P1(31), P3(19)		
66	P1(31)	P3(19)	P2(13)	
71	P3(19)	P1(36)	P2(13)	
72	P3(20)	P1(36), P2(13)		
72	P1(36)	P2(13)	P3(20)	
77	P2(13)	P1(41)	P3(20)	
78	P2(14)	P1(41), P3(20)		
79	P1(41)	P3(20)	P2(15)	
81	P3(20)		P2(15)	P1(43)
84			P2(15), P3(23)	P1(43)
85	P2(15)		P3(23)	P1(43)
88			P3(23), P2(18)	P1(43)
90	P3(23)		P2(18)	P1(43)
94	P3(27)	P2(18)		P1(43)
94	P2(18)		P3(27)	P1(43)
96			P3(27), P2(20)	P1(43)
100	P3(27)		P2(20)	P1(43)
101			P2(20), P3(28)	P1(43)
102	P2(20)		P3(28)	P1(43)
104			P3(28), P2(22)	P1(43)
107	P3(28)		P2(22)	P1(43)

Round Robin (quantum = 5)				
Time	CPU	Ready Queue	Blocked (waiting) Queue	Terminated
110	P3(31)	P2(22)		P1(43)
112	P2(22)	P3(33)		P1(43)
114	P3(33)		P2(24)	P1(43)
118			P2(24), P3(37)	P1(43)
120	P2(24)		P3(37)	P1(43)
123			P3(37)	P1(43), P2(27)
124	P3(37)			P1(43), P2(27)
129	P3(42)			P1(43), P2(27)
131				P1(43), P2(27), P3(44)



SJF				
Time	CPU	Ready Queue	Blocked (waiting) Queue	Terminated
0	P1(0)	P2(0), P3(0)		
14	P2(0)	P3(0)	P1(14)	
16	P3(0)		P1(14), P2(2)	
20	P3(4)	P1(14)	P2(2)	
22	P3(6)	P1(14), P2(2)		
22	P2(2)	P1(14)	P3(6)	
24	P1(14)		P3(6), P2(4)	
28	P1(18)	P3(6)	P2(4)	
30	P1(20)	P3(6), P2(4)		
36	P2(4)	P3(6)	P1(26)	
38	P3(6)		P1(26), P2(6)	
41			P1(26), P2(6), P3(9)	
42	P1(26)		P2(6), P3(9)	
44	P1(28)	P2(6)	P3(9)	
47	P1(31)	P2(6), P3(9)		
59	P2(6)	P3(9)		P1(43)
62	P3(9)		P2(9)	P1(43)
68	P3(15)	P2(9)		P1(43)
70	P2(9)		P3(17)	P1(43)
72			P3(17), P2(11)	P1(43)
76	P3(17)		P2(11)	P1(43)
78	P3(19)	P2(11)		P1(43)
78	P2(11)		P3(19)	P1(43)
80			P3(19), P2(13)	P1(43)
84	P3(19)		P2(13)	P1(43)
85			P2(13), P3(20)	P1(43)
86	P2(13)		P3(20)	P1(43)
88			P3(20), P2(15)	P1(43)
91	P3(20)		P2(15)	P1(43)
94	P3(23)	P2(15)		P1(43)
94	P2(15)		P3(23)	P1(43)
97			P3(23), P2(18)	P1(43)
100	P3(23)		P2(18)	P1(43)
103	P3(26)	P2(18)		P1(43)
104	P2(18)		P3(27)	P1(43)
106			P3(27), P2(20)	P1(43)
110	P3(27)		P2(20)	P1(43)
111			P2(20), P3(28)	P1(43)
112	P2(20)		P3(28)	P1(43)
114			P3(28), P2(22)	P1(43)
117	P3(28)		P2(22)	P1(43)
120	P3(31)	P2(22)		P1(43)
126	P2(22)		P3(37)	P1(43)
128			P3(37), P2(24)	P1(43)
132	P3(37)		P2(24)	P1(43)
134	P3(39)	P2(24)		P1(43)
139	P2(24)			P1(43), P3(44)
142				P1(43), P3(44), P2(27)

Preemptive SJF				
Time	CPU	Ready Queue	Blocked (waiting) Queue	Terminated
0	P2(0)	P1(0), P3(0)		
2	P3(0)	P1(0)	P2(2)	
8	P3(6)	P1(0), P2(2)		
8	P2(2)	P1(0)	P3(6)	
10	P1(0)		P3(6), P2(4)	
14	P3(6)	P1(4)	P2(4)	
16	P3(8)	P1(4), P2(4)		
17	P2(4)	P1(4)	P3(9)	
19	P1(4)		P3(9), P2(6)	
23	P1(8)	P3(9)	P2(6)	
25	P2(6)	P3(9), P1(10)		
28	P1(10)	P3(9)	P2(9)	
32	P3(9)		P2(9), P1(14)	
34	P2(9)	P3(11)	P1(14)	
36	P3(11)		P1(14), P2(11)	
38	P3(13)	P1(14)	P2(11)	
42	P3(17)	P1(14), P2(11)		
42	P2(11)	P1(14)	P3(17)	
44	P1(14)		P3(17), P2(13)	
48	P3(17)	P1(18)	P2(13)	
50	P3(19)	P1(18), P2(13)		
50	P2(13)	P1(18)	P3(19)	
52	P1(18)		P3(19), P2(15)	
56	P3(19)	P1(22)	P2(15)	
57	P1(22)		P2(15), P3(20)	
58	P2(15)	P1(23)	P3(20)	
61	P1(23)		P3(20), P2(18)	
63	P1(25)	P3(20)	P2(18)	
64	P3(20)		P2(18), P1(26)	
67	P3(23)	P2(18)	P1(26)	
67	P2(18)		P1(26), P3(23)	
69			P1(26), P3(23), P2(20)	
70	P1(26)		P3(23), P2(20)	
73	P3(23)	P1(29)	P2(20)	
75	P2(20)	P1(29), P3(25)		
77	P3(25)	P1(29)	P2(22)	
79	P1(29)		P2(22), P3(27)	
83	P2(22)	P1(33)	P3(27)	
85	P2(24)	P1(33), P3(27)		
85	P3(27)	P1(33)	P2(24)	
86	P1(33)		P2(24), P3(28)	
91	P2(24)	P1(38)	P3(28)	
92	P2(25)	P1(38), P3(28)		
94	P1(38)	P3(28)		P2(27)
99	P3(28)			P2(27), P1(43)
108			P3(37)	P2(27), P1(43)
114	P3(37)			P2(27), P1(43)
121				P2(27), P1(43), P3(44)

RR (quantum=5) with priority (P2=P3 > P1)				
Time	CPU	Ready Queue	Blocked (waiting) Queue	Terminated
0	P2(0)	P1(0), P3(0)		
2	P3(0)	P1(0)	P2(2)	
7	P3(5)	P1(0)	P2(2)	
8	P3(6)	P1(0), P2(2)		
8	P2(2)	P1(0)	P3(6)	
10	P1(0)		P3(6), P2(4)	
14	P3(6)	P1(4)	P2(4)	
16	P3(8)	P1(4), P2(4)		
17	P2(4)	P1(4)	P3(9)	
19	P1(4)		P3(9), P2(6)	
23	P3(9)	P1(8)	P2(6)	
25	P3(11)	P1(8), P2(6)		
28	P2(6)	P1(8), P3(14)		
31	P3(14)	P1(8)	P2(9)	
34	P1(8)		P2(9), P3(17)	
37	P2(9)	P1(11)	P3(17)	
39	P1(11)		P3(17), P2(11)	
40	P3(17)	P1(12)	P2(11)	
42	P1(12)		P2(11), P3(19)	
44			P2(11), P3(19), P1(14)	
45	P2(11)		P3(19), P1(14)	
47			P3(19), P1(14), P2(13)	
48	P3(19)		P1(14), P2(13)	
49			P1(14), P2(13), P3(20)	
50	P1(14)		P2(13), P3(20)	
53	P2(13)	P1(17)	P3(20)	
55	P2(15)	P1(17), P3(20)		
55	P3(20)	P1(17)	P2(15)	
58	P1(17)		P2(15), P3(23)	
61	P2(15)	P1(20)	P3(23)	
64	P2(18)	P1(20), P3(23)		
64	P3(23)	P1(20)	P2(18)	
68	P1(20)		P2(18), P3(27)	
70	P2(18)	P1(22)	P3(27)	
72	P1(22)		P3(27), P2(20)	
74	P3(27)	P1(24)	P2(20)	
75	P1(24)		P2(20), P3(28)	
77			P2(20), P3(28), P1(26)	
78	P2(20)		P3(28), P1(26)	
80			P3(28), P1(26), P2(22)	
81	P3(28)		P1(26), P2(22)	
83	P3(30)	P1(26)	P2(22)	
86	P3(33)	P1(26), P2(22)		
86	P2(22)	P1(26), P3(33)		
88	P3(33)	P1(26)	P2(24)	
92	P1(26)		P2(24), P3(37)	
94	P2(24)	P1(28)	P3(37)	
97	P1(28)		P3(37)	P2(27)
98	P3(37)	P1(29)		P2(27)
103	P3(42)	P1(29)		P2(27)
105	P1(29)			P2(27), P3(44)
110	P1(34)			P2(27), P3(44)
115	P1(39)			P2(27), P3(44)
119				P2(27), P3(44), P1(43)

Multi-level Feedback Queues						
Time	CPU	Queue 0	Queue 1	Queue 2	Blocked	Terminated
0	P1(0)	P2(0), P3(0)				
2	P2(0)	P3(0)	P1(2)			
4	P3(0)		P1(2)		P2(2)	
6	P1(2)		P3(2)		P2(2)	
10	P2(2)		P3(2)	P1(6)		
12	P3(2)			P1(6)	P2(4)	
16	P1(6)				P2(4), P3(6)	
18	P2(4)			P1(8)	P3(6)	
20	P1(8)				P3(6), P2(6)	
22	P3(6)			P1(10)	P2(6)	
24	P3(8)			P1(10)	P2(6)	
25	P1(10)				P2(6), P3(9)	
26	P2(6)			P1(11)	P3(9)	
28	P2(8)			P1(11)	P3(9)	
29	P1(11)				P3(9), P2(9)	
31	P3(9)			P1(13)	P2(9)	
33	P3(11)			P1(13)	P2(9)	
35	P2(9)		P3(13)	P1(13)		
37	P3(13)			P1(13)	P2(11)	
39	P1(13)			P3(15)	P2(11)	
40	P3(15)				P2(11), P1(14)	
42					P2(11), P1(14), P3(17)	
43	P2(11)				P1(14), P3(17)	
45					P1(14), P3(17), P2(13)	
46	P1(14)				P3(17), P2(13)	
48	P1(16)	P3(17)			P2(13)	
48	P3(17)		P1(16)		P2(13)	
50	P1(16)				P2(13), P3(19)	
51	P2(13)		P1(17)		P3(19)	
53	P1(17)				P3(19), P2(15)	
56	P3(19)			P1(20)	P2(15)	
57	P1(20)				P2(15), P3(20)	
59	P2(15)			P1(22)	P3(20)	
61	P2(17)			P1(22)	P3(20)	
62	P1(22)				P3(20), P2(18)	
63	P3(20)			P1(23)	P2(18)	
65	P3(22)			P1(23)	P2(18)	
66	P1(23)				P2(18), P3(23)	
68	P2(18)			P1(25)	P3(23)	
70	P1(25)				P3(23), P2(20)	
71					P3(23), P2(20), P1(26)	
72	P3(23)				P2(20), P1(26)	
74	P3(25)				P2(20), P1(26)	
76	P2(20)				P1(26), P3(27)	
77	P2(21)	P1(26)			P3(27)	
78	P1(26)				P3(27), P2(22)	
80	P1(28)				P3(27), P2(22)	
82	P3(27)		P1(30)		P2(22)	
83	P1(30)				P2(22), P3(28)	
84	P2(22)		P1(31)		P3(28)	
86	P1(31)				P3(28), P2(24)	
87	P1(32)				P3(28), P2(24)	
89	P3(28)			P1(34)	P2(24)	
91	P3(30)			P1(34)	P2(24)	
92	P2(24)		P3(31)	P1(34)		

Multi-level Feedback Queues						
Time	CPU	Queue 0	Queue 1	Queue 2	Blocked	Terminated
94	P3(31)		P2(26)	P1(34)		
97	P2(26)			P1(34), P3(34)		
98	P1(34)			P3(34)		P2(27)
107	P3(34)					P2(27), P1(43)
110					P3(37)	P2(27), P1(43)
116	P3(37)					P2(27), P1(43)
118	P3(39)					P2(27), P1(43)
122	P3(43)					P2(27), P1(43)
123						P2(27), P1(43), P3(44)