CS 5/7343 Fall 2021

Homework 1

Question 1:

A. Shortest job first. Avg wait time: 13.4

Time	0	1	3	4	6	11	15	23	32
Current	P2	P2	P2	P2	P2	P3	P4	P5	P1
Queue	{}	{P5}	{P3, P5}	{P3, P4, P5}	{P3, P4, P5, P1}	{P4, P5, P1}	{P5, P1}	{P1}	{}

B. Shortest remaining job first. Avg wait time: 12.2

Time	0	1	3	4	6	7	14	22	32
Current	P2	P5	P3	P3	P3	P5	P4	P2	P1
Queue	{}	{P2}	{P5, P2}	{P5, P4, P2}	{P5, P4, P2, P1}	{P4, P2, P1}	{P2, P1}	{P1}	{}

C. Priority-based, quantum = 3. Avg wait time: 15.4

0	1	3	4	6	12	15	18	21	22	25	32
P2	P5	P5	P4	P4	P3	P5	P1	P3	P5	P1	P2
{}	4: {P2}	2: {P3} 4: {P2}	2: {P3, P5} 4: {P2}	2: {P3, P5, P1} 4: {P2}	2: {P5, P1} 4: {P2}	2: {P1, P3} 4: {P2}	2: {P3, P5} 4: {P2}	2: {P5, P1} 4: {P2}	2: {P1} 4: {P2}	4: {P2}	{}

D. Priority-based, quantum = 5. Avg wait time: 14.6

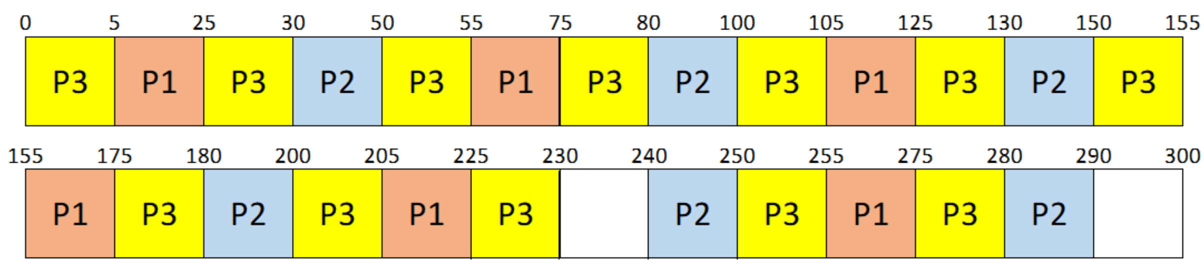
Time	0	1	3	4	6	12	16	21	26	27	32
Current	P2	P5	P5	P4	P4	P3	P5	P1	P5	P1	P2
Queue	{}	4: {P2}	2: {P3} 4: {P2}	2: {P3, P5} 4: {P2}	2: {P3, P5, P1} 4: {P2}	2: {P5, P1} 4: {P2}	2: {P1} 4: {P2}	2: {P5} 4: {P2}	2: {P1} 4: {P2}	4: {P2}	{}

Question 2:

A. $(r_1/s_1) + (r_2/s_2) + (r_3/s_3) = (20/50) + (20/60) + (5/25) = 0.4 + 0.33 + 0.2 =$ **0.93**

B. $N = 3$, $N(2^{1/N} - 1) = 3(2^{1/3} - 1) = 0.78$

The total utilization is greater than the lower bound, however it is less than 1, so it is not proven either way by its utilization. However, the schedule will repeat every 300 time units (Lowest common denominator of periods. All periods will have a deadline at t = 300, and therefore the periods will 'reset' as if they were at 0 again.), and *can* be scheduled within that window as such:



C. Since the total utilization is under 100%, EDF can be guaranteed to be able to schedule the processes theoretically (not accounting for the utilization cost of context switching) according to the textbook. Due to the utilization being only 93%, we can safely assume that the cost of context switching would not exceed the 7% we have left on our CPU, and therefore can be scheduled.

Question 3:

Assuming that test and set is not atomic, there is a situation which breaks the critical section problem. If, for example, two processes check the lock before either one sets it to true, then both processes will continue to run even though one should have waited. If test and set was an atomic operation, this would not be possible, as if one process ran the operation, then another would not be able to check the lock before it was set, but since the operation is assumed to not be atomic, this situation fails to solve the problem, and therefore non-atomic test and set can not be used to solve the problem.

Question 4:

A. This does not change anything. If that line is reached, either the other process has already finished its critical section, or the other process is going to wait regardless because it will set turn to that value itself or already has and therefore turn does not change.

B. This will also break the solution. If either of the processes run line 2 and then run the wait loop without being interrupted, they will not wait and immediately enter the critical section.