CSCI 423/501 Discussion Note 09

Name: Feng Wang ________________ Last 4 Digits of Student #:____________________

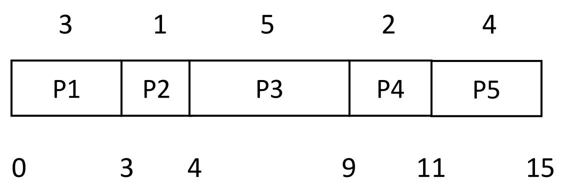
Group Member:
Name1: CSCI 423/501 Class ________
Name2: __________________________
Name3: __________________________

Discussion Questions: (for **Chapter 6.1 to 6.4**, as well as Slides on **Chapter 6**)

1. For the following processes have arrived in the order P1, P2, P3, P4, P5, all at time 0.

Process	Burst Time	Priority
P1	3	3
P2	1	1
P3	5	4
P4	2	2
P5	4	3

Draw the Gantt chart that illustrate the execution of these processes using the FCFS scheduling algorithm. What is the turnaround time of each process, the waiting time of each process, and the average waiting time over all processes? Explain what is the convoy effect.



Turnaround time: (finishing time – arrival time)
P1: 3
P2: 4
P3: 9
P4: 11
P5: 15

Waiting time: (turnaround time - burst time)
P1: 3-3=0
P2: 4-1 =3
P3: 9-5 = 4
P4: 11-2 = 9
P5: 15-4 = 11

Average waiting time:
(0+3+4+9+11)/5 = 5.4

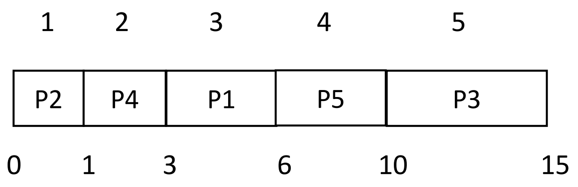
Convoy effect refers to the situation where all other short processes wait for one big long process to get off the CPU. E.g., one CPU-bound and many I/O-bound processes. After I/O, all I/O-bound processes must wait in the ready queue until the CPU-bound process finishes using CPU.

CSCI 423/501 Discussion Note 09

2. For the following processes have arrived in the order P1, P2, P3, P4, P5, all at time 0 (same to previous question).

Process	Burst Time	Priority
P1	3	3
P2	1	1
P3	5	4
P4	2	2
P5	4	3

Draw the Gantt chart that illustrate the execution of these processes using the SJF scheduling algorithm. What is the turnaround time of each process, the waiting time of each process, and the average waiting time over all processes? Also explain in practice how the next CPU burst of each process can be predicted.



Turnaround time: (finishing time – arrival time)
P1: 6
P2: 1
P3: 15
P4: 3
P5: 10

Waiting time: (turnaround time - burst time)
P1: 6-3=3
P2: 1-1 =0
P3: 15-5 = 10
P4: 3-2 = 1
P5: 10-4 = 6

Average waiting time:
(3+0+10+1+6)/5=4

The next CPU burst is approximated by predicting with an exponential average of the measured lengths of previous CPU bursts.

1. $t_n =$ actual length of n^{th} CPU burst
2. $\tau_{n+1} =$ predicted value for the next CPU burst
3. $\alpha, 0 \le \alpha \le 1$
4. Define :

$$\tau_{n=1} = \alpha\, t_n + (1-\alpha)\tau_n.$$

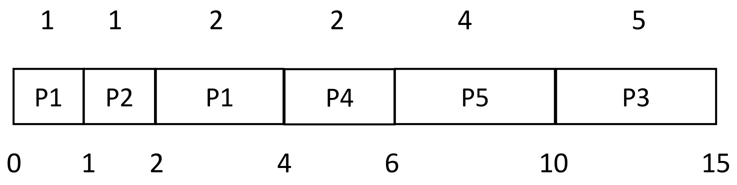
Commonly, α is set to ½.

CSCI 423/501 Discussion Note 09

3. For the following processes have arrived in the order P1, P2, P3, P4, P5.

Process	Burst Time	Priority	Arrival Time
P1	3	3	0
P2	1	1	1
P3	5	4	2
P4	2	2	3
P5	4	3	4

Draw the Gantt chart that illustrate the execution of these processes using the Shortest-Remaining-Time-First scheduling algorithm. What is the turnaround time of each process, the waiting time of each process, and the average waiting time over all processes? What is the difference between this scheduling algorithm and the original SJF algorithm?



Turnaround Time: (finishing time – arrival time)
P1: 4-0=4
P2: 2-1 = 1
P3: 15-2 = 13
P4: 6-3 = 3
P5: 10-4 = 6

Waiting time: (turnaround time - burst time)
P1: 4-3 = 1
P2: 1-1 = 0
P3: 13-5 = 8
P4: 3-2 = 1
P5: 6-4 = 2

Average waiting time:
(1+0+8+1+2)/5 = 2.4

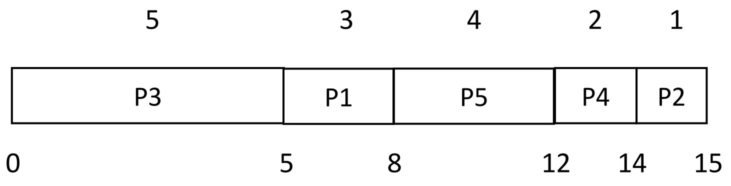
The Shortest-Remaining-Time-First scheduling algorithm is the preemptive version of SJF scheduling algorithm.

CSCI 423/501 Discussion Note 09

4. For the following processes have all arrived in the order P1, P2, P3, P4, P5, at time 0.

Process	Burst Time	Priority
P1	3	3
P2	1	1
P3	5	4
P4	2	2
P5	4	3

Draw the Gantt chart that illustrate the execution of these processes using the Priority scheduling algorithm (Assuming largest integer means highest priority). What is the turnaround time of each process, the waiting time of each process, and the average waiting time over all processes?



Turnaround Time: (finishing time – arrival time)
P1: 8
P2: 15
P3: 5
P4: 14
P5: 12

Waiting time: (turnaround time - burst time)
P1: 8-3 = 5
P2: 15-1 = 14
P3: 5-5 = 0
P4: 14-2 = 12
P5: 12-4 = 8

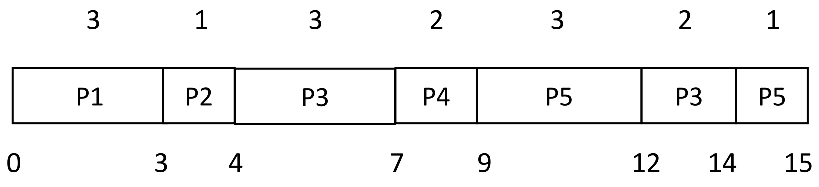
Average waiting time:
(5+14+0+12+8)/5 = 7.8

CSCI 423/501 Discussion Note 09

5. For the following processes have all arrived in the order P1, P2, P3, P4, P5, at time 0.

Process	Burst Time	Priority
P1	3	3
P2	1	1
P3	5	4
P4	2	2
P5	4	3

Draw the Gantt chart that illustrate the execution of these processes using the Round Robin scheduling algorithm with time quantum set to 3. What is the turnaround time of each process, the waiting time of each process, and the average waiting time over all processes? Also explain what would happen if the time quantum is set too small or too large.



Turnaround Time: (finishing time – arrival time)
P1: 3
P2: 4
P3: 14
P4: 9
P5: 15

Waiting time: (turnaround time - burst time)
P1: 3-3 = 0
P2: 4-1 = 3
P3: 14-5 = 9
P4: 9-2 = 7
P5: 15-4 = 11

Average waiting time:
(0+3+9+7+11)/5 = 6

If time quantum is set too small (e.g., comparable or small as to context switch time), there will be too many context switch, and the total context switch overhead will become too high.

If time quantum is set too large, every process/thread can finish within one time quantum, then RR becomes FCFS.

CSCI 423/501 Discussion Note 09

6. Among simple scheduling algorithms (i.e., FCFS, SJF, Priority, RR), which can minimize the average waiting time? Which is especially suitable for time-sharing systems? Which must be non-preemptive? Which may cause starvation and how to address the starvation issue? Explain the similarity and differences between Multilevel Queue Scheduling and Multilevel Feedback Queue Scheduling. Also, among all the scheduling algorithms (i.e., FCFS, SJF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), which is the most complex and most general one?

SJF

RR

FCFS (Note that SJF and Priority have both preemptive and non-preemptive versions, so they may not be non-preemptive)

Priority scheduling and SJF, where low priority (long) processes may never execute if high priority (short) processes keep coming.

Aging mechanism can help address the starvation, which increases the priority of the process as time progresses. So a low priority process can eventually have high enough priority to be scheduled on the CPU.

Similarity: partition ready queue into separate queues; each queue has its own scheduling algorithm; Scheduling must be done between queues; need to determine which queue a new coming process should enter.

Difference: whether allow processes to move between queues
In Multilevel Queue Scheduling, process permanently in a given queue.
In Multilevel Feedback Queue Scheduling, process can move between various queues, so need additional mechanism to determine when to upgrade/demote a process (i.e., to move a process from one queue to another).

Multilevel Queue Feedback Scheduling