COMP 7500/7506 Advanced Operating Systems
Homework 1: CPU Scheduling Algorithms

1. [30 points] Discuss how the following pairs of scheduling criteria conflict in certain settings.

1.1 [15 points] CPU utilization and response time

Ans: The CPU utilization is a relative measure of the non-idle processing taking place on the processor. To enhance CPU utilization, it should remain as busy as possible. Response time is the difference in time at which a process gets CPU first time and arrival time. It's the duration of time taken by a processor to respond. For performance purposes, it's ideal to increase CPU utilization and reduce response time but there is a conflicting relationship between the two in certain settings. For example, CPU utilization is hampered by the context switch even though it reduces response time for processes. In a brief, if context switches have been increased, the processes won't have to wait in the ready queue for a long time that reduces response time but at the same time, it hampers CPU utilization. So, it's not possible to improve CPU utilization and reduce response time together in this setting.

1.2. [15 points] Average turnaround time and maximum waiting time

Ans: Average turnaround time is the average time required for a process to complete its job. On the other hand, waiting time is the difference between turnaround time and burst time.
Average turnaround time and maximum waiting time conflicts in certain settings. For example, average turnaround time can be reduced by implementing the shortest job first (SJF) algorithm but SJF can result in starvation (indefinite wait for longer jobs) and therefore increasing the waiting time.

2. [40 points] Consider the following set of processes, with the length of the CPU burst time given in milliseconds. The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

Process	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

2.1 [10 points] Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, non-preemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).

Ans:

Gantt chart for FCFS:

P1	P2	P3	P4	P4	
0	2	3	11	15	20

Gantt chart for SJF:

P2	P1	P4	P5	P3	
0	1	3	7	12	20

Gantt chart for non-preemptive priority:
Equal-priority processes are scheduled in FCFS order so incase of P1 and P4 having equal priorities, P1 will take precedence over P4.

P3	P5	P1	P4	P2	
0	8	13	15	19	20

Gantt chart for RR:
Time slice (quantum) = 2

P1	P2	P3	P4	P5	P3	P4	P5	P3	P5	P3	
0	2	3	5	7	9	11	13	15	17	18	20

2.2 [10 points] What is the turnaround time of each process for each of the scheduling algorithms in Question 2.1?

2.3 [10 points] What is the waiting time of each process for each of these scheduling algorithms?

Ans: Turn around is the time interval from the submission of the process to its completion.
Turnaround time = Completion time – Arrival time

Waiting time is the amount of time spent in the waiting queue.
Waiting time = Turnaround time – burst time

First Come First Serve (FCFS):

process	Arrival Time	Burst Time	Completion Time	Turn-around time	Waiting Time
P1	**0**	**2**	**2**	**2**	**0**
P2	**0**	**1**	**3**	**3**	**2**
P3	**0**	**8**	**11**	**11**	**3**
P4	**0**	**4**	**15**	**15**	**11**
P5	**0**	**5**	**20**	**20**	**15**

Shortest job first (SJF):

Process	Arrival time	Burst Time	Completion time	Turnaround time	Waiting time
P1	**0**	**2**	**3**	**3**	**1**
P2	**0**	**1**	**1**	**1**	**0**
P3	**0**	**8**	**20**	**20**	**12**
P4	**0**	**4**	**7**	**7**	**3**
P5	**0**	**5**	**12**	**12**	**7**

Non-Preemptive priority based:

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting time
P1	**0**	**2**	**15**	**15**	**13**
P2	**0**	**1**	**20**	**20**	**19**
P3	**0**	**8**	**8**	**8**	**0**
P4	**0**	**4**	**19**	**19**	**15**
P5	**0**	**5**	**13**	**13**	**8**

Round Robin (RR):

Process	Arrival time	Burst time	Completion time	Turnaround Time	Waiting time
P1	**0**	**2**	**2**	**2**	**0**
P2	**0**	**1**	**3**	**3**	**2**
P3	**0**	**8**	**20**	**20**	**12**
P4	**0**	**4**	**13**	**13**	**9**
P5	**0**	**5**	**18**	**18**	**13**

2.4 [10 points] Which of the algorithms results in the minimum average waiting time (over all processes)?

Ans:

Average waiting time for :

FCFS = 0+2+3+11+15/5 = 6.2

SJF = 1+0+12+3+7/5 = 4.6

Non-preemptive Priority = 13+19+0+15+8/5 = 11

RR = 0+2+12+9+13/5 = 7.2

SJF algorithm has the minimum average waiting time.

3. [15 points] Which of the following scheduling algorithms could result in starvation? Why?

(1) First-come, first-served
(2) Shortest job first
(3) Round robin
(4) Priority

Ans: Starvation is a condition where a process does not get the resources it needs for a long time because the resources are being allocated to other processes. It generally occurs in a Priority-based Scheduling System. Where High Priority requests get processed first. In the case of priority-based, processes with low priority may remain within the ready queue for an indefinite amount of time while CPU is allocated to high priority processes.
The Shortest Job First (SJF) algorithm may also result in starvation or indefinite blocking. For example, shortest jobs are executed first while long jobs must wait for the completion of shorter jobs. This could also result in starvation if we have a high number of short jobs.

4. **[15 points] Consider a preemptive priority scheduling algorithm based on dynamically changing priorities. Larger priority numbers implies higher priority. When a process is waiting for the CPU (in the ready queue, but not running), its priority changes at a rate a; when it is running, its priority changes at a rate b. All processes are given a priority of 0 when they enter the ready queue. The parameters a and b can be set to give many different scheduling algorithms. What is the algorithm that results from b > a > 0? Please justify your answer.**

Ans:
The algorithm that results from b > a >0 is First-come, first-served (FCFS).

Justification: All the processes in the ready queue has the same initial priority 0 (P_0 =0). When a process is waiting for the CPU in the ready queue, its priority increases at the

same positive rate a (a>0). Thus, the earlier the process enters the ready queue (t_0), the higher its priority will be.
Once the process with the highest priority (first - come compared to other processes in the ready queue) is running, its priority increases at a higher rate (b>a>0) than the priorities of those processes in the ready queue. So, no other process will preempt it and it will run to its completion.