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

Points Possible: **100**
Submission via **Canvas**

This is an individual assignment; no collaboration among students. Students shouldn't share any homework solution with any other student. Collaborations among students in any form will be treated as a serious violation of the University's academic integrity code.

Learning Objectives:

- To understand CPU scheduling
- To describe various CPU scheduling algorithms
- To study evaluation criteria for CPU scheduling algorithms

Questions:

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

1.1 [15 points] CPU utilization and response time

Answer 1.1: If we want to maximize the CPU usage then we need to minimize the context switches and minimizing the context switches result in increase in waiting time as process has to wait for longer time.

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

Answer 1.2: Turnaround time is the time for complete execution of a process from start till end and our main objective is to minimize the average turnaround time so that many process get executed in less time. The minimizing of average turnaround time could be possible using shortest job first as it execute processes with least burst time and this strategy could possibly hold large burst time process. The waiting of process with large burst time could increase the maximum waiting time for the process. Therefore, to balance this situation we require tradeoff between them using different scheduling algorithms.

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, nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).

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?

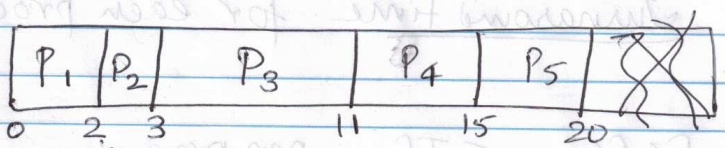
Answer: Note : Answer of question 2 is attached in below documents.

Q2

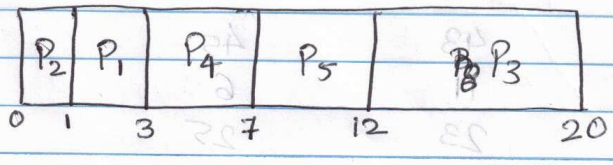
Process	Brust Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

2.1

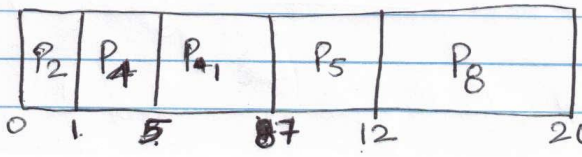
(1) FCFS



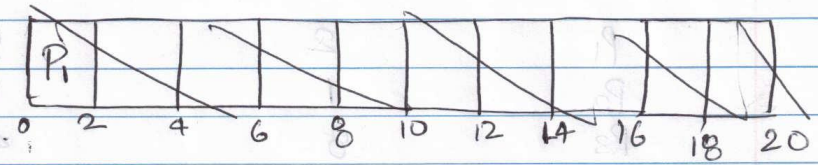
(2) SJF

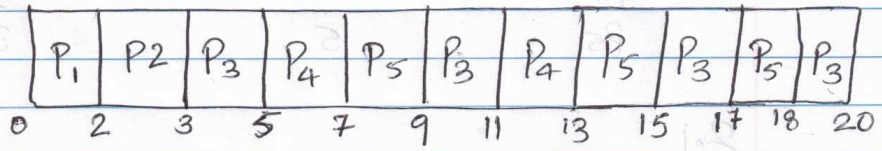


(3) Non-preemptive priority.



④ RR (quantum = 2)





Q2.2

Turnaround time for each process

	FCFS	SJF non-preemptive	RR (q=2)
P_1	2	4	13	2
P_2	5	1	1	3
P_3	16	43	45	20
P_4	31	11	6	13
P_5	51	23	25	18

All process Arrived at time=0

1) FCFS =

$P_1 = 2$

$P_2 = 2+3 = 5$

$P_3 = 2+3+11 = 16$

$P_4 = P_3 + 15 \Rightarrow 31$

$P_5 = P_4 + 20 \rightarrow 51$

(2) SJF

$P_2 = 1$

$P_1 = 1+3 = 4$

$P_4 = 1+3+7 = 11$

$P_5 = 1+3+7+12 = 23$

$P_3 \rightarrow P_5 + 20 \rightarrow 43$

(3) Non-preemptive

$P_2 = 0+1 = 1$

$P_4 = 1+5 = 6$

$P_1 = 6+7 = 13$

$P_5 = 6+7+12 \Rightarrow 25$

$P_3 = P_5 + 20 = 45$

(4) RR

$P_1 = 2$

$P_2 = 3$

$P_3 = 20$

$P_4 = 13$

$P_5 = 18$

2.23 Waiting time for each process.

Waiting Time = Turnaround Time - CPU Time.

	fcfs	SJF	Preemptive	RR(q=2)
P_1	0	11	6	0
P_2	2	10	0	0
P_3	5	23	25	12
P_4	16	4	1	9
P_5	31	11	13	13

① fcfs

$P_1 = 0$
$P_2 = 2$
$P_3 = 2+3 = 5$
$P_4 = 2+3+11 = 16$
$P_5 = 2+3+11+15 = 31$

② SJF

$P_1 = 1$
$P_2 = 0$
$P_3 = 0+1+3+7+12 = 23$
$P_4 = 0+1+3 = 4$
$P_5 = 0+1+3+7 = 11$

③ Preemptive

$P_1 = 0+1+5 = 6$
$P_2 = 0$
$P_3 = 0+1+5+7+12 = 25$
$P_4 = 1$
$P_5 = 0+1+5+7 = 13$

④ $P_1 = 2-2 = 0$
$P_2 = 1-1 = 0$
$P_3 = 20-8 = 12$
$P_4 = 13-4 = 9$
$P_5 = 18-5 = 13$

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

Answer 2.4:

Avg waiting time = waiting time / 5

FCFS : (0+2+5+16+31)/5 => 54/5 => 10.8 milliseconds
SJF : (1+0+23+4+11)/5 => 39/5 => 7.8 milliseconds
Preemptive: (6+0+25+1+13)/5 => 45/5 => 9 milliseconds
RR : (0+0+12+9+13)/5 => 34/5 => 6.8 milliseconds

The Round Robin with quantum of 2 has 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

Answer 3: I think Shortest Job first and Priority Scheduling will starve because in priority scheduling algorithm, processes with low priority will not execute until all above process with higher priority is executed. Usually, there are always some process that has high priority will execute first and processes with least priority will starve. For Shortest job first (SJF), the process with least execution will be executed first and process with large execution has wait for small process to finish and if in some scenario where there are plenty of small process then large execution process with starve to get executed. Hence, I think option 2 and 4 are correct.

4. [15 points] Consider a preemptive priority scheduling algorithm based on dynamically changing priorities. Larger priority numbers imply 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.

Answer 4:

I think the implemented algorithm should be the FCFS algorithm because of the following reasons:

- **The initial priority of all the algorithms is 0 and as the priority increase to alpha, the process is ready in the queue and as it start executing its priority changed to beta, So if were to use FCFS scheduling algorithm we would the same**

change in rate condition. For example, if a process (p1) is in ready queue waiting for it chance to get CPU, its current priority rate is alpha. Now the p1 gets the CPU and occupy CPU until it complete execution without interruption which implicitly means that process p1 must have the highest priority among all other processes present in the ready queue. Therefore, the rate changes to beta (high priority). Now, if we connect dot based from the above example then this condition b>a>0 fits well with FCFS scheduling algorithm as the process in FCFS occupy the CPU until it is completely executed.

Submission:

- Submit your solution as a PDF file named as ""hw1.pdf" through Canvas

Late Submission Penalty:

- Ten percent (10%) penalty per day for late submission. For example, an assignment submitted after the deadline but up to 1 day (24 hours) late can achieve a maximum of 90% of points allocated for the assignment. An assignment submitted after the deadline but up to 2 days (48 hours) late can achieve a maximum of 80% of points allocated for the assignment.
- Assignment submitted more than 3 days (72 hours) after the deadline will not be graded.

Rebuttal period:

- You will be given a period of one week (i.e., 7 days) to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.