Some exercises on Scheduling

1. Explain how the following pairs of scheduling criteria can conflict:
   (a) Maximizing CPU utilization while minimizing response time
   (b) Minimizing average turnaround time while minimizing waiting time
   (c) Maximizing I/O device utilization and maximizing CPU utilization

2. "Lottery scheduling" is a scheduling algorithm that works by assigning lottery tickets to threads, which are used for allocating CPU time. Whenever the scheduler needs to schedule a thread, a lottery ticket is chosen at random, and the thread holding that ticket gets the CPU. Assume we have a scheduler that implements the lottery scheduling by holding a lottery 50 times each second, with each lottery winner getting 20 milliseconds of CPU time (20 milliseconds $\times$ 50 = 1 second). Describe how this scheduler can ensure that higher-priority threads receive more attention from the CPU than lower-priority threads.

3. A variation of the round-robin scheduling algorithm is the "regressive round-robin" algorithm This algorithm assigns each thread a time quantum and a priority. The initial value of a time quantum is 50 milliseconds. However, every time a thread has been allocated the CPU and uses its entire time quantum (does not block for I/O), 10 milliseconds is added to its time quantum, and its priority level is boosted. (The time quantum for a process can be increased to a maximum of 100 milliseconds.) When a process blocks before using its entire time quantum, its time quantum is reduced by 5 milliseconds, but its priority remains the same. What type of thread (CPU-bound or I/O-bound) does the regressive round-robin scheduling algorithm favor? Explain.

4. Consider the following set of processes, with the length of the CPU burst time given in milliseconds

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

The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5 all at time 0.

(a) Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1).

(b) What is the turnaround time of each process for each of the scheduling algorithms in part a?

(c) What is the waiting time of each process for each of these scheduling algorithms?

(d) Which of the algorithms results in the minimum average waiting time (over all processes)?

5. The following processes are being scheduled using a preemptive, round-robin scheduling algorithm. Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an idle task (which consumes no CPU resources and is identified as P_{idle}). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.

Thread	Priority	Burst	Arrival
P_1	40	20	0
P_2	30	25	25
P_3	30	25	30
P_4	35	15	60
P_5	5	10	100
P_6	10	10	105

(a) Show the scheduling order of the processes using a Gantt chart.

(b) What is the turnaround time for each process?

(c) What is the waiting time for each process?

(d) What is the CPU utilization rate?

6. Which of the following scheduling algorithms could result in starvation? a) First-come, first-served; b) Shortest job first; c) Round robin; d) Priority

7. 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 α; when it is running, its priority changes at a rate β. All processes are given a priority of 0 when they enter the ready queue. The parameters α and β can be set to give many different scheduling algorithms.

(a) What is the algorithm that results from $\beta > \alpha > 0$?

(b) What is the algorithm that results from $\alpha < \beta < 0$?

8. Explain the differences in how much the following scheduling algorithms discriminate in favor or against short processes (sum of CPU and I/O bursts is short):

   (a) FCFS:

   (b) RR:

   (c) Multilevel feedback queues:

9. Consider the scheduling algorithm in the Solaris operating system for time-sharing threads:

   (a) What is the time quantum (in milliseconds) for a thread with priority 10? With priority 55?

   (b) Assume a thread with priority 35 has used its entire time quantum without blocking. What new priority will the scheduler assign this thread?

   (c) Assume a thread with priority 35 blocks for I/O before its time quantum has expired. What new priority will the scheduler assign this thread?

10. Using the Windows scheduling algorithm, determine the numeric priority of each of the following threads

    (a) A thread in the REALTIME_PRIORITY_CLASS with a relative priority of HIGHEST.

    (b) A thread in the NORMAL_PRIORITY_CLASS with a relative priority of NORMAL.

    (c) A thread in the HIGH_PRIORITY_CLASS with a relative priority of ABOVE_NORMAL.