Week 1 Exercises: Processes

From "Operating Systems Concepts" Silberschatz, Galvin and Gagne.

1. A CPU scheduling algorithm determines an order for the execution of its scheduled processes. Given n processes to be scheduled on one processor, how many possible different schedules are there? Give a formula in terms of n.

2. Define the difference between preemptive and nonpreemptive scheduling.

3. Suppose that the following processes arrive for execution at the times indicated. Each process will run the listed amount of time. In answering the questions, use nonpreemptive scheduling and base all decisions on the information you have at the time the decision must be made.

Process	Arrival Time	Burst Time
P1	0	8
P2	0.4	4
P3	1	1

   a. What is the average turnaround time for these processes with the FCFS scheduling algorithm?
   b. What is the average turnaround time for these processes with the SJF scheduling algorithm?
   c. The SJF algorithm is supposed to improve performance, but notice that we chose to run process P1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average turnaround time will be if the CPU is left idle for the first 1 unit and then SJF scheduling is used. Remember that processes P1 and P2 are waiting during this idle time, so their waiting time may increase. This algorithm could be known as future-knowledge scheduling.

4. What advantage is there in having different time-quantum sizes on different levels of a multilevel queueing system?

5. Many CPU-scheduling algorithms are parametrised. For example, the RR algorithm requires a parameter to indicate the time slice. Multilevel feedback queues require parameters to define the number of queues, the scheduling algorithms for each queue, the criteria used to move processes between queues, and so on.
   These algorithms are thus really sets of algorithms (for example, the set of RR algorithms for all time slices, and so on). One set of algorithms may include another (for example, the FCFS algorithm is the RR algorithm with an infinite time quantum).What (if any) relation holds between the following pairs of sets of algorithms?
   a. Priority and SJF
   b. Multilevel feedback queues and FCFS
   c. Priority and FCFS
   d. RR and SJF

6. Suppose that a scheduling algorithm (at the level of short-term CPU scheduling) favours those processes that have used the least processor time in the recent past. Why will this algorithm favor I/O-bound programs and yet not permanently starve CPU-bound programs?

7. Why is it important for the scheduler to distinguish I/O-bound programs from CPU-bound programs?

8. Discuss how the following pairs of scheduling criteria conflict in certain settings.
   a. CPU utilization and response time
   b. Average turnaround time and maximum waiting time
   c. I/O device utilization and CPU utilization

9. Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

   The processes are assumed to have arrived in the order P1, P2,P3, P4, P5, all at time 0.

   a. Draw four Gantt charts (like the pictures in the lecture slides) that illustrate the execution of these processes using the following scheduling algorithms:
      1. FCFS;
      2. SJF;
      3. Non-pre-emptive priority (a smaller priority number implies a higher priority),
      4. 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 the scheduling algorithms in part a?
   d. Which of the algorithms in part a results in the minimum average waiting time (over all processes)?

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

11. Consider a variant of the RR scheduling algorithm in which the entries in the ready queue are pointers to the PCBs.
    a. What would be the effect of putting two pointers to the same process in the ready queue?
    b. What would be two major advantages and two disadvantages of this scheme?
    c. How would you modify the basic RR algorithm to achieve the same effect without the duplicate pointers?

12. Consider a system running ten I/O bound tasks and one CPU-bound task. Assume that the I/O-bound tasks issue an I/O operation once for every- millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context-switching overhead is 0.1 millisecond and that all processes are long-running tasks.
    What is the CPU utilization for a round-robin scheduler when:

a. The time quantum is 1 millisecond
b. The time quantum is 10 milliseconds

13. Consider a system implementing multilevel queue scheduling. What strategy can a computer user employ to maximize the amount of CPU time allocated to the user's process?

14. Consider a pre-emptive 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 p. All processes are given a priority of 0 when they enter the ready queue. The parameters a and p can be set to give many different scheduling algorithms.
a. What is the algorithm that results from $p > a > 0$?
b. What is the algorithm that results from $a < p < 0$?

15. Explain the differences in the degree to which the following scheduling algorithms discriminate in favour of short processes:
a. FCFS
b. RR
c. Multilevel feedback queues

16. Using the Windows XP scheduling algorithm, what is the numeric priority of a thread for the following scenarios?
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