

CPU Scheduling



Practice Exercises

- 5.1 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 .
- 5.2 Define the difference between preemptive and nonpreemptive scheduling.
- 5.3 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.

<u>Process</u>	<u>Arrival Time</u>	<u>Burst Time</u>
P_1	0.0	8
P_2	0.4	4
P_3	1.0	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 P_1 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 P_1 and P_2 are waiting during this idle time, so their waiting time may increase. This algorithm could be known as future-knowledge scheduling.

- 5.4 What advantage is there in having different time-quantum sizes on different levels of a multilevel queueing system?
- 5.5 Many CPU-scheduling algorithms are parameterized. 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?
- Priority and SJF
 - Multilevel feedback queues and FCFS
 - Priority and FCFS
 - RR and SJF
- 5.6 Suppose that a scheduling algorithm (at the level of short-term CPU scheduling) favors 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?
- 5.7 Distinguish between PCS and SCS scheduling.
- 5.8 Assume an operating system maps user-level threads to the kernel using the many-to-many model where the mapping is done through the use of LWPs. Furthermore, the system allows program developers to create real-time threads. Is it necessary to bind a real-time thread to an LWP?