# **Exercises-Chapter 5 CPU Scheduling**

## 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 different schedules are possible? Give a formula in terms of *n.*

## *2.* Explain the difference between preemptive and non-preemptive scheduling.

## Give an example of each type of scheduling.

## 3. What events cause the CPU scheduler to run?

## 4. What is a good job mix? How does it affect the performance of the system?

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

## Process Arrival Time Burst Time

## *P*1 0 8

## *P*2 4 4

## *P*3 10 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 (non-preemtive) 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 10 units 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.

## 6. Compute the waiting time, turnaround time and response time for each process under the following scheduling schemes for the process list below.

## Process         Arrival Time    Burst Time    Priority

## P1                   0                    10                3

## P2                   2                     1                  1

## P3                   3                     5                  2

## P4                   5                     8                  2

## P5                   7                     6                  3

## a.       SJF scheduling (Shortest Job First)

## b.      SRTF scheduling (Shortest Remaining Time First)

## c.       FCFS scheduling (First Come First Served)

## d.      Priority scheduling (with preemption)

## e.       RR scheduling with time slice of 2 units. ( Round Robin)

7. Name one advantage and disadvantage of each of the following scheduling schemes.

## SRTF scheduling (Shortest Remaining Time First)

## FCFS scheduling (First Come First Served)

## Priority scheduling (with preemption)

## RR scheduling with time slice ( Round Robin)

## 8. Consider the following priority scheduling scheme. Larger  priority numbers imply higher priorities. When the process is waiting for CPU, its priority changes at rate x. When it is running, its priority changes at the rate y. All processes have priority 0 to start with.

## a) What is the algorithm we get if y > x > 0? Explain.

## b) What is the algorithm we get if x > y > 0? Explain.

## 9. Describe the differences between multilevel queues and multilevel feedback queues? Can any of these cause starvation? Explain

## 10. Explain the differences in the degree to which the following algorithms discriminate in favor of a short process: (heavily favors short process, somewhat favors short process, heavily discriminates against short process)

## a) FCFS

## b) RR

## c)  Multilevel Feedback queues

## 

## 11. What advantage is there in having different time-quantum sizes at different levels of a multilevel queueing system?

## 12. 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 algorithm sets?

## a. Priority and SJF

## b. Multilevel feedback queues and FCFS

## c. Priority and FCFS

## d. RR and SJF

## 13. 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?

## 12. The traditional UNIX scheduler enforces an inverse relationship between priority numbers and priorities: the higher the number, the lower the priority. The scheduler recalculates process priorities once per second using the following function:

## Priority = (recent CPU usage / 2) + base

## where base = 60 and *recent CPU usage* refers to a value indicating how often a process has used the CPU since priorities were last recalculated. Assume that recent CPU usage for process *P*1 is 40, for process *P*2 is 18 and for process *P*3 is

## What will be the new priorities for these three processes when priorities are recalculated? Based on this information, does the traditional UNIX scheduler raise or lower the relative priority of a CPU-bound process?