

Worked out Examples

1. What scheduling policy will you use for each of the following cases? Explain your reasons for choosing them.

- a. The processes arrive at large time intervals:
- b. The system's efficiency is measured by the percentage of jobs completed.
- c. All the processes take almost equal amounts of time to complete.

Ans:

- a. FCFS
- b. STF
- c. FCFS or RR

2. In which of the following operations, the scheduler is not called into play?

- a. Process requests for I/O.
- b. Process finishes execution.
- c. Process finishes its time allotted.
- d. All of the above through c
- e. None of the a through c above.

Ans: d

3. What are the factors that need to be considered to determine the degree of multiprogramming in a system?

Ans: The two factors that need to be considered are:

- 1. The overheads in context switching may become excessive.
- 2. With excessive multiprogramming the response times may become unacceptable.

4. What happens if the time allocated in a Round Robin Scheduling is very large? And what happens if the time allocated is very low?

Ans: It results in a FCFS scheduling. If time is too low, the processor through put is reduced. More time is spent on context switching

5. What is the difference between the idle and blocked state of a process?

Ans: In idle state, the process is waiting for the processor to become free so that it can execute. In blocked state, the process has been put out from the running state by the processor due to some I/O.

6. Put the following in the chronological order in the context of the birth of a process executes: Ready, suspended, execute, terminate, create.

Ans: Create, Ready, Execute, Suspended, Terminate

7. When a process requests for I/O, how many process switches take place?

Ans: Two. In the first switch, the process to be switched is taken out and the scheduler starts executing. Then the next process is brought to execution. So there are two process switches.

8. A Shortest Job First algorithm may lead to starvation where a process with large execution time is made to wait for indefinitely long times. Suggest a modification to the SJF that overcomes this problem.

Ans: A clock value (arrival time) is stored for each process. This helps to determine the priority of a process as a function of execution time and the clock value.

9. If the waiting time for a process is p and there are n processes in the memory then the CPU utilization is given by,

- a. p/n
- b. p^n (p raised to n)
- c. $1-p^n$
- d. $n-(p^n)$

Ans: p^n

10. Suppose a new process in a system arrives at an average of six processes per minute and each such process requires an average of 8 seconds of service time. Estimate the fraction of time the CPU is busy in a system with a single processor.

Ans: Given that there are on an average 6 processes per minute.

So the arrival rate = 6 process/min.

i.e. every 10 seconds a new process arrives on an average.

Or we can say that every process stays for 10 seconds with the CPU

Service time = 8 sec.

Hence the fraction of time CPU is busy = service time / staying time

$$= 8 / 10$$

$$= 0.8$$

So the CPU is busy for 80% of the time.

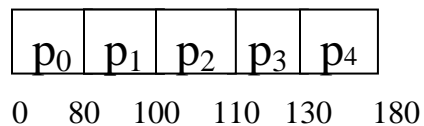
11. Assume you have the following jobs to execute with one processor, with the jobs arriving in the order listed here:

<u>i</u>	<u>T(pi)</u>
0	80
1	20
2	10
3	20
4	50

- Suppose a system uses FCFS scheduling .Create a Gantt chart illustrating the execution of these processes?
- What is the turnaround time for process p3?
- What is the average wait time for the processes?

Ans:

- The Gantt chart is as follows:



- The turnaround time for process p3 is

$$\begin{aligned}
 \text{T.A. (p3)} &= \text{T (p3)} + \text{T.A. (p2)} \\
 &= \text{T (p3)} + (\text{T (p2)} + \text{T.A. (p1)}) \\
 &= \text{T (p3)} + (\text{T (p2)} + (\text{T (p1)} + \text{T.A. (p0)})) \\
 &= \text{T (p3)} + (\text{T (p2)} + (\text{T (p1)} + \text{T (p0)})) \\
 &= 20 + 10 + 20 + 80 \\
 &= 130.
 \end{aligned}$$

- Average waiting time calculation:

Waiting Time for process p0 = 0 sec.

“ “ “ “ p1 = 80 sec.

“ “ “ “ p2 = 100 sec.
 “ “ “ “ p3 = 110 sec.
 “ “ “ “ p4 = 130 sec.

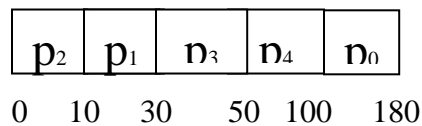
Hence the average waiting time = $(0+80+100+110+130)/5$
 $= 84$ sec.

12. Using the process load in the previous problem, suppose a system uses SJN scheduling.

- Create a Gantt chart illustrating the execution of these processes?
- What is the turnaround time for process p4?
- What is the average wait time for the processes?

Ans:

- The Gantt chart is:



- The turnaround time for process P4 is = 100.
- Average waiting time calculation:

Waiting time for process p0 = 0 sec.

“ “ “ “ p1 = 10 sec.
 “ “ “ “ p2 = 30 sec.
 “ “ “ “ p3 = 50 sec.
 “ “ “ “ p4 = 100 sec.

Therefore, the average waiting time is = $(0+10+30+50+100)/5$
 $= 38$

13. Assume you have the following jobs to execute with one processor, with the jobs arriving in the order listed here: **M3**

<u>i</u>	<u>T (p_i)</u>	<u>Arrival Time</u>
0	80	0
1	20	10
2	10	10
3	20	80

4**50****85**

- a. Suppose a system uses RR scheduling with a quantum of 15 .Create a Gantt chart illustrating the execution of these processes?
- b. What is the turnaround time for process p3?
- c. What is the average wait time for the processes?

Ans:

- a. As the Round-Robin Scheduling follows a circular queue implementation, the Gantt chart is as follows:

p ⁰	p ¹	p ²	p ⁰	p ¹	p ²	p ⁰	p ¹	p ³	p ⁴	p ⁰	p ³	p ⁴	p ⁰	p ⁴	
0	15	30	45	60	75	85	100	110	125	140	155	160	175	190	205

- b. The turnaround time for process P3 is =160-80
= 80 sec.

- c. Average waiting time :

Waiting time for process p₀ = 0 sec.

“ “ “ “ p₁ = 5 sec.

“ “ “ “ p₂ = 20 sec.

“ “ “ “ p₃ = 30 sec.

“ “ “ “ p₄ = 40 sec.

Therefore, the average waiting time is $(0+5+20+30+40)/5 = 22$ sec.

14. Suppose a new process in a system arrives at an average of six processes per minute and each such process requires an average of 8 seconds of service time. Estimate the fraction of time the CPU is busy in a system with a single processor.

Ans: Given that there are on an average 6 processes per minute.

So the arrival rate = 6 process/min.

i.e. every 10 seconds a new process arrives on an average.

Or we can say that every process stays for 10 seconds with the CPU

Service time = 8 sec.

Hence the fraction of time CPU is busy = service time / staying time

$$= 8 / 10$$

$$= 0.8$$

So the CPU is busy for 80% of the time.

15. 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 .

Ans: Since there are n processes to be scheduled, and the first schedule can be done for any of the n processes, the total numbers of possible schedules are n factorial $\Rightarrow n!$.

16. Consider the following 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 X when it is running, its priority changes at a rate Y . All processes are given a priority of 0 when they enter the ready queue. The parameters can be set to give many different scheduling algorithms. What is the algorithm that results from $Y > X > 0$?

- a. LIFO
- b. FCFS
- c. Round Robin
- d. None of the above

Ans: b

17. 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 .

- a. $n(n-1)$
- b. n^2

c. $n!$

d. $n/2$

Ans: c