Q.1) Max Marks: 1

Consider all the processes are arriving at large time intervals.
Let t be the time interval between two processes p_i and p_{i+1} for any i and service time of
p_i is s_i.
$t>s_i$ for every i, then find the best strategy to schedule the processes.

A **FCFS** Correct Option

Solution: (A)
Answer:A
Explanation:
Preemption causes the context switching between the processes and the switching time is very high compared to the service times. SJF, RR, SRTF algorithms allows preemption. So FCFS is the suggestable algorithm as it does not involves the preemption.

B **SJF**

C **RR**

D **SRTF**

Q.2) Max Marks: 1

Which of the following is not TRUE
A. Consider a scenario involving 4 processes to be scheduled. Assume that scheduling and context switch overhead takes 0 ms. There is a case where non preemptive First Come First Served scheduling will have a better average wait time than preemptive Round Robin
B. Advantages of a preemptive CPU scheduling algorithm have over a non-preemptive one is it avoids starvation due to one process running forever.

A **a, b**

B **only a**

C **only b** Correct Option

Solution: (C)
SOLUTION:
a is TRUE : Some common cases where this might happen are: when there are equal duration jobs and they all arrive at the same time; when jobs arrive in increasing order of their durations; and when the time quantum of the pre-emptive scheduler is very small resulting in the wait time for all jobs by increasing at least the length of the shortest job.
B is False : Priority scheduling and Shortest remaining time first algorithm cannot avoid starvation. We cannot generalize by saying *preemptive scheduling algorithms avoid starvation*.

D **None of the above**

Q.3) Max Marks: 1

Assume that main() calls the following function test() exactly once

```
void test(void)
{
    if(fork() == 0)
    {
        printf("GATE ");
        exit(0);
    }
    printf("2020");
}
```

What will be the possible output?

A **GATE 2020**

B **2020 GATE**

C **Both (A) and (B)** Correct Option

Solution: (C)

Answer: C

Explanation :

Fork() returns 0 to the child process and some +ve value to the parent process. So Gate will be printed by the child and 2020 will be printed by the parent. Now order of execution cannot be determined. Any of them can execute at first. So A and B both are possible.

If we don't write exit(0) in the *if statement* then both the child and parent will print the "2020". So this exit(0) statement makes the child to not print the "2020"

D None of these

Q.4) Max Marks: 1

Which of the following strategy is employed for overcoming the priority inversion problem?

A **Temporarily raise the priority of lower priority level process** Correct Option

Solution: (A)
Soln) Answer is A.
In aging problem , lower priority processes lead to starvation .
Priority inversion is a problematic scenario in scheduling, in which a high priority task is indirectly preempted by a lower priority task effectively "inverting" the relative priorities of the two tasks.
https://en.wikipedia.org/wiki/Priority_inversion

B **Have a fixed priority level scheme.**

C **Implement Kernel pre-emption scheme.**

D **Allow lower priority process to complete its job.**

Q.5) Max Marks: 1

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.

A **60%**

B **70%**

C **80%** Correct Option

Solution: (C)
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.

D **None of these**

Q.6) Max Marks: 1

Which of the following Algorithm favour CPU bound Bound process ?
1) RR
2) FCFS
3) Multilevel feedback queue

A **1 only**

B **2 only**

C **1 and 2 only** Correct Option

Solution: (C)
Explanation:
Multi level feedback queue favours IO bound process, because those processes are

idle for a long time , their priority increases.and also priority of cpu bound process decreases.
FCFS: as it is non preemptive by nature , it favours cpu bound processes
Round Robin:
Each process gets put back at the end of the queue no matter how much or how little of the quantum was used. I/O bound processes tend to run for a short period of time and then block which means they might have to wait in the queue a long time.

D 1 and 3 only

Q.7) Max Marks: 1

There are five processes that follow non preemptive approach, whose expected running times are 7, 3, 15, 8, and 4 also arrived at the same time. Which of the following order should they be run to minimize the average response time?

A 7, 3, 15, 8 and 4

B 15, 8, 7, 4 and 3

C 3, 4, 7, 8, and 15 — Correct Option

Solution: (C)
Solution: Shortest job First algorithm gives the best response time and hence the order is 3, 4, 7, 8 and 15

D 3, 15, 4, 8 and 7

Q.8) Max Marks: 1

Suppose that there are 5 processes in the ready queue to be scheduled on one processor. How many different possible schedules are there(Assuming that these process follows non preemptive approach)? _____

Correct Answer

Solution: (120)
Solution: 120
Given that these 5 processes follows non preemptive approach i.e once the job is scheduled then the process runs until its job is done. All the process have equal priority any process can be selected. Therefore number of schedules are 5*4*3*2*1 = 120

Q.9) Max Marks: 1

Which of the following are TRUE? 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. Provided with the relation that holds between the following pairs of algorithm sets.

a. Priority and Shortest Job First algorithm, the Shortest job has the least priority
b. FCFS and RR, the FCFS algorithm is the RR algorithm with an infinite time quantum
c. Priority and FCFS, FCFS gives the highest priority to the job having been in existence the longest.

A only a

B only a,b

C only b,c — Correct Option

Solution: (C)

D Only a,b and c

Q.10) Max Marks: 2

Assume that a process has CPU burst time for last three runs as 4, 5 and 4 (last). Given that last predict burst time was 5 and $\alpha = 0.8$. The next predict of CPU burst time for process when CPU scheduler is shortest process next with exponential averaging is _____. in sec (upto 1 to decimal places).

Correct Answer

Solution: (4.2)
Explanation:
$T_{n+1} = \alpha t_n + (1-\alpha) T_n$

where T_{n+1} is the predicted value of the next CPU burst time.
$0 \le \alpha \le 1$
T_n is the predicted value of the last CPU burst (here T_n = 5s)

tn is the (actual) last run time of the process (here tn is 4s) and $\alpha = 0.8$
Putting in all values,
T4 = 0.8 * 4 + (1 - 0.8)*5 = 3.2 + 1 = 4.2 sec.
So, the next predict of CPU burst time for process is 4.2 sec.

Q.11) Max Marks: 2

Consider three CPU intensive processes, which require 10,20 and 30 units of time and arrive at times 0, 2 and 6 respectively. How many context switches are needed if the operating system implements a shortest remaining time first scheduling algorithm? Do not count the context switches at time zero and at the end.

A 4

B 2 Correct Option

Solution: (B)

Answer: B
Explanation:
Shortest remaining time first :- Select the process with the smallest amount of remaining time among all the process to execute first.
So here we will 1st select P1 then P2 and at last P3.

P1	P2	P3
0-10	10-30	30-60

It is mentioned in question, do not count the context switches at time zero and at the end
So, we will have take only 2 context switch into account i.e. P1 to P2 and P2 to P3.

C 3

D 1

Q.12) Max Marks: 2

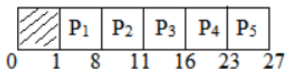
In a cricket match different players of team batting first arrives in different times,Arrival times are as follows,1,4,6,8,and 11. Due to rain initially match delayed 1 unit of time. The time spent by players in the crease are 7,3,5,7 and 4 respectively.Find the completion time of first session with 5 players using first come first serve algorithm is? (Note: Assume that only one player is batting there is no second player at the other end. After the batsmen departs i.e duration as mentioned above the next chosen batsmen arrives at the crease).

A 22

B 27 Correct Option

Solution: (B)

Solution:



Process	AT	BT	CT	TAT
P_1	1	7	8	7
P_2	4	3	11	7
P_3	6	5	16	10
P_4	8	7	23	15
P_5	11	4	27	16

C 29

D 21

Q.13) Max Marks: 2

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 P1 is 40, for process P2 is 18, and for process P3 is 10. What will be the new priorities for these three processes when priorities are recalculated? Based on this information, which of the following are the Priorities after the calculation?

A. P1 > P2 > P3

B. P2 > P1 > P3

C. P2 > P3 > P1

D. P3 > P2 > P1 — Correct Option

Solution: (D)

Solution: The priorities assigned to the processes are 80, 69, and 65 respectively.

Q.14) Max Marks: 2

Consider the following table

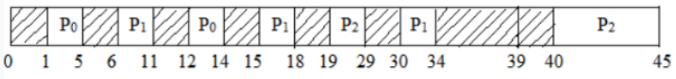
Process	BT	I/O	BT
P_0	4	6	2
P_1	8	10	4
P_2	10	10	5

Using the shortest remaining time first, find the completion time of process p2, if the context switching overhead is 1 unit.

A. 45 — Correct Option

Solution: (A)

Process	BT	I/O	BT
P_0	4	6	2
P_1	8	10	4
P_2	10	10	5



B. 42

C. 32

D. 34

Q.15) Max Marks: 2

A process control block (PCB) exists only for processes in:

A. the ready state.

B. ready and running states.

C. ready and blocked states.

D. ready, running, and blocked states — Correct Option

Solution: (D)

Answer: D

Solution:

The PCB exists for processes in all states until the process exits and the parent picks up its exit via wait.

close