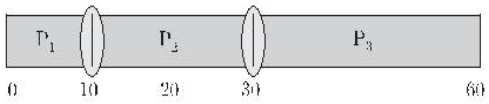
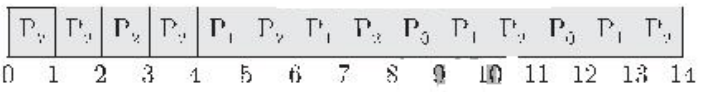
Admission open for UGC-NET June 2016 Courses:
Computer Science, Management, Commerce, Education and Paper 1 (Research & Teaching Aptitude)

Q.No.	Question
1.	Consider n processes sharing the CPU in a round-robin scheduling. Assuming that each process switch takes s seconds, what must be the quantum size q such that the overhead resulting from process switching is minimized but, at the same time, each process is guaranteed to get its turn at the CPU at least every t seconds (i.e., it is idle for no more than t seconds between time slices)? Solution: The next execution of the process should occur within t seconds. Each process runs for q period and if there are n process p1 p2 p3 ..... pn p1 p2 ... then p1 turn comes again when it has completed time quantum for remaining process p2 to pn i.e it would take at most (n-1)q time. So, each process in round robin gets its turn after <= (n-1)q time when we don't consider overheads but if we consider over head(s) then it would be ns+(n-1)q So we have ns + (n-1)q <= t overhead will be reduced when time quantum is maximum allowable i.e, q <= (t-ns)/(n-1)
2.	Consider a system with m resources of same type being shared by n processes. Resources can be requested and released by processes only on at a time. The system is deadlock free if and only if a. The sum of all max needs is < m+n b. The sum of all max needs is > m+n c. Both of above d. None Ans: a
3.	A computer system has 6 tape drives, with n process competing for them. Each process may need three tape drives. The maximum value of n for which the system is guaranteed to be deadlock free is a. 1 b. 2 c. 3 d. 4 **Ans b : by using above formula** The sum of all max needs is < m+n 3n<6+n => 2n < 6 => n<3 i.e. 2 **Ans b:** The maximum value of n for which the system is guaranteed to be deadlock free is 2.Two processes can never lead to deadlock as the peak time demand of (3+3 =6) tape drives can be satisfied. But three processes can lead to deadlock if each process holds 2 drives and then demand one more.
4.	A computer system has 6 tape drives, with n process competing for them. Each process may need two tape drives. The maximum value of n for which the system is guaranteed to be deadlock free is _____? Ans: 2n < 6+n => n < 6 i.e. 5 With number of process n = 3, the drive allocation among process is (2,2,2),which allow all the 3 processes to complete : It is deadlock free With number of process n = 4, the drive allocation among process is (2,2,1,1),which allow first two processes to complete : It is deadlock free With number of process n = 5, the drive allocation among process is (2,1,1,1,1),which allow first to complete : It is deadlock free With number of process n = 6, the drive allocation among process is (1,1,1,1,1,1).Each process holding one tape drive and waiting for another one and hence there is deadlock. Hence for n < 6 the system is deadlock free
5.	Consider a system having m resources of the same type. These resources are shared by 3 processes A, B and C which have peak demands of 3,4 and 6 respectively. For what value of m deadlock will not occur?

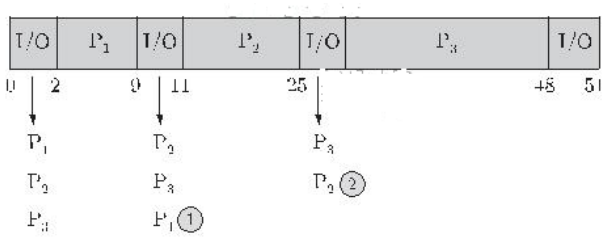
	A. 7 B. 9 C. 10 D. 13 Ans: The sum of all max needs is < m+n (3+4+6) < m + 3 => 13 < m + 3 => m > 10 So ans is d (13) Consider the peak demand situation of resources (A,B,C)=(3,4,6). For a specific number of resources, to conclude there is possibility of deadlock (not deadlock free), we have to find atleast one resource allocation which results in deadlock, that is an allocation that cannot completely satisfy the resource requirements of even one process. With number of shared resources m = 7, the following resource allocation (for example)among 3 process A,B,C (2,3,2).Process A holding 2 resources and waiting for 1, Process B holding 3 resource and waiting for 1 and process C holding 2 resources and waiting for 4 more resources. This is a deadlock situation. With number of shared resources m = 9, the following resource allocation (for example)among 3 process A,B,C (2,3,4).Process A holding 2 resources and waiting for 1, Process B holding 3 resource and waiting for 1 and process C holding 4 resources and waiting for 2 more resources. This is a deadlock situation. With number of shared resources m = 10, the following resource allocation (for example)among 3 process A,B,C (2,3,5).Process A holding 2 resources and waiting for 1, Process B holding 3 resource and waiting for 1 and process C holding 5 resources and waiting for 1 more resource. This is a deadlock situation. But for m >10, the resource allocation is deadlock free.
6.	Suppose n processes, P1……Pn share m identical resource units, which can be reserved and released one at a time. The maximum resource requirement of process Pi is sp where si<0 Which one of the following is a sufficient condition for ensuring that deadlock does not occur? (A) $\forall$ i,s<m (B) $\forall$ i,s < n (C) $\sum_{i=1}^{n}$ si $< (m+n)$ (D) $\sum_{i=1}^{n}$ si $< (m*n)$ Ans: For every P_iS_i is maximum resource requirement where Si > 0. To allot resources to all processes without any deadlock situation is $$\sum_{i=1}^{n} \text{si} < (m+n)$$ i.e. sum of all maximum resource requirement should be less than m +n. Hence (C) is correct option.
7.	Consider a machine with 64 MB physical memory and a 32-bit virtual address space. If the page size is 4 KB, what is the approximate size of the page table? (A) 16 MB (B) 8 MB (C) 2 MB (D) 24 MB Ans: Size of main memory = 64 MB = 2^{26}B Size of virtual memory = 2^{32} B page size = 4KB = 2^{12}B No. of pages = $2^{32}/2^{12} = 2^{20}$ pages, so required 1MB entries. But each entry has both a virtual address & corresponding physical address. Total bits in each entry = 32 +26 (Physical) = 58 = 58/8 = 8 Bytes. So total memory = 8*1 MB= 8 MB
8.	Consider an operating system capable of loading and executing a single sequential user process at a time. The disk head scheduling algorithm used is First Come First Served (FCFS). If FCFS is replaced by shortest seek Time Fist (SSTF), claimed by the vendor to given 50% better benchmark results,

	what is the expected improvement in the I/O performance of user programs? (A) 50% (B) 40% (C) 25% (D) 0% Ans: I/O performance is not entirely dependent upon disk access, it has effect of various other devices, so using SSTF in place of FCFS may reduce disk access time but no improvement in the I/O is done. Hence (D) is correct option.
9.	Consider three CPU-intensive processes, which require 10, 20 and 30 time units 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) 1 (B) 2 (C) 3 (D) 4 Ans: When CPU burst are given to another process, called context switching. The Gantt chart for shortest remaining time first is  So there are two context switches at T = 10 & T = 30 Hence (B) is correct option.
10.	Consider three processes (process id 0,1,2, respectively) with compute time bursts 2,4, and 8 time units. All processes arrive at time zero. Consider the longest remaining time first (LRTF) scheduling algorithm. In LRTF ties are broken by giving priority to the process with the lowest process id. The average turnaround time is (A) 13 units (B) 14 units (C) 15 units (D) 16 units **SOLUTION** Process id 0 1 2 CPU Burst 2 4 8 So we draw Gantt chart for scheduler  At t = 0 longest remaining time for P2 At t = 4 remaining times for both P1 & P2 is 4 so P1 is given priority. At t = 8 remaining time for P0P1 & P2 is 2. P0 is given priority & cyclically done till t = 14. Process T A T P0 12 –0 = 12 P1 13 –0 = 13 P2 14 –0 = 14 Average turnaround time is : 39/3=13 units Hence (A) is correct option.
11.	Consider three processes, all arriving at time zero, with total execution time of 10, 20 and 30 units, respectively. Each process spends the first 20% of execution time doing I/O, the next 70% of time doing computation, and the last 10% of time doing I/O again. The operating system uses a shortest remaining compute time first scheduling algorithm and schedules a new process either when the running process get blocked on I/O or when the running process finishes its compute burst. Assume

that all I/O operations can be overlapped as much as possible. For what percentage of time does the CPU remain idle?
(A) 0% (B) 10.6% (C) 30.0% (D) 89.4%

Process	First I/O	Computation	Second I/O	Total
1	2	7	1	10
2	4	14	2	20
3	6	21	3	30

Since I/0 can be done parallely.
Gantt Chart



Total time taken = 51 units
CPU has to wait = 6 units
=(6/51)*100 =10.6%
Hence (B) is correct option.

12. Two processes, P1 and P2, need to access a critical section of code. Consider the following synchronization construct used by the processes:

```
/* P1 */                        /*P2*/
while (true) {                  while (true) {
wants1=true;                    wants2 = true;
while(wants2==true);            while (wants1 == true);
/* Critical Section */          /* Critical Section */
wants 1 = false;                wants 2 = false;
}                               }
/* Remainder section*/          /*Remainder section*/
```

Here, wants 1 and wants 2 are shared variables, Which are initialized to false. Which one of the following statements is TRUE about the above construct?
(A)It does not ensure mutual exclusion. (B)It does not ensure bounded waiting.
(C)It requires that processes enter the critical section in strict alternation.
(D)It does not prevent deadlocks, but ensures mutual exclusion
SOLUTION
If P1 make wants 1 = true then P2 goes in critical section & vice versa so both together are implementing mutual exclusion but. Since both are accessing wants 1 & wants 2 concurrently 4 wants 1 is first captured by P1 so P2 will wait & P2 captures want 2 so P1 will have to wait.
So a definite deadlock.
Hence (D) is correct option.

13. A process executes the following code for(i = 0;i < n;i ++) fork();
The total number of child processes created is
(A) n (B)2^n-1 (C) 2^n (D)2^{n+1}-1
SOLUTION
The loop is called for n times. The first process is parent process so this should not be counted in child process. But after that every child process has its own child created so after every loop.
2^0,2^1,.....2^n total threads. But subtracting the parent.
Hence (B) is correct option.

14. Consider the methods used by processes P1 and P2 for accessing their critical sections whenever needed, as given below. The initial values of shared Boolean variables S1 and S2 are randomly assigned.

Method used by P1	Method used by P2
While(S1= = S2) Critical Section S1= S2;	While(S1! = S2) Critical Section S1= not(S2);

While one of the following statements describes properties achieved?
(A)Mutual exclusion nut nor progress (B)Progress but not mutual exclusion
(C)Neither mutual exclusion nor progress (D)Both mutual exclusion and progress
Ans: Method used by P1 & P2 enters into critical section when S1 = S2 & S1! = S2 respectively, since both are opposite conditions so mutual exclusion is there, but if P1's while loop true then it will always be true & in P2 if while loop true it would run infinitely so no progress.
Hence (A) is correct option.

15. A system uses FIFO policy for page replacement. It has 4 page frames with no pages loaded to begin with. The system first accesses 100 distinct pages in some order and then access the same 100 pages but now in the reverse order. How many page faults will occur?
(A) 196 (B) 192 (C) 197 (D) 195
Ans: In FIFO page replacement policy, the pages which entered first are replaced first so for first 100 accesses, 100 page faults will occur. Now in memory there are last 4 pages 97, 98, 99 & 100th page. So during reverse access there 4 pages would not create any page fault & other 97 page faults. So total 100 +96 = 196 page faults.
Hence (A) is correct option.

16. The following program consists of 3 concurrent processes and 3 binary semaphores. The semaphores are initialized as S0 = 1, S1 = 0, S2 = 0

Process P0	Process P1	Process P2

while(true){
wait (S0);
print '0'
release (S1);
release (S2);
}
	wait (S1);
	release (S0);
		wait (S2);
		release (S0);

How many times will process P0 print '0'?
(A) At least twice (B) Exactly twice (C) Exactly thrice (D) Exactly once
Ans: Let us see what will happen initially S0 = 1, S1 = 0 S2 = 0 so P1 & P2 will wait & P0 runs & make S0 = 0 wait decrements semaphore by 1 & release increments if by 1.

	So after P0's 1st run S1 = 1 & S2 = 1 '0' is printed Now P1 & P2 can run & make S1 = 0, S2 = 0 but both increments S0 = 1 again. So P0 again starts and print '0' so this would continue again & again. So '0' will be printed at least twice. Hence (A) is correct option.

ALL THE BEST FOR EXAM

FROM

CAREER ACADEMY