GATE CS Topic wise Questions

Operating System

YEAR 2001

Question. 1

Which of the following statements is false ?

(A) Virtual memory implements the translation of a program's address space into physical memory address space.

(B) Virtual memory allows each program to exceed the size of the primary memory.

(C) Virtual memory increases the degree of multi-programming

(D) Virtual memory reduces the context switching overhead.

SOLUTION

Virtual memory enables a program to exceed the size of primary memory so it increases degree of multi-programming.
Since data required by executing program is available here so context switching is reduced.
But virtual memory doesn't translate program's address space into physical memory.
Hence (A) is correct option.

Question. 2

Consider a set of n tasks with known runtimes $r_1, r_2, \ldots\ldots r_n$ to be run on a uniprocessor machine. Which of the following processor scheduling algorithms will result in the maximum throughput ?

(A) Round-Robin (B) Shortest-Job-First

(C) Highest-Response-Ratio-Next (D) First-come-First-Served

SOLUTION

Here the running times $r_1....r_n$ are already known, single processor system. In this scenario, throughput i.e. CPU is maximum utilized in shortest job first scheduling.
Hence (B) is correct option.

Question. 3

Where does the swap space reside ?

(A) RAM (B) Disk

(C) ROM (D) On-chip cache

SOLUTION

Swap space is the memory space where the part of the program not currently in main memory for execution is stored, this program part can be swapped into memory when required.
This space is generally in disk.
Hence (B) is correct option.

Question. 4

Consider a virtual memory system with FIFO page replacement policy. For an arbitrary page access pattern, increasing the number of page frames in main memory will.

(A) Always decrease the number of page faults

(B) Always increase the number of page faults

(C) Sometimes increase the number of page faults

(D) Never affect the number of page faults

SOLUTION

During F1F0 page replacement policy, due to increase in the no. of page frames in memory should decrease the no. of page faults since more frames can be kept there.
But due to Belady's Anomaly after certain limit or in some page access instances no. of page faults are high.
Hence (C) is correct option.

Question. 5

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

SOLUTION

$$\begin{aligned} \text{Size of main memory} &= 64 \text{ MB} \\ \text{Size of virtual memory} &= 2^{32} \text{ B} \\ \text{No. of pages} &= \frac{2^{32}}{2^{12}} \\ &= 2^{20} \text{ pages} \end{aligned}$$

Required 1 M enteries.

But each entry has both a virtual address & corresponding physical address.

$$\begin{aligned} \text{Total bits in each entry} &= 32 + 26 \text{ (Physical)} \\ &= 58 \\ &= \frac{58}{8} = 8 \text{ Bytes.} \\ \text{So total memory} &= 8 \times 1 \text{ MB} \\ &= 8 \text{ MB} \end{aligned}$$

Hence (B) is correct option.

Question. 6

Consider Peterson's algorithm for mutual exclusion between two concurrent processes i and j. The program executed by process is shown below.

```
repeat
    flag[i]=true;
    turn=j;
    while(p)do no-op;
    Enter critical section, perform actions, then
    exit critical section
    Flag[i]=false;
    Perform other non-critical section actions.
Until false;
```

For the program to guarantee mutual exclusion, the predicate P in the while loop should be

(A) flag [j]= true and turn =j (B) flag [j]=true and turn =j

(C) flag [i]=true and turn=j (D) flag [i]=true and turn=i

SOLUTION

While loop if true predicate then the program enters into critical region. This program enters into critical region of flag [i]=true act as semaphore, & true =j, the requirement of resource is by some other process.
Hence (B) is correct option.

YEAR 2002

Question. 7

Which of the following scheduling algorithms is non-preemptive ?

(A) Round Robin

(B) First-In First-Out

(C) Multilevel Queue Scheduling

(D) Multilevel Queue Scheduling with Feedback

SOLUTION

Round robin is preemptive since processes are cycled for CPU time, & run for a particular time stamp in one cycle. Multilevel queue scheduling maintains various quenes, each having different priorities. But in FIFO scheme, only the process which enters once, would be completed first, so no. preemption.
Hence (B) is correct option.

Question. 8

The optimal page replacement algorithm will select the page that

(A) Has not been used for the longest time in the past.

(B) Will not be used for the longest time in the future.

(C) Has been used least number of times.

(D) Has been used most number of times

SOLUTION

Optimal page replacement algorithm assumes that the pages that will come in future are already known, so replacement of the page which will not be used in future occurs.

Hence (B) is correct option.

Question. 9

Which combination of the following features will suffice to characterize an OS as a multi-programmed OS ? More than one program may be loaded into main memory at the same time for execution. (B) If a program waits for certain events such as I/O, another program is immediately scheduled for execution. (C) If the execution of a program terminates, another program is immediately scheduled for execution.

(A) A
(B) A and B
(C) A and C
(D) A, B and C

SOLUTION

Multi-programmed:- More than one program can run on single CPU, when one is blocked.

(A) Is true and a characteristic of multi-programmed
(B) Is true & also characterise a multi-programmed OS
(C) Is true but no necessary for this type this happens in all OS, even in batch processor.

Hence (B) is correct option.

Question. 10

In the index allocation scheme of blocks to a file, the maximum possible size of the file depends on

(A) The size of the blocks, and the size of the address of the blocks
(B) The number of blocks used for the index, and the size of the blocks.
(C) The size of the blocks, the number of blocks used for the index, and the size of the address of the blocks.
(D) None of the above.

SOLUTION

When indexes are created, the maximum no. of blocks given to a file are totally dependent upon size of the index which tells how many blocks can be there, & size of each block.
Hence (B) is correct option.

YEAR 2003

Question. 11

Using a larger block size in a fixed block size file system leads to

(A) better disk throughput but poorer disk space utilization

(B) better disk throughput and better disk space utilization

(C) poorer disk throughput but better disk space utilization

(D) poorer disk throughput and poorer disk space utilization

SOLUTION

Using larger block size in a fixed block size system lead to poor disk space utilization due to data items which are very small comparable to block size cause fragmentation. But it leads to better disk through put since no. of blocks needs to fetch & replace become less.
Hence (A) is correct option.

Question. 12

In a system with 32 bit virtual addresses and 1 KB page size, use of one-level page tables for virtual to physical address translation is not practical because of

(A) the large amount of internal fragmentation

(B) the large amount of external fragmentation

(C) the large memory overhead in maintaining page tables

(D) the large computation overhead in the translation process

SOLUTION

32 bit virtual address, i.e. 2^{32} kB of virtual memory & 1 kB page size.
So total pages $= 2^{32}$.
So. we need to maintain a page table of 2^{32} rows, this require 4 GB main memory which is quite impractical due to large memory

overhead.
Hence (C) is correct option.

Question. 13

A uni-processor computer system only has two processes, both of which alternate 10 ms CPU bursts with 90 ms I/O bursts. Both the processes were created at nearly the same time. The I/O of both processes can proceed in parallel. Which of the following scheduling strategies will result in the least CPU utilizations (over a long period of time) for this system ?

(A) First come first served scheduling

(B) Shortest remaining time first scheduling

(C) Static priority scheduling with different priorities for the two processes

(D) Round robin scheduling with a time quantum of 5 ms.

SOLUTION

There should be no doubt that round robin scheduling would lead to maximum CPU utilization, but since in FCFS one task would starve for a long time so min CPU utilization would be in this case.
Hence (A) is correct option.

Data for Q. 14 & 15 are given below.

A processor uses 2-level page table fro virtual to physical address translation. Page table for both levels are stored in the main memory. Virtual and physical addresses are both 32 bits wide. The memory is byte addressable. For virtual to physical address translation, the 10 most significant bits of the virtual address are used as index into the first level page table while the next 10 bits are used as index into the second level page table. The 12 least significant bits of the virtual address are used as offset within the page. Assume that the page table entries in both levels of page tables are 4 a bytes wide. Further, the processor has a translation look aside buffer(TLB), with a hit rate of 96%. The TLB caches recently used virtual page numbers and the corresponding physical page numbers. The processor also has a physically addressed cache with a bit ratio of 90%. Main memory access time is 10 ns, cache access time is 1 ns, and {LB access time is also $1ns$.

Question. 14

Assuming that no page faults occur, the average time taken to access a virtual address is approximately (to the nearest 0.5 ns)

(A) 1.5 ns (B) 2 ns

(C) 3 ns (D) 4 ns

SOLUTION

TLB is successfully 96% of total request & for remaining 4%. RAM is accessed twice.

So average time taken.

$$\begin{aligned} &= .96(1 + (0.9 \times 1) + 0.1 \times (1 + 10)) \\ &\qquad + .04(21 + (.9 \times .1)) + 0.1 \times (1 + 10) \\ &= .96(1 + .9 + 1.1) + 0.4(21 + .09 + 1.1) \\ &= .96 \times 3 + 0.4 \times 23 \\ &= 2.88 + .92 \\ &= 3.80 \cong 4 \text{ ns (Nearest .5)} \end{aligned}$$

Hence (D) is correct option.

Question. 15

Suppose a process has only the following pages in its virtual address space; two contiguous code pages starting at virtual address 0×0000000, two contiguous data pages starting at virtual address 0×00400000,and a stack page starting at virtual address $0 \times FFFFF000$. The amount of memory required for storing the page tables of this process is

(A) 8 KB (B) 12 KB

(C) 16 KB (D) 20 KB

SOLUTION

$$\begin{aligned} \text{Total no. of pages required} &= 5 \\ \text{But due to 2 level page table} &= 2 \times 4 \text{ kB} \times 2 \\ &= 16 \text{ kB} \end{aligned}$$

Hence (D) is correct option.

Data for Q. 16 & 17 are given below.

Suppose we want to synchronize two concurrent processes P and Q using binary semaphores S and T. The code for the processes P and Q is shown below.

```
Process P                Process Q:
     while(1) {               while(1) {
W:                       Y:
     print '0';               print '1'
     print '0';               print '1'
X:                       Z:
}                        }
```

Synchronization statements can be inserted only at points W,X,Y and Z.

Question. 16

Which of the following will always lead to an output staring with '001100110011'?

(A) P(S) at W, V(S) at X, P(T) at Y, V(T) at Z, S and T inutuakkt 1

(B) P(S) at W, V(T) at X, P(T) at Y, V(S) at Z, S initially 1, and T initially 0

(C) P(S) at W, V(T) at X, P(T) at Y, V(S) at Z, S and T initially 1

(D) P(S) at W, V(T) at X, P(T) at Y, V(S) at Z, S initially 1, and T initially 0

SOLUTION

For output string 001100110011 alternatingly we require process P & Q to execute.
For this to happen P(s) with S = 1 should be placed at W.
At the same time P(T) with T = 0 will be at Y.
At X we have V(T) which will have $T = 1$ so process Q starts.
At the same time at Z we have V(s) which make S = 0 to stop process P.
Hence (B) is correct option.

Question. 17

Which of the following will ensure that the output string never contains a substring of the form 0.1" or 10"1 where n is odd?

(A) P(S) at W, V(S) at X, P(T) at Y, V(T) at Z, S and T initially 1

(B) P(S) at W, V(T) at X, P(T) at Y, V(S) at Z, S and T initially 1

(C) P(S) at W, V(S) at X, P(T) at Y, V(S) at Z, S initially 1

(D) (S) at W, V(T) at X, P(T) at Y, P(S) at Z, S and T initially 1

SOLUTION

To ensure this condition that substring of form 01^n0 or 10^n1, where n is odd S should be initially 1, we will case only 1 semaphore S.
So at W P(s), at X V(s)
Whereas at Y P(s), at Z V(s)
Hence (C) is correct option.

YEAR 2004

Question. 18

Consider the following statements with respect to user-level threads and kernel-supported threads

(i) Context which is faster with kernel-supported threads

(ii) For user-level threads. a system call can block the entire process

(iii) Kernel-supported threads can be scheduled independently

(iv) User-level threads are transparent to the kernel

Which of the above statements are true?

(A) (ii),(iii) and (iv) only

(B) (ii) and (iii) only

(C) (i) and (iii) only

(D) (i) and (ii) only

SOLUTION

(I) It is false, context switch is not faster in support of kernel threads.

(II) A system call can truly block the user level threads, since they don't have permission to do that.

(III) True since kernel supported threads have their independent memory & resources.

(IV) False since user level threads might need support of kernel threads.

Hence (B) is correct option.

Question. 19

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 beachmark results, what is the expected improvement in the I/O performance of user programs?

(A) 50%
(B) 40%
(C) 25%
(D) 0%

SOLUTION

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.

Question. 20

The minimum number of page frames that must be allocated to a running process in a virtual memory environment is determined by

(A) the instruction set architecture
(B) page size
(C) physical memory size
(D) number of processes in memory

SOLUTION

Page frames are allocated in main memory, for virtual memory pages. This no. of page frames depends upon the instruction set architecture.
Hence (A) is correct option.

Question. 21

Consider the following set of processes, with the arrival times and the CPU-burst times given in milliseconds

Process	Arrival time	Burst time

P1	0	5
P2	1	3
P3	2	3
P4	4	1

What is the average turnaround time for these processes with the preemptive shortest remaining processing time first (SRPT) algorithm?

(A) 5.50 (B) 5.75

(C) 6.00 (D) 6.25

SOLUTION

Then around time = (Submit time – finish time)

Gantt chart for the scheduler.

P_1	P_2	P_2	P_2	P_4	P_3	P_3	P_3	P_1	P_1	P_1	P_1

0 1 2 3 4 5 6 7 8 9 10 11 12

$$\begin{aligned} \text{T.A time for } P_1 &= 12 - 0 = 12 \\ P_2 &= 4 - 1 = 3 \\ P_3 &= 8 - 2 = 6 \\ P_4 &= 5 - 4 = 1 \\ \text{Total} &= 22 \\ \text{Avg turn around time} &= \frac{22}{4} \\ &= 5.5 \end{aligned}$$

Hence (A) is correct option.

Question. 22

Consider a system with a two-level paging scheme in which a regular memory access takes 150 nanoseconds, and servicing a page fault takes 8 milliseconds. An average instruction takes 100 nanoseconds of CPU time, and two memory accesses. The TLB hit ratio is 99%, and the page fault rate is one in every 10,000 instructions. What is the effective average instruction execution time?

(A) 645 nanoseconds (B) 1050 nanoseconds

(C) 1215 nanoseconds (D) 1230 nanoseconds

SOLUTION

Memory access time

$$= .90 \times 150 + .10 \times (150 + 150)$$
$$= 135 + 30$$
$$= 165 \text{ ns}$$

$$\text{The error rate} = \frac{1}{10000} = 10^{-4}$$

$$\text{CPU burst time} = 100 \text{ ns}$$

Total execution time

$$= [100 + 2[165] + 10^{-4} \times 8 \times 10^{6}]$$
$$= 100 + 330 + 800$$
$$= 100 + 1130$$
$$= 1230 \text{ ns}$$

Hence (D) is correct option.

Question. 23

Consider two processes P_1 and P_2 accessing the shared variables X and Y protected by two binary semaphores S_x and S_y respectively, both initialized to 1. P and V denote the usual semaphore operators, where P decrements the semaphore value, and V increments the semaphore value. The pseudo-code of P_1 and P_2 is as follows:

P_1:

```
while true do{
  :......
  :.....



                }
```

:

```
while true do{
  :......
  :.......



                }
```

In order to avoid deadlock, the correct operators at L_1, L_2, L_3 and L_4 are respectively

(A) $P(S_Y), P(S_X); P(S_X), P(S_Y)$ (B) $P(S_X), P(S_Y); P(S_Y), P(S_X)$

(C) $P(S_X), P(S_X); P(S_Y), P(S_Y)$ (D) $P(S_X), P(S_Y); P(S_X), P(S_Y)$

SOLUTION

Here semaphores are required to obtain mutual exclusion since both

access X & Y. So at L_1 $P(S_x)$ which means now $S_x =$ wait at $L_2 P(S_y)$ S_y wait, this prevents process P_2 to start access X & Y.
$V(S_x)$ & $V(S_y)$ in the end of P_1 makes S_x & S_y signal so that at L_3 & L_4 $P(S_x)$ & $P(S_y)$ can start.
Hence (D) is correct option.

Question. 24

A Unix-style I-node has 10 direct pointers and one single, one double and one triple indirect pointers. Disk block size is 1 Kbyte, disk block address is 32 bits, and 48-bit integers are used. What is the maximum possible file size?

(A) 2^{24} bytes
(B) 2^{32} bytes
(C) 2^{34} bytes
(D) 2^{48} bytes

SOLUTION

$$\begin{aligned}\text{Size of 1 block} &= 1 \text{ kB}\\ \text{Block addresses size 1 pointer size} &= 32 \text{ bit}\\ &= 4 \text{ bytes.}\\ \text{So no. of pointers in} &= \frac{2^{10} B}{2^2 B}\\ \text{1 block} &= 256\\ \text{So direct pointer will have} &= 10 \times 1\text{kB} = 10\text{kB}\\ \text{Double will have} &= 256 \times 256 \times 1 \text{ kB}\\ \text{Triple will have} &= 256 \times 256 \times 256 \times 1 \text{ kB}\\ &= 2^8 \times 2^8 \times 2^8 \times 2^{10} \text{ B}\\ &= 2^{34} \text{ B}\end{aligned}$$

Hence (C) is correct option.

YEAR 2005

Question. 25

Suppose n processes, $P_1, \ldots.. P_n$ share m identical resource units, which can be reserved and released one at a time. The maximum resource requirement of process P_i is s_p where $s_i < 0$. 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) $\sum_{i=1}^{n} s_i < (m + n)$
(D) $\sum_{i=1}^{n} s_i < (m * n)$

SOLUTION

For every P_i S_i is maximum resource requirement where $S_i > 0$.
To allot resources to all processes without any deadlock situation is

$$\sum_{i=1}^{n} S_i < (m+n)$$

i.e. sum of all maximum resource requirement should be less than $m+n$.
Hence (C) is correct option.

Question. 26

Consider the following code fragment:

if (fork()==0

{$a = a+5$; print f("%d,%/ n"a, and a);}

else {$a-5$; print f("%d,% d/n", a,& a);}

let u, vbe the values printed by the parent process, and x, y be the values printed by the child process. Which one of the following is TRUE?

(A) $u = x+10$ and $v = y$ (B) $u = x+10$ and $v \neq y$

(C) $u+10= x$ and v = y (D) $u+10 = x$ and v ≠ y

SOLUTION

Initial value of a is let 10 and its address & a would be different for both parent & child process so.
(A) & (B) are incorrect also parent process executes $a = a-5 = 5 = u$ & child executes $a = a+5 = 15 = x$ so $u+10 = x$
Hence (D) is correct option.

YEAR 2006

Question. 27

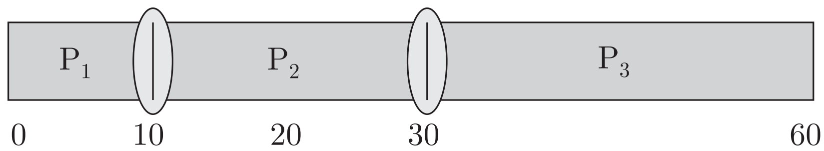
Consider three CPU-intensive processes, which require 10,20 and 30 time units and arrive at times 0,2, and 6, respectively. How many context switches are needed if the operating system implements a shortes remaining time first scheduling algorithm? Do not count the context switches at time zero and at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

(A) 1 (B) 2

(C) 3 (D) 4

SOLUTION

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 & witches at $T = 10$ & $T = 30$
Hence (B) is correct option.

Question. 28

The atomic feth-and-set x, y instruction unconditionally sets the memory location x to 1 and fetches the old value of x in y without allowing any intervening access to the memory location x. Consider the following implementation of P and V functions on a binary semaphore S.

```
void p (binary_semaphore*S){
     unsigned y;
unsigned*x =& (S->value);}
          do {
          fetch-and-set x,y;
     } while(y);
}
void V (binary_semphore*S){
     {S_>value = 0;
}
```

Which one of the following is true?

(A) The implementation may not work if context switching is disabled in P

(B) Instead of using fetch-and-set, a pair of normal load/store can be used

(C) The implementation of V is wrong

(D) The code does not implement a binary semaphore

SOLUTION

If there are more than two processes and context & switching processes is disabled in P then this implementation doesn't work properly and can't synchronize the processes.
Hence (A) is correct option.

Question. 29

A CPU generates 32-bit virtual addresses. The page size is 4 KB. The processor has a translation look-aside buffer (TLB) which can hold a total of 128 page table entries and is 4-way set associative.

The minimum size of the TLB tag is

(A) 11 bits (B) 13 bits

(C) 15 bits (D) 20 bits

SOLUTION

TLB has 128 page table enteries, each page table would have. 64 bits i.e. $32+32$ virtual addresses.
So total memory required.

$$2^7 \times 2^6$$

But 4 way set associative.

$$\frac{2^7 \times 2^6}{2^2} = 2^{11}$$

So 11 bits are required.
Hence (A) is correct option.

Question. 30

A computer system supports 32-bit virtual addresses as well as 32-bit physical addresses, Since the virtual address space is of the same size as the physical address space, the operating system designers decide to get rid of the virtual entirely. Which one of the following is true?

(A) Efficient implementation of multi-user support is no longer possible

(B) The processor cache organization can be made more efficient now

(C) Hardware support for memory management is no longer needed

(D) CPU scheduling can be made more efficient now

SOLUTION

Since both virtual and physical memory has 32 bit addresses, so there is no need for address translation hardware is required.
Hence (C) is correct option.

Question. 31

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 turn around 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

P_2	P_2	P_2	P_2	P_1	P_2	P_1	P_2	P_0	P_1	P_2	P_0	P_1	P_2

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

At $t = 0$ longest remaining time for P_2
At $t = 4$ remaining times for both P_1 & P_2 is 4 so P_1 is given priority.
At $t = 8$ remaining time for $P_0 P_1$ & P_2 is 2 P_0 is given priority & cyclically done till $t = 14$.

Process	T A T
P_0	$12 - 0 = 12$
P_1	$13 - 0 = 13$
P_2	$14 - 0 = 14$

$$\frac{39}{3} = 13 \text{ units.}$$

Hence (A) is correct option.

Question. 32

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?

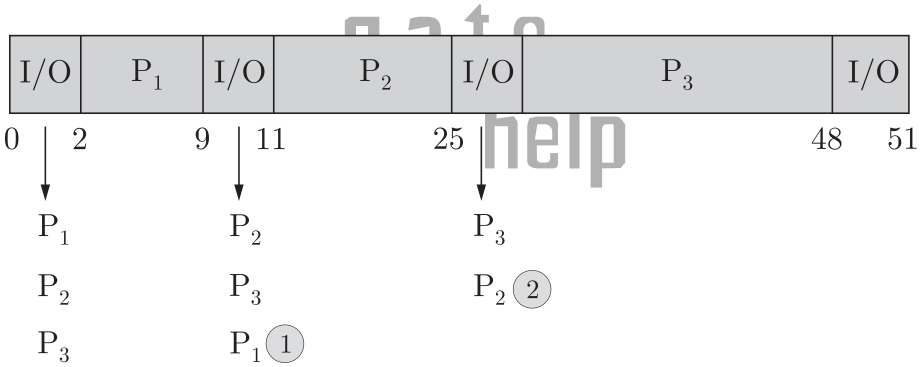
(A) 0% (B) 10.6%

(C) 30.0% (D) 89.4%

SOLUTION

Process	First $I/0$	Computation	Second $I/0$	Total
	2	7	1	10
2	4	14	2	20
3	6	21	3	30

Since $I/0$ can be done parallely.
Gantt chart



$$\text{Total time taken} = 51 \text{ units}$$
$$\text{CPU has to wait} = 6 \text{ units}$$
$$= \frac{6}{51} \times 100$$
$$= 10.6\%$$

Hence (B) is correct option.

Question. 33

Consider the following snapshot of a system running n processes. Process i is holding x_i instances of a resource R, for $1 \le i \le n$. Currently, all instances of R are occupied. Further, for all i, process

i has placed a request for an additional y, instances while holding the x_i instances it already has, There are exactly two processes p and q such that $y_p = y_q = 0$: Which one of the following can serve as a necessary condition to guarantee that the system is not approaching a deadlock?

(A) $\min(x_p, x_q) < \max_{k \neq p,q} y_k$

(B) $x_p + x_q \leq \max_{k \neq p'q} y_k$

(C) $\min(x_p, x_q) < 1$

(D) $\min(x_p, x_q) > 1$

SOLUTION

Here option (B) is that

$\min\,(X_p, X_q) < \max_{k \neq p,q} y_k$

Means the min no. of resources allocated should be less than the maximum no. of resources required by any process other than p & q .It prevent from deadlock.

Hence (B) is correct option.

Data for Q. 34 & 35 are given below.

Barrier is a synchronization construct where a set of processes synchronizes globally i.e. each process in the set arrives at the barrier and waits for all others to arrive and then all processes leave the barrier. Let the number of processes in the set be three and S be a binary semaphore with the usual P and V functions. Consider the following C implementation of a barrier with line numbers shown on the left.

```
Void barrier(void) {
1    :    P(S)
2    :    Process_arrived++;
3    :    V (S) :
4    :    while (process_arrived'=3);
5    :    P(S);
6    :    Precess_left++;
7    :    if(process_left==3)
8    :         process_arrived=0;
9    :         process_left+0;
10   :    }
11   :    V(S);
}
```

The variable process_arrived and process_left are shared among all processes and are initialized to zero. In a concurrent program all the three processes call the barrier function when they need to synchronize globally.

Question. 34

The above implementation of barrier is incorrect. Which one of the following is true?

(A) The barrier implementation is wrong due to the use of binary semaphore S

(B) The barrier implementation may lead to a deadlock if two barrier invocations are used in immediate succession

(C) Lines 6 to 10 need not be inside a critical section

(D) The barrier implementation is correct if there are only two processes instead of three

SOLUTION

This barrier implementation is to keep track of arrival & completion of processes in system, by incrementing no. of process arrived & left. This implementation may lead to deadlock if two barrier function's invocations are used is immediate sessions.

Since $V(s)$ in first invocation at line 3 removes 1-3 from critical section bring 5-11 in critical section at line 7.

Hence (B) is correct option.

Question. 35

Which one of the following rectifies the problem in the implementation?

(A) lines 6 to 10 are simply replaced by process_arrived

(B) At the beginning of the barrier the first process to enter the barrier waits until process_arrived becomes zero before proceeding to execute P(S)

(C) Context switch is disabled at the beginning of the barrier and re-enabled at the end.

(D) The variable process_left is made private instead of shared

SOLUTION

To rectify the barrier function at the beginning of the functions the first process which inters the function waits until process arrived becomes zero before executing $P(s)$.
This removes any deadlock situation.
Hence (B) is correct option.

YEAR 2007

Question. 36

Group-1 contains some CPU scheduling algorithms and group-2 contains some applications. Match entries in Group-1 entries in Group-2

Group-1	Group-2
P. Gang Scheduling	1. Guaranteed Scheduling
Q. Rate Monotonic Scheduling	2. Real-time Scheduling
R. Fair Share scheduling	3. Thread Scheduling

(A) P-3;Q-2;R-1
(B) P-1;Q-2;R-3
(C) P-2;Q-3;R-1
(D) P-1;Q-3;R-2

SOLUTION

Rate monotonic scheduling is for real time processes, Gang scheduling is used to schedule group of program threads and fair share scheduling is a scheduling which guarantees a fair share of CPU burst to every competing process.
Hence (A) is correct option.

Question. 37

Consider the following statements about user level threads and kernel level threads. Which one of the following statements is FALSE?

(A) Context switch time is longer for kernel level threads than for user level threads
(B) User level threads do not need any hardware support
(C) Related kernal level thread can be scheduled on different processors in a multiprocessor system

(D) Blocking one kernel level thread blocks all related threads

SOLUTION

Threading a method of executing small sub processes instead of single big process. This prevents thread from blocking. So blocking of one kernel level thread doesn't block other related threads, they are unaffected.
So (D) is false.
Hence (D) is correct option.

Question. 38

An operating system uses Shortest Remaining Time first (SRT) process scheduling algorithm. Consider the arrival times and execution times for the following processes

Process	Execution time	Arrival time
P1	20	0
P2	25	15
P3	10	30
P4	15	45

What is the total waiting time for process P2?

(A) 5 (B) 15

(C) 40 (D) 55

SOLUTION

Gantt chart for the processes.

P_1	P_1	P_2	P_3	P_2	P_2	P_2	P_4	P_4

0 15 20 30 40 50 60 70

P_2 came at $t = 15$
Scheduled first time at $t = 20$ wait $= 5$.
Wait between $t = 30$ & $t = 40$ due to short remaining time of new process P_3.
Wait now $= 5 + 10 = 15$
Then complete at $t = 55$

Hence (B) is correct option.

Question. 39

A virtual memory system uses first In First Out (FIFO) page replacement policy and allocates a fixed number of frames to a process. Consider the following statements:

P: Increasing the number of page frames allocated to a process sometimes increases the page fault rate.

Q: Some program do not exhibit locality of reference.

Which one of the following is TRUE?

(A) Both P and Q are ture, and Q is the reason for P

(B) Both P and Q are true, but Q is not the reason for P

(C) P is false, but Q is true

(D) Both P and Q are false

SOLUTION

Dueto Belady's Anomaly, increasing the number of page frames allocated to a process sometimes increase the page fault rate so P is true.
Also some program do not exhibit locality of reference, so both are true but Q is not reason for P.
Hence (B) is correct option.

Question. 40

A single processor system has three resource types X, Y, and Z, which are shared by three processes. There are 5 units of each resource type. Consider the following scenario, where the column alloc denotes the number of units of each resource type allocated to each process, and the column request denotes the number of units of each resource type requested by a process in order to complete execution. Which of these processes will finish LAST?

	alloc	request
	X Y Z	X Y Z
P0	121	103
P1	201	012
P2	221	120

(A) P0

(B) P1

(C) P2

(D) None of the above, since the system is in a deadlock

SOLUTION

Initially.

Process	Allow	Request/need	Available
	XYZ	XYZ	XYZ
P_0	1 2 1	1 0 3	0 1 2
P_1	1 2 1	0 1 2	
P_2	2 2 1	1 2 0	

Here looking at available resources, only need of P_1 can be completed.
So P_1 will execute & free 2 0 1 so now available XYZ = 203.
This is need for P_0 so at second no. P_0 will execute & free XYZ = 121 so available 3 2 4 & in the end need of P_2 is fulfilled.
Hence (C) is correct option.

Question. 41

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

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

```
/*P2*/
    while (true) {
    wants2 = true;
    while (wants1 == true);
    /*    Critical
    Section*/
    wants 2 = false;
}
/*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 P_1 make wants 1 = true then P_2 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 P_1 so P_2 will wait & P_2 captures want 2 so P_1 will have to wait.

So a definite deadlock.

Hence (D) is correct option.

Data for Q. 42 & 43 are given below.

A process has been allocated 3 page frames. Assume that none of the pages of the process are available in the memory initially. The process makes the following sequence of page references (reference string): 1,2,1,3,7,4,5,6,3,1.

Question. 42

If optimal page replacement policy is used, how many page faults occur for the above reference string?

(A) 7 (B) 8

(C) 9 (D) 10

SOLUTION

Reference string 1, 2, 1, 3, 7, 4, 5, 6, 3, 1

Using optimal replacement policy, we will replace that page in memory which either will not be used or latest used in future.

			3	3	3	3	3	3	3
	2	2	2	7	4	5	6	6	6
1	1	1	1	1	1	1	1	1	1
1	2	1	3	7	4	5	6	3	1
*	*		*	*	*	*	*		

* denotes page fault
So no. of page faults are 7.
Hence (A) is correct option.

Question. 43

Least Recently Used (LRU) page replacement policy is a practical approximation to optimal page replacement. For the above reference string, how many more page faults occur with LRU than with the optimal page replacement policy?

(A) 0 (B) 1

(C) 2 (D) 3

SOLUTION

Instead of optimal policy if we use LRU then we don't have future knowledge but we replace that page in memory which has been recently used in past.
Order $\rightarrow 1, 2, 1, 3, 7, 4, 5, 6, 3, 1$

			3	3	3	5	5	5	1
	2	2	2	7	7	7	6	6	6
1	1	1	1	1	4	4	4	3	3
*	*		*	*	*	*	*	*	*
1	2	1	3	7	4	5	6	3	1

* denotes page faults.

In this case no. of page fault $= 9$

In optimal strategy $= 7$

Difference $= 9 - 7 = 2$

Hence (C) is correct option.

YEAR 2008

Question. 44

Which of the following system calls results in the sending of SYN packets?

(A) socket (B) bind

(C) listen (D) connect

SOLUTION

SYN packets are used for synchronization between sender & receiver, these packets are sent by sender during connect system call for synchronous connection.
Hence (D) is correct option.

Question. 45

The data block of a very large file in the Unix file system are allocated using

(A) Contiguous allocation

(B) Linked allocation

(C) indexed allocation

(D) an extension of indexed allocation

SOLUTION

Generally a large file system for UNIX OS use indexed allocation, but for very large systems an extension of indexed allocation i.e. ext 2, ext 3 are used.
Hence (D) is correct option.

Question. 46

The P and V operations on counting semaphores, where s is a counting ewmaphore, are defined as follows:

$P(s); s = s - 1;$

if$s < 0$ then wait;

$V(s): s = s + 1;$

if$s <= 0$ then wakeup a process waiting on s;

Assume that P_b and V_b the wait and signal operations on binary semaphores are provided. Two binary semaphores X_b and Y_b are used to implement the semaphore operations $P(s)$ and $V(s)$ as follows:

```
P(s):Pb(Xb);

     if


     }
     else
  s
     s  s
     if  s
```

The initial values of x_b and y_b are respectively

(A) 0 and 0 (B) 0 and 1

(C) 1 and 0 (D) 1 and 1

SOLUTION

X_b & Y_b are binary semaphores used to implement mutual exclusion here. So when X_b is 1 Y_b should be zero so only 1 code between the two could run.

Since $P_b(X_b)$ module implementing wait process so X_b should be 1.

$P_b(Y_b)$ implementing signal $S = S+1$

So Y_b should be 0 initially.

Hence (C) is correct option.

Question. 47

Which of the following statements about synchronous and asynchronous I/O is NOT true?

(A) An ISR is invoked on completion of I/O in synchronous I/O but not in asynchronous I/O

(B) In both synchronous and asynchronous I/O an ISR (Interrupt Serive Routine) is invoked after completion of the I/O

(C) A process making a synchronous I/O cal waits until I/O is complete, but a process making an asynchronous I/O call does not wait for completion of the I/O

(D) In the case of synchronous I/O, the process waiting for the completion of I/O is woken up by the ISR that is invoked afterr the completion of I/O

SOLUTION

For the completion of $I/0$ an interrupt should be generated for CPU in case of both synchronous & asynchronous $I/0$ and this ISR call is before them.
So CPU can switch to $I/0$.
Hence (B) is correct option.

Question. 48

Which of the following is NOT true of deadlock prevention and deadlock avoidance schemes?

(A) In deadlock prevention, the request for resources is always granted if the resulting state is safe

(B) In deadlock avoidance, the request for resources is always granted if the resulting state is safe

(C) Deadlock avoidance is less restrictive than deadlock prevention

(D) Deadlock avoidance requires knowledge of resource requirements a priori

SOLUTION

Both deadlock prevention, & avoidance allocate resources if the resulting state is safe state so option (A) & (B) are tree. The difference in both schemes is that in avoidance we know the requirement a prior. But Deadlock avoidance is less restrictive than prevention is false.
Hence (C) is correct option.

Question. 49

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.
$2^0, 2', ...2^n$ total threads.
But subtracting the parent.
Hence (B) is correct option.

Question. 50

A processor uses 36 bit physical addresses and 32 bit virtual addresses, with a page frame size of 4 Kbytes. Each page table entry is of size 4 bytes. A three level page table is used for virtual-to-physical address translation, where the virtual address is used as follows

- bits 30-31 are used to index into the first level page table,
- bits 21-29 are used to index into second level page table
- bits 12-20 are used to index into third level page table
- bits 0-11 are used as offset within the page

The number of bits required for addressing the next level page table (or page frame) in the page table entry of the first, second and third level page table are respectively.

(A) 20,20 and 20 (B) 24,24 and 24
(C) 24,24 and 20 (D) 25,25 and 24

SOLUTION

$$\begin{aligned} \text{Total address single} &= 36 \text{ bit} \\ \text{First level no. of bits} &= 2 \times 12 = 24 \text{ bits} \\ \text{Second level} &= 4 \times 6 = 24 \text{ bits} \\ \text{Third level} &= 8 \times 3 = 24 \text{ bits} \end{aligned}$$

Hence (B) is correct option.

YEAR 2009

Question. 51

Consider a system with 4 type of resources R1 (3 units), R2 (2 units), R3 (3 units), R4 (4units). A non-preemptive resource allocation policy is used. At any give instance, a request is not entertained if it cannot be completely satisfied. Three processes P1, P2, P3 request the resources as follows if executed independently.

Process P1:	Process P2:	Process P3
t=0; requests 2 units of R2 t=0; request 1 units of R3 t=0; request 2 units of R1 t=5; release 1 unit of R2 and 1 units of R1 t=7; release 1 units of R3 t=8; request 2 units of R4 t=10; Finishes	t=0 : request 2 units of R3 t=2 ; request 1 units of R4 t=4 ; request 1 units of R1 t=6 ; release 1 units of R3 t=8 ; Finishes	t=0 ; request 1 units of R4 t=2 ; request 2 units of R1 t=5 ; release 2 units of R1 t=8 ; request 1 units of R3 t=9 ; Finishes

Which one of the following statements is TRUE if all three processes run concurrently starting at time t = 0 ?

(A) All processes will finish without any deadlock

(B) Only P1 and P2 will be in deadlock

(C) Only P1 and P3 will be in deadlock

(D) All three processes will be in deadlock

SOLUTION

Total resources = 3 2 3 2

	Process 1	Process 2	Process 3	Available
Time	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4
0	0 2 0 0	0 0 2 0	0 0 0 1	3 0 1 1
1	0 2 1 0	0 0 2 0	0 0 0 1	3 0 0 1
2	0 2 1 0	0 0 2 1	2 0 0 1	1 0 0 0
3	0 2 1 0	0 0 2 1	2 0 0 1	1 0 0 0
4	0 2 1 0	1 0 2 1	2 0 0 1	0 0 0 0
5	1 1 1 0	1 0 2 1	0 0 0 1	1 1 0 0
6	1 1 1 0	1 0 1 1	0 0 0 1	1 1 1 0
7	1 1 0 0	1 0 1 1	0 1 0 1	1 0 2 0
8	1 1 0 0	0 0 0 0	0 1 1 1	2 0 2 1
9	1 1 0 2	0 0 0 0	0 0 0 0	2 1 3 0
10	0 0 0 0	0 0 0 0	0 0 0 0	3 2 3 2

All process computed without deadlock

Hence (A) is correct option.

Question. 52

In which of the following page replacement policies, Belady's anomaly may occur ?

(A) FIFO
(B) Optimal
(C) LRU
(D) MRU

SOLUTION

Belady's anomaly is a properly of FIFO page replacement policy in which the no. of page faults increases even if the no. of page frames are increased in some cases.
Hence (A) is correct option.

Question. 53

The essential content(S) in each entry of a page table is/are

(A) virtual page number
(B) page frame number
(C) Both virtual page number and page frame number
(D) access right information

SOLUTION

A page table is in main memory which mop a page frame to a virtual memory page.
So essential content is page frame number
Hence (B) is correct option.

Question. 54

Consider a disk system with 100 cylinders. The requests to access the cylinders occur in following sequence :

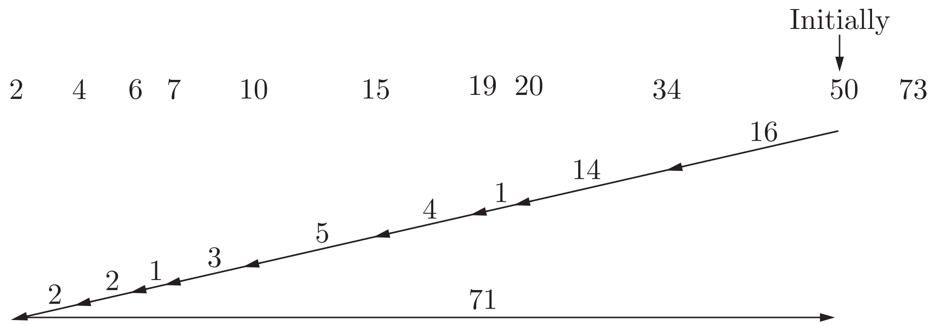
4, 34, 10, 7, 19, 73, 2, 15, 6, 20

Assuming that the head is currently at cylinder 50, what is the time taken to satisfy all requests if it takes 1 ms to move from one cylinder to adjacent one and shortest seek time first policy is used ?

(A) 95 ms
(B) 119 ms
(C) 233 ms
(D) 276 ms

SOLUTION

Arranging in order.

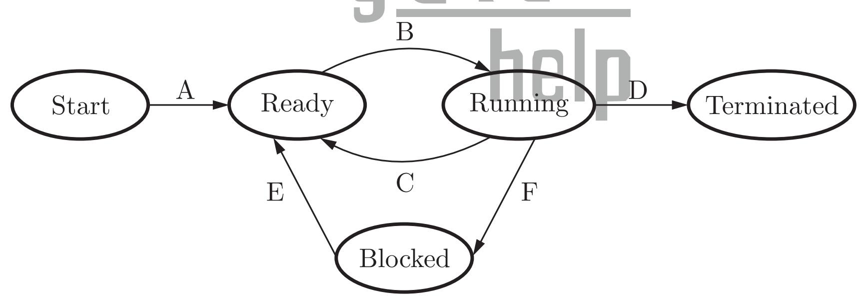


$$\text{Total} = 16 + 14 + 1 + 4 + 5 + 3 + 1 + 2 + 2 + 71$$
$$= 119 \text{ ms}$$

Hence (B) is correct option.

Question. 55

In the following process state transition diagram for a uniprocessor system, assume that there are always some processes in the steady state :



Now consider the following statements :

I. If a process makes a transition D, it would result in another process making transition A immediately

II A process P_2 in blocked state can make transition E while another process P_1 is in running state

III The OS uses non-premptive scheduling

IV The OS uses non-premptive scheduling

Which of the above statement are TRUE ?

(A) I and II

(B) I and III

(C) II and III

(D) II and IV

SOLUTION

I Is not necessary that a new process enters immediately.

II Is true different process can be in different states one is ready and another in running state.

III Is also correct since if one blocked it stop running so other can run.

IV Is not correct.

Hence (C) is correct option.

Question. 56

The enter_CS ( ) and leave_CS ( ) functions to implement critical section of a process are realized using test and set instruction as follows :

```
Void enter_cs (X)
{
     while (test-and-set)(X)) :
}
Void leave_CS(X)
{
     X=0;
}
```


In the above solution, X is a memory location associated with the CS and is initialized to 0. Now consider the following statements

I The above solution to CS problem is deadlock-free

II The solution is starvation free

III The processes enter CS in FIFO order

IV More than one process can enter CS at the same time

Which of the above statements are TRUE

(A) I only
(B) I and II
(C) II and III
(D) IV only

SOLUTION

I Is true since X is initialized to zero, this cause only 1 process to enter into critical so no deadlock.

II & III Are false, other process which doesn't capture X will wait so ultimately starve.

IV Since I true so it is false.
Hence (A) is correct option.

Question. 57

A multilevel page table is preferred in comparison to a single level page table for translating virtual address to physical address because

(A) It reduces the memory access time to read or write and memory location

(B) It helps to reduce the size of page table needed to implement the virtual address space of a process

(C) If is required by the translation lookaside buffer

(D) If helps to reduce the number of page faults in page replacement algorithms.

SOLUTION

Single level page table is not preferred since it requires the no. of page table enteries equal to the no. of virtual memory addresses, but a multilevel page table has smaller no. of enteries so reduce the size of page table needed.
Hence (B) is correct option.

YEAR 2010

Question. 58

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 S2=not (S1)

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

SOLUTION

Method used by P_1 & P_2 enters into critical section when $S_1 = S_2$ & $S_1 ! = S_2$ respectively, since both are opposite conditions so mutual exclusion is there, but if P_1's while loop true then it will always be true & in P_2 if while loop true it would run infinitely so no. progress. Hence (A) is correct option.

Question. 59

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 accesses the same 100 pages but now in the reverse order .How many page faults will occur ?

(A) 196 (B) 192

(C) 197 (D) 195

SOLUTION

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 & 100^{th} 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.

Question. 60

Which of the following statements are true ?

I Shortest remaining time first scheduling may cause starvation

II Preemptive scheduling may cause starvation

II Round robin in better than FCFS in terms of response time

(A) I only (B) I and III only

(C) II and III only (D) I, II and III

SOLUTION

I SRIF scheduling may cause starvation since the processes which require large CPU burst periods would have to wait.

II Generally preemptive scheduling doesn't cause starvation but in some cases, it may cause starvation when one process doesn't block due to I/O so others have to wait.

III It is quite obvious that due to preemption involvement round orbit is better than FCFS, in case of response time.

All are true
Hence (D) is correct option.

Question. 61

The following program consists of 3 concurrent precesses and 3 binary semaphores. The semaphores are initialized as $S_0 = 1, S_1 = 0, S_2 = 0$

Process P_0	Process P_1	Process P_2

While (true){
wait (S_0);
print '0'
release (S_1);
release (S_2);
}
	wait (S_1);
	release (S_0);
		wait (S_2)
		release (S_0);

How many times will precess P0 print '0' ?

(A) At least twice
(B) Exactly twice
(C) Exactly thrice
(D) Exactly once

SOLUTION

Let us see what will happen initially $S_0 = 1$, $S_1 = 0$ $S_2 = 0$ so P_1 & P_2 will wait & P_0 runs & make $S_0 = 0$ wait decrements semaphore by 1 & release increments if by 1.
So after P_0's 1^{st} run $S_1 = 1$ & $S_2 = 1$ '0' is printed
Now P_1 & P_2 can run & make $S_1 = 0$, $S_2 = 0$ but both increments $S_0 = 1$ again.
So P_0 again starts and print '0' so this would continue again & again.
So '0' will be printed at least twice.
Hence (A) is correct option.

Question. 62

A system has n resources $R_0..R_{n-1}$, and k processes $P_0..P_{k-1}$.The implementation of the resource request logic of each process P_i, is as follows:

```
if (i%2==0){
      if(i<n)request i;
      if(i+2<n)request  i+2;
}
else {
      if (i<n)request  n  ;
      if (i+2<n)request n i 2;
}
```

In which one of the following situations is a deadlock possible ?

(A) $n = 40, k = 26$

(B) $n = 21, k = 12$

(C) $n = 20, k = 10$

(D) $n = 41, k = 19$

SOLUTION

Allotment of resources would be successful to all cases but in option

(B) where $n = 21$ & $k = 12$.

Let use take $i = 10$

So for P_{10} only R_{10} can be allotted whereas for P_{11} odd case (else).

R_{21-11} i.e R_{10} can only be allotted.

This is an conflict.

Hence (B) is correct option.

GATE Multiple Choice Questions
For Computer Science

By NODIA and Company

Available in Two Volumes

FEATURES

- The book is categorized into units and the units are sub-divided into chapters.
- Chapter organization for each unit is very constructive and covers the complete syllabus
- Each chapter contains an average of 40 questions
- The questions are standardized to the level of GATE examination
- Solutions are well-explained, tricky and consume less time. Solutions are presented in such a way that it enhances you fundamentals and problem solving skills
- There are a variety of problems on each topic
- Engineering Mathematics is also included in the book