Question 1 [5 Marks]

A shared variable x, initialized to zero, is operated on by four concurrent processes W, X, Y, Z as follows. Each of the processes W and X read x from memory, increments by one, stores it to memory, and then terminates. Each of the processes Y and Z reads x from memory, decrements by two, stores it to memory, and then terminates. Each process before reading x invokes the P operation (i.e., wait) on a counting semaphore S and invokes the V operation (i.e., signal) on the semaphore S after storing x to memory. Semaphore S is initialized to two. What is the maximum possible value of x after all processes complete execution?

A -2

B -1

C 1

✓ 2

Explanation

Processes can run in many ways possible. Below mentioned is an order which lets *x* attain the maximum value possible:

Problems Courses Get Hired Practice
Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Overview Learn Quiz Contest
LIVE BATCHES

Question 2 [5 Marks]

A process executes the code as given below:

```
fork();
fork();
fork();
```

The number of child processes created:

A 3

B 4

✓ 7

D 8

Explanation

Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

LIVE BATCHES

Question 3 [5 Marks]

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


while (S1 == S2);

//Critical Section

S1 = S2;
```

- Mutual Exclusion but not Progress
- B Progress but not Mutual Exclusion
- C Neither Mutual Exclusion nor Progress
- D Both Mutual exclusion and Progress

Problems Courses Get Hired Practice

Overview Learn Quiz Contest

Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

LIVE BATCHES

Question 4 [5 Marks]

If the time-slice used in the round-robin scheduling policy is more than the maximum time required to execute any process, then the policy will:

- A degenerate to the Shortest-Job-First algorithm
- B degenerate to the Priority Scheduling algorithm
- degenerate to the First-Come-First-Serve algorithm
- D none of the above

Explanation

Round-robin algorithm executes processes in an FCFS manner with the additional condition of executing each of them in a time-slice. If the time-quantum is kept higher than the largest time taken by any process to complete execution. Then all processes will execute without pre-emption and that too in an FCFS fashion. Thus, it will degenerate to FCFS.

Question 5 [5 Marks]

The following are some events that occur after a device controller issues an interrupt while process L is under execution.

- **(P)** The processor pushes the process status of L onto the control stack.
- **(Q)** The processor finishes the execution of the current instruction.
- **(R)** The processor executes the interrupt service routine.
- **(S)** The processor pops the process status of L from the control stack.
- **(T)** The processor loads the new PC value based on the interrupt.

Which of the following is the correct order in which the events above occur?

QPTRS

B PTRSQ

C TRPQS

D QTPRS

Explanation

Problems Courses Get Hired
Practice
Overview Learn Quiz Contest
Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
LIVE BATCHES

Question 6 [5 Marks]

A starvation free job scheduling policy guarantees that no job indefinitely waits for a service. Which of the following job scheduling policies is starvation free?

A Priority Scheduling

B Shortest Job First

C Youngest Job First

Round Robin

Explanation

Round Robin is a starvation free scheduling algorithm as it imposes a strict time bound on the response time of each process i.e. for a system with 'n' processes running in a round robin system with time quanta t_q, no process will wait for more than (n-1) t_q time units to get its CPU turn. Option (D) is correct.

Problems Courses Get Hired

Practice

Overview Learn Quiz Contest

Back To Course

Learn

Classroom

Theory

Quiz

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

LIVE BATCHES

Question 7 [5 Marks]

Which of the following need not necessarily be saved on a context switch between processes?

A General Purpose Registers

Translation look-aside Buffer

C Program Counter

D All of the above

Explanation

Translation look-aside Buffer is a kind of a Cache that is just used to improve performance (by reading data directly from the cache and not accessing memory). It has no significance after a context-switch. However, Register values & Program Counter indicate the current state of the executing process and must be saved so as to continue after execution resumes. Thus TLB values need not be saved.

Problems Courses Get Hired

Practice

Overview Learn Quiz Contest

Back To Course

Learn

Classroom

Theory

Quiz

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

LIVE BATCHES

Question 8 [5 Marks]

What is the output of the following program?

```
#include <stdio.h>
#include <unistd.h>
int main()
{
   int a = 10;
   if ((fork() == 0))
      a++;
   printf ("%d\n", a);
   return 0;
}
```

10 and 11

B 10

C 11

D 11 and 11

Question 9 [5 Marks]

Here are the two concurrent process P1, P2 with respective codes.
The code for P1 is as:

```
while (true) {
    A :____;
    printf("%d", 1);
    printf("%d", 1);
    B:____;
}
```

The code for P2 is as:

A = P(N)
B = V(M)
C = P(M)
D = V(N)
M = 0
N = 1

B
A = P(N)
B = V(M)
C = P(M)

Problems Courses Get Hired
Practice
Overview Learn Quiz Contest
Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
LIVE BATCHES

Question 10 [5 Marks]

Suppose n processes, $P_1, P_2, \ldots, P_n$ 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_i, where $S_i > 0$. Which one of the following is a sufficient condition for ensuring that deadlock does not occur?

(a) $\forall i, s_i < m$ (b) $\forall i, s_i < n$

(c) $\sum_{i=1}^{n} s_i < (m+n)$ (d) $\sum_{i=1}^{n} s_i < (m*n)$

A
A

B
B

C

D
D

Explanation
In the extreme condition, all processes acquire (S_i - 1) resources and need 1 more resource. So the

Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Question 11 [5 Marks]

Consider the 3 processes, P1, P2 and P3 shown in the table.

Process	Arrival time	Time Units Required
P1	0	5
P2	1	7
P3	3	4

The completion order of the 3 processes under the policies FCFS and RR2 (Round-Robin scheduling with CPU time-quantum of 2 units) are:

A FCFS: P1, P2, P3
RR: P1, P2, P3

B FCFS: P1, P3, P2
RR: P1, P3, P2

FCFS: P1, P2, P3
RR: P1, P3, P2

D FCFS: P1, P3, P2
RR: P1, P2, P3

Problems Courses Get Hired
Practice
Overview Learn Quiz Contest
Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
LIVE BATCHES

Question 12 [5 Marks]

Consider the following table of arrival time and burst time for three processes P0, P1 and P2.

Process	Arrival time	Burst Time
P0	0 ms	9 ms
P1	1 ms	4 ms
P2	2 ms	9 ms

The pre-emptive Shortest-Job-First scheduling algorithm is used. Scheduling is carried out only at the arrival or completion of processes. What is the average waiting time for the three processes (in ms)?

5.0

B 4.33

C 6.33

D 7.33

Question 13 [5 Marks]

Which of the following statements are true?

1. Shortest-Remaining-Time-First scheduling may cause starvation.
2. Preemptive-Scheduling may cause starvation.
3. Round-Robin is better than FCFS in terms of response time.

A. I only

B. I and III only

C. II and III only

✓ I, II and III

Explanation

Regarding the following statements:

1. SRTF scheduling is a preemptive algorithm which after each instance (each time unit passage) finds out the shortest-remaining-time job. This can cause starvation for some process having a

Problems Courses Get Hired
Practice
Overview Learn Quiz Contest
Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
LIVE BATCHES

Question 14 [5 Marks]

Consider a uniprocessor system executing three tasks T1, T2 and T3, each of which is composed of an infinite sequence of jobs (or instances) which arrive periodically at intervals of 3, 7 and 20 milliseconds, respectively. The priority of each task is the inverse of its period and the available tasks are scheduled in order of priority, with the highest priority task scheduled first. Each instance of T1, T2 and T3 requires an execution time of 1, 2 and 4 milliseconds, respectively. Given that all tasks initially arrive at the beginning of the 1st milliseconds and task preemptions are allowed, the first instance of T3 completes its execution at the end of _____________ milliseconds.

A. 5

B. 10

✓ 12

D. 15

Explanation

Periods of T1, T2, and T3 are 3ms, 7ms, and 20ms.

Problems Courses Get Hired Practice LIVE BATCHES

Back To Course

Learn
Classroom
Theory

Quiz

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Overview Learn Quiz Contest

Question 15 [5 Marks]

Three concurrent processes X, Y, and Z execute three different code segments that access and update certain shared variables. Process X executes the P operation (i.e., wait) on semaphores a, b and c; process Y executes the P operation on semaphores b, c, and d; process Z executes the P operation on semaphores c, d, and a before entering the respective code segments. After completing the execution of its code segment, each process invokes the V operation (i.e., signal) on its three semaphores. All semaphores are binary semaphores initialized to one. Which one of the following represents a deadlock-free order of invoking the P operations by the processes?

A
X: P(a)P(b)P(c)
Y:P(b)P(c)P(d)
Z:P(c)P(d)P(a)

✓
X: P(b)P(a)P(c)
Y:P(b)P(c)P(d)
Z:P(a)P(c)P(d)

C
X: P(b)P(a)P(c)
Y:P(c)P(b)P(d)
Z:P(a)P(c)P(d)

D
X: P(a)P(b)P(c)
Y:P(c)P(b)P(d)
Z:P(c)P(d)P(a)

Problems Courses Get Hired Practice LIVE BATCHES

Back To Course

Learn
Classroom
Theory

Quiz

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Overview Learn Quiz Contest

Question 16 [5 Marks]

Two shared resources R_1 and R_2 are used by processes P_1 and P_2. Each process has a certain priority for accessing each resource. Let T_{ij} denote the priority of P_i for accessing R_j. A process P_i can snatch a resource R_h from process P_j if T_{ik} is greater than T_{jk}. Given the following:

1. $T_{11} > T_{21}$
2. $T_{12} > T_{22}$
3. $T_{11} < T_{21}$
4. $T_{12} < T_{22}$

Which of the following conditions ensures that P_1 and P_2 can never deadlock?

A (I) and (IV)

B (II) and (III)

✓ (I) and (II)

D None of the Above

Explanation

Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Overview Learn Quiz Contest
LIVE BATCHES

Question 17 [5 Marks]

A system shares 9 tape drives. The current allocation and maximum requirement of tape drives for 4 processes are shown below:

Process	Maximum need	Current allocation
P1	9	3
P2	6	1
P3	5	3
P4	10	0

Which of the following best describes the current state of the system?

A Safe, Deadlocked

Safe, Not Deadlocked

Not Safe, Deadlocked

D Not Safe, Not Deadlocked

Explanation

Problems Courses Get Hired
Practice

Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Overview Learn Quiz Contest
LIVE BATCHES

Question 18 [5 Marks]

Consider a system having "n" resources of the same type. These resources are shared by 3 processes: A, B, C. These have peak demands of 3, 4, and 6 respectively. For what value of "n" deadlock won't occur?

A 15

B 9

C 10

11

Explanation

The value of n for which the deadlock can't occur = n(k-1) + 1 Number of min resources required = (3-1) + (4-1) + (6-1) + 1 = 11 So, option (D) is correct.

Your submitted response was correct.

Back To Course

Learn
Classroom
Theory

Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Overview Learn Quiz Contest

Question 19 [5 Marks]

An operating system uses the Banker's algorithm for deadlock avoidance when managing the allocation of three resource types X, Y, and Z to three processes P0, P1, and P2. The table given below presents the current system state. Here, the Allocation matrix shows the current number of resources of each type allocated to each process and the Max matrix shows the maximum number of resources of each type required by each process during its execution.

	Allocation			Max		
	X	Y	Z	X	Y	Z
P0	0	0	1	8	4	3
P1	3	2	0	6	2	0
P2	2	1	1	3	3	3

There are 3 units of type X, 2 units of type Y and 2 units of type Z still available. The system is currently in a safe state. Consider the following independent requests for additional resources in the current state:

A. Only REQ1 can be permitted.

✓ Only REQ2 can be permitted.

C. Both REQ1 and REQ2 can be permitted.

D. Neither REQ1 nor REQ2 can be permitted.

Problems Courses Get Hired
Practice

LIVE BATCHES

Back To Course

Learn
Classroom
Theory

Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Overview Learn Quiz Contest

Question 20 [5 Marks]

In a system, there are three types of resources: E, F, and G. Four processes P0, P1, P2, and P3 execute concurrently. At the outset, the processes have declared their maximum resource requirements using a matrix named Max as given below. For example, Max[P2, F] is the maximum number of instances of F that P2 would require. The number of instances of the resources allocated to the various processes at any given state is given by a matrix named Allocation. Consider a state of the system with the Allocation matrix as shown below, and in which 3 instances of E and 3 instances of F are the only resources available.

Allocation

	E	F	G
P_0	1	0	1
P_1	1	1	2
P_2	1	0	3
P_3	2	0	0

Max

	E	F	G
P_0	4	3	1
P_1	2	1	4
P_2	1	3	3
P_3	5	4	1

✓ The system is in a safe state.

B. The system is not in a safe state but would be safe if one more instance of E were available.

C. The system is not in a safe state but would be safe if one more instance of F were available.

D. The system is not in a safe state but would be safe if one more instance of G were available

Question 21 [5 Marks]

Consider the virtual page reference string:
[1, 2, 3, 2, 4, 1, 3, 2, 4, 1]
On a demand-paged virtual memory system running on a computer system that main memory size of 3 pages frames which are initially empty. Let **LRU**, **FIFO**, and **OPTIMAL** denote the number of page faults under the corresponding page replacements policy. Then:

A. OPTIMAL < LRU < FIFO

✓ OPTIMAL < FIFO < LRU

C. OPTIMAL = LRU

D. OPTIMAL = FIFO

Explanation

In all the diagrams given, digits marked as red indicate page fault.

First In First Out (FIFO): In this algorithm, the page which was referred to the earliest will also be the first to get replaced. The OS keeps a queue for storing the pages referred. Pages are inserted onto

Problems Courses Get Hired
Practice

Overview Learn Quiz Contest

Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

LIVE BATCHES

Question 22 [5 Marks]

A system uses FIFO policy for page replacement. It has a total of 4 page frames with no pages loaded initially. 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?

✓ 196

B. 192

C. 197

D. 195

Explanation

Access to 100 pages will cause a total of 100 page faults. When these pages are accessed in reverse order, the first 4 accesses will not cause a page fault. All other access to pages will cause page faults. So the total number of page faults will be 100 + 96.

Question 23 [5 Marks]

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 the second level page table • Bits 12-20 are used to index into the third level page table, and • 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 tables are respectively.

A. 20, 20 and 20

B. 24, 24 and 24

C. 24, 24 and 20

D. 25, 25 and 24 ✓

Explanation

Virtual address size is given as 32-bits.
Physical address size is given as 36-bits. Thus, the physical memory size is 2^{36} bytes.

Problems Courses Get Hired
Practice
Overview Learn Quiz Contest
Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
LIVE BATCHES

Question 24 [5 Marks]

Which of the following is **NOT** an advantage of using shared, dynamically linked libraries as opposed to using statically linked libraries?

A. Smaller sizes of executable files.

B. Lesser overall page fault rate in the system.

C. Faster program startup. ✓

D. Existing programs need not be re-linked to take advantage of newer versions of libraries.

Explanation

Dynamically linked libraries is a linking scheme in which libraries required by a program are fetched from the disk on demand, and thus they are not part of the final executable. This creates a great advantage in shipping non-bloat executables. i.e. The final size of the executable is small. A DLL may be required ny multiple programs at a time. So, once we fetch and keep it in the main memory, it can be used by several programs, causing less page-faults overall. Also, there is an added

Question 25 [5 Marks]

The process of assigning load addresses to the various parts of the program and adjusting the code and data in the program to reflect the assigned addresses is called:

A Assembly

B Parsing

C Relocation

D Symbol Resolution

Explanation

Relocation of code is the process done by the linker-loader when a program is copied from external storage into main memory.

A linker relocates the code by searching files and libraries to replace symbolic references of libraries with actually usable addresses in memory before running a program.

Thus, option (C) is the answer.

Problems Courses Get Hired
Practice

Back To Course

Learn
Classroom
Theory

Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Overview Learn Quiz Contest

LIVE BATCHES

Question 26 [5 Marks]

Where does the swap space reside?

A RAM

B Disk

C ROM

D On-chip cache

Explanation

Swap space is an area on disk that temporarily holds a process memory image. When memory is full and process needs memory, inactive parts of the process are put in swap space of the disk.

Your submitted response was correct.

Problems Courses Get Hired

Practice

LIVE BATCHES

Back To Course

Learn

Classroom

Theory

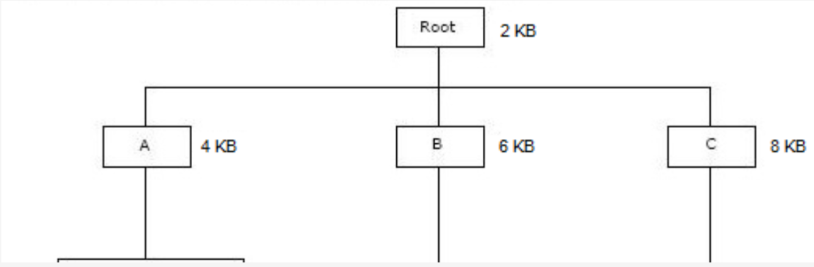
Quiz

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Overview Learn Quiz Contest

Question 27 [5 Marks]

The overlay tree for a program is as shown below:



A 12 KB

14 KB

C 10 KB

8 KB

Problems Courses Get Hired

Practice

LIVE BATCHES

Back To Course

Learn

Classroom

Theory

Quiz

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Overview Learn Quiz Contest

Question 28 [5 Marks]

Thrashing:

A reduces page I/O

B decreases the degree of multiprogramming

implies excessive page I/O

D improves system performance

Explanation

Due to thrashing, excessive page-faults occur, because of which most of the useful time is wasted in swapping pages with the disk ~ excessive page I/O, or option (C).

Your submitted response was correct.

Question 29 [5 Marks]

Which of the following system call is used for opening or creating a file?

A read

B create

C open

D close

Explanation
We generally use ***fopen()*** function to create a file in the disk (if not present already). However, under the hood, fopen calls the ***open*** system call to perform the I/O instruction.

Your submitted response was correct.

Problems Courses Get Hired
Practice
LIVE BATCHES

Back To Course
Learn
Classroom
Theory
Quiz
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Overview Learn Quiz Contest

Question 30 [5 Marks]

Which system call is used to reap child processes?

A exec

B reap

C wait

D collect

Explanation
wait system call is used to reap up any child process. It is a blocking call that halts the parent to collect the exit status of all its child processes.

Your submitted response was correct.