send ur account details

Topic - Operating system

Q1: Which of the following is NOT a valid deadlock prevention scheme?

1.Release all resources before requesting a new resource

2.Number the resources uniquely and never request a lower numbered resource than the last one requested.

3.**Never request a resource after releasing any source**

4.Request and all required resources be allocated before execution

Q2: Let m[0]…m[4] be mutexes (binary semaphores) and P[0] …. P[4] be processes.

Suppose each process P[i] executes the following:

```
 wait (m[i]); wait(m[(i+1) mode 4]);
  ------
     release (m[i]); release (m[(i+1)mod 4]);
```

This could cause

(a) Thrashing

(b) Deadlock

(c) Starvation, but not deadlock

(d) None of the above

>>You can easily see a deadlock in a situation where..

P[0] has acquired m[0] and waiting for m[1]

P[1] has acquired m[1] and waiting for m[2]

P[2] has acquired m[2] and waiting for m[3]

P[3] has acquired m[3] and waiting for m[0]

Q3: A graphics card has on board memory of 1 MB. Which of the following modes can the card not support?

(a) 1600 x 400 resolution with 256 colours on a 17 inch monitor.

(b) 1600 x 400 resolution with 16 million colours on a 14 inch monitor.

(c) 800 x 400 resolution with 16 million colours on a 17 inch monitor.

(d) 800 x 800 resolution with 256 colours on a 14 inch monitor.

>>Monitor size doesn’t matter here. So, we can easily deduce that answer should be (b) as this has the highest memory requirements. Let us verify it.
Number of bits required to store a 16M colors pixel = ceil(log2(16*1000000)) = 24
Number of bytes required for 1600 x 400 resolution with 16M colors = (1600 * 400 * 24)/8 which is 192000000 (greater than 1MB).

Q4: 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

Q5: Which of the following requires a device driver?

(a) Register

(b) Cache

(c) Main memory

(d) Disk

Q6: Suppose the time to service a page fault is on the average 10 milliseconds, while a memory access takes 1 microsecond. Then a 99.99% hit ratio results in average memory access time of

(a) 1.9999 milliseconds

(b) 1 millisecond

(c) 9.999 microseconds

(d) 1.9999 microseconds

>>Average memory access time =

[(% of page miss)*(time to service a page fault) +

(% of page hit)*(memory access time)]/100

Q7: Suppose the time to service a page fault is on the average 10 milliseconds, while a memory access takes 1 microsecond. Then a 99.99% hit ratio results in average memory access time of

(a) General purpose registers

(b) **Translation look-aside buffer**

(c) Program counter

(d) All of the above

>>In a process context switch, the state of the first process must be saved somehow, so that, when the scheduler gets back to the execution of the first process, it can restore this state and continue. The state of the process includes all the registers that the process may be using, especially the program counter, plus any other operating system specific data that may be necessary.
A Translation lookaside buffer (TLB) is a CPU cache that memory management hardware uses to improve virtual address translation speed. A TLB has a fixed number of slots that contain page table entries, which map virtual addresses to physical addresses. On a context switch, some TLB entries can become invalid, since the virtual-to-physical mapping is different. The simplest strategy to deal with this is to completely flush the TLB.

Q8: Where does the swap space reside ?

(a) RAM

(b) Disk

(c) ROM

(d) On-chip cache

>>Swap space is an area on disk that temporarily holds a process memory image. When physical memory demand is sufficiently low, process memory images are brought back into physical memory from the swap area. Having sufficient swap space enables the system to keep some physical memory free at all times.

Q9: Which of the following does not interrupt a running process?

(a) A device

(b) Timer

(c) Scheduler process

(d) Power failure

Scheduler process doesn’t interrupt any process, it’s Job is to select the processes for following three purposes.

Long-term scheduler(or job scheduler) –selects which processes should be brought into the ready queue

Short-term scheduler(or CPU scheduler) –selects which process should be executed next and allocates CPU.

Mid-term Scheduler (Swapper)- present in all systems with virtual memory, temporarily removes processes from main memory and places them on secondary memory (such as a disk drive) or vice versa. The mid-term scheduler may decide to swap out a process which has not been active for some time, or a process which has a low priority, or a process which is page faulting frequently, or a process which is taking up a large amount of memory in order to free up main memory for other processes, swapping the process back in later when more memory is available, or when the process has been unblocked and is no longer waiting for a resource.

Q10: 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

Q11: 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

>>If block size is large then seek time is less (fewer blocks to seek) and disk performance is improved, but remember larger block size also causes waste of disk space.

Q12: Consider the following statements with respect to user-level threads and kernel supported threads

i. context switch 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

Q13: 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

>>Each process needs minimum number of pages based on instruction set architecture. Example IBM 370: 6 pages to handle MVC (storage to storage move) instruction
Instruction is 6 bytes, might span 2 pages.
2 pages to handle from.
2 pages to handle to

Q14: 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

>>Size of page table =

(total number of page table entries) *(size of a page table entry)

Number of entries in page table =

(virtual address space size)/(page size)

= (2^32)/(2^10)

= 2^22

Note that page table entry also holds auxiliary information about the page such

as a present bit, a dirty or modified bit, address space or process ID information,

amongst others. So size of page table

> (total number of page table entries) *(size of a page table entry)

> (2^22 *19) bytes

> 9.5 MB

Q15: A file system with 300 GByte uses a file descriptor with 8 direct block address. 1 indirect block address and 1 doubly indirect block address. The size of each disk block is 128 Bytes and the size of each disk block address is 8 Bytes. The maximum possible file size in this file system is

(a) 3 Kbytes

(b) 35 Kbytes

(c) 280 Bytes

(d) Dependent on the size of the disk

>>Total number of possible addresses stored in a disk block = 128/8 = 16
Maximum number of addressable bytes due to direct address block = 8*128
Maximum number of addressable bytes due to 1 single indirect address block = 16*128
Maximum number of addressable bytes due to 1 double indirect address block = 16*16*128
The maximum possible file size = 8*128 + 16*128 + 16*16*128 = 35KB

Q 16. Consider four processes P, Q, R, and S scheduled on a CPU as per round-robin algorithm with a time quantum of 4 units. The processes arrive in the order P, Q, R, S, all at time t = 0. There is exactly one context switch from S to Q, exactly one context switch from R to Q, and exactly two context switches from Q to R. There is no context switch from S to P. Switching to a ready process after the termination of another process is also considered a context switch. Which one of the following is NOT possible as CPU burst time (in time units) of these processes?

(A)P = 4, Q = 10, R = 6, S = 2

(B)P = 2, Q = 9, R = 5, S = 1

(C)P = 4, Q = 12, R = 5, S = 4

(D)P = 3, Q = 7, R = 7, S = 3

Q 17. Consider the following statements about process state transitions for a system using preemptive scheduling.

I. A running process can move to ready state.

II. A ready process can move to ready state.

III. A blocked process can move to running state.

IV. A blocked process can move to ready state.

Which of the above statements are TRUE?

>> Options I, II and IV

Q18.The following C program is executed on a Unix/Linux system:

```
    #include < unistd.h >
      int main ()
      {
           int i ;
           for (i=0; i<10; i++)
                if (i%2 == 0) fork ( ) ;
           return 0 ;
      }
```

>>2^5-1=31

Q 19.Consider an arbitrary set of CPU-bound processes with unequal CPU burst lengths submitted at the same time to a computer system. Which one of the following process scheduling algorithms would minimize the average waiting time in the ready queue?

A >Shortest remaining time first

B>Round-robin with the time quantum less than the shortest CPU burst

C>Uniform random

D>Highest priority first with priority proportional to CPU burst length

Q 20.The maximum number of processes that can be in Ready state for a computer system with n CPUs is

(A) n

(B) n2

(C) 2n

(D) Independent of n

Q 21.

Which one of the following is FALSE?

A) User level threads are not scheduled by the kernel.

B) When a user level thread is blocked, all other threads of its process are blocked.

C) Context switching between user level threads is faster than context switching between kernel level threads.

D) Kernel level threads cannot share the code segment.

Q 22.A scheduling algorithm assigns priority proportional to the waiting time T of a process. Every process starts with zero (the lowest priority). The scheduler re-evaluates the process priorities every

time units and decides the next process to schedule. Which one of the following is TRUE if the processes have no I/O operations and all arrive at time zero?

A) This algorithm is equivalent to the first-come-first-serve algorithm.

B) This algorithm is equivalent to the round-robin algorithm.

C) This algorithm is equivalent to the shortest-job-first algorithm.

D) This algorithm is equivalent to the shortest-remaining-time-first algorithm.

Q23.Consider the following statements about user level threads and kernel level threads. Which one of the following statement 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 kernel level threads can be scheduled on different processors in a multiprocessor system.

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

>>Kernel level threads are managed by the OS, therefore, thread operations are implemented in the kernel code. Kernel level threads can also utilize multiprocessor systems by splitting threads on different processors. If one thread blocks it does not cause the entire process to block.

Q 24.A process executes the code

```
fork ();

fork ();

fork ();
```

The total number of child processes created is?

>> 7

Q 25.A Computer handles several interrupt sources of which the following are relevant for this question:

A> Interrupt from cpu temperature sensor (raises interrupt if cpu temperature is too high)

B>Interrupt from Mouse (raises interrupt if the mouse is moved or a button is pressed)

C>Interrupt from keyboard (raises interrupt when a key is pressed or release)

D>Interrupt from Hard Disk (raises interrupt when a disk read is completed)

Which one these will be handled at the HIGHEST priority?

Q 26. The time taken to switch between user and kernel modes of execution be t1 while the time taken to switch between two processes be t2.

Which of the following is TRUE?

(A) t1 > t2

(B) t1 = t2

(C) t1 < t2

(D) nothing can be said about the relation between t1 and t2

Q 27.A thread is usually defined as a “light weight process” because an operating system (OS) maintains smaller data structures for a thread than for a process. In relation to this, which of the following is TRUE?

(A) On per-thread basis, the OS maintains only CPU register state

(B) The OS does not maintain a separate stack for each thread

(C) On per-thread basis, the OS does not maintain virtual memory state

(D) On per-thread basis, the OS maintains only scheduling and accounting information

Q 28.Which of the following statements are true?

I. Shortest remaining time first scheduling may cause starvation
II. Preemptive scheduling may cause starvation
III. Round robin is 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

Q29.A CPU generally handles an interrupt by executing an interrupt service routine

(A) As soon as an interrupt is raised

(B) By checking the interrupt register at the end of fetch cycle.

(C) By checking the interrupt register after finishing the execution of the current instruction.

(D) By checking the interrupt register at fixed time intervals.

>>Hardware detects interrupt immediately, but CPU acts only after its current instruction. This is followed to ensure integrity of instructions.

Q30: Group 1 contains some CPU scheduling algorithms and Group 2 contains some applications. Match entries in Group 1 to entries in Group 2.

Group I	Group II
(P) Gang Scheduling	(1) Guaranteed Scheduling
(Q) Rate Monotonic Scheduling	(2) Real-time Scheduling
(R) Fair Share Scheduling	(3) Thread Scheduling

P – 3

Q – 2

R – 1

Q31: 1) 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

Q32: Consider the following statements with respect to user-level threads and kernel-supported threads.

i) Context switch 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) Use-level threads are transparent to the kernel

Which of the above statements are true?

Q33: 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

Q34: A CPU has two modes-privileged and non-privileged. In order to change the mode from privileged to non-privileged

(A) a hardware interrupt is needed

(B) a software interrupt is needed

(C) a privileged instruction (which does not generate an interrupt) is needed

(D) a non-privileged instruction (which does not generate an interrupt is needed

Q35: Which of the following does not interrupt a running process?

(A) A device

(B) Timer

(C) Scheduler process

(D) Power failure

Q36: Consider a set of n tasks with known runtimes r1, r2, …. rn 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

>>Throughput means total number of tasks executed per unit time i.e. sum of waiting time and burst time.
Shortest job first scheduling is a scheduling policy that selects the waiting process with the smallest execution time to execute next.
Thus, in shortest job first scheduling, shortest jobs are executed first. This means CPU utilization is maximum. So, maximum number of tasks are completed.

Q37: System calls are usually invoked by using:

(A) A software interrupt

(B) Polling

(C) An indirect jump

(D) A privileged instruction

Q38: Which of the following is an example of a spooled device?

(A) The terminal used to enter the input data for the C program being executed.

(B) An output device used to print the output of a number of jobs.

(C) The secondary memory device in a virtual storage system.

(D) The swapping area on a disk used by the sapper.

Q39: Consider n processes sharing the CPU in a round-robin fashion. 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 ?

>> $q*(n-1) + n*s <= t$

$q*(n-1) <= t - n*s$

$q <= (t-n.s) / (n-1)$

Q40: Which of the following is an example of spooled device?

I> A line printer used to print the output of a number of jobs

II>A terminal used to enter input data to a running program

III>A secondary storage device in a virtual memory system

IV>A graphic display device

Q41: Which scheduling policy is most suitable for a time shared operating system?

Shortest Job First

Round Robin

First Come First Serve

Elevator

Q42: Which of the following standard C library functions will always invoke a system call when executed from a single-threaded process in a UNIX/Linux operating system?

(A) exit

(B) malloc

(C) sleep

(D) strlen

Q43: 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

(C) 12

(D) 15

>>Time-Interval	Tasks
0-1	T1
1-2	T2
2-3	T2
3-4	T1 [Second Instance of T1 arrives]
4-5	T3
5-6	T3
6-7	T1 [Third Instance of T1 arrives] [Therefore T3 is preempted]
7-8	T2 [Second instance of T2 arrives]
8-9	T2
9-10	T1 [Fourth Instance of T1 arrives]
10-11	T3
11-12	T3 [First Instance of T3 completed]

Q44: Three processes A, B and C each execute a loop of 100 iterations. In each iteration of the loop, a process performs a single computation that requires tc CPU milliseconds and then initiates a single I/O operation that lasts for tio milliseconds. It is assumed that the computer where the processes execute has sufficient number of I/O devices and the OS of the computer assigns different I/O devices to each process. Also, the scheduling overhead of the OS is negligible. The processes have the following characteristics:

Process id	tc	tio
A	100 ms	500 ms
B	350 ms	500 ms
C	200 ms	500 ms

The processes A, B, and C are started at times 0, 5 and 10 milliseconds respectively, in a pure time sharing system (round robin scheduling) that uses a time slice of 50 milliseconds. The time in milliseconds at which process C would complete its first I/O operation is _____.

>>There are three processes A, B and C that run in round robin manner with time slice of 50 ms.

Processes start at 0, 5 and 10 milliseconds.

The processes are executed in below order

A, B, C, A

50 + 50 + 50 + 50 (200 ms passed)

Now A has completed 100 ms of computations and goes for I/O now

B, C, B, C, B, C

50 + 50 + 50 + 50 + 50 + 50 (300 ms passed)

C goes for i/o at 500ms and it needs 500ms to finish the IO.

So C would complete its first IO at 1000 ms

Q45: 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 quantum of 2 time units) are

(A)FCFS: P1, P2, P3 RR2: P1, P2, P3

(B)FCFS: P1, P3, P2 RR2: P1, P3, P2

(C)FCFS: P1, P2, P3 RR2: P1, P3, P2

(D)FCFS: P1, P3, P2 RR2: P1, P2, P3

>>FCFS is clear.

In RR, time slot is of 2 units.

Processes are assigned in following order

p1, p2, p1, p3, p2, p1, p3, p2, p2

This question involves the concept of ready queue. At t=2, p2 starts and p1 is sent to the ready queue and at t=3 p3 arrives so then the job p3 is queued in ready queue after p1. So at t=4, again p1 is executed then p3 is executed for first time at t=6.

12 May

Q46: 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 arrival or completion of processes. What is the average waiting time for the three processes?

(A) 5.0 ms

(B) 4.33 ms

(C) 6.33

(D) 7.33

Q47: A process executes the following code

for (i = 0; i < n; i++) fork();

The total number of child processes created is

(A) n

(B) 2n - 1

(C) 2n

(D) 2(n+1) - 1

Q48: Which of the following is/are true of the auto-increment addressing mode?

I. It is useful in creating self-relocating code.

II. If it is included in an Instruction Set Architecture, then an additional ALU is required for effective address calculation.

III. The amount of increment depends on the size of the data item accessed.

(A) I only

(B) II only

(C) III Only

(D) II and III only

Q49: 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

Q50: 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

.

>>Let the processes be p0, p1 and p2. These processes will be executed in following order.

```
  p2  p1  p2  p1  p2  p0  p1   p2   p0   p1   p2
0   4   5   6   7   8   9   10    11   12   13   14
```

Turn around time of a process is total time between submission of the process and its completion

Q51: 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 gets 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%

Q52: 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

>> The following is Gantt Chart of execution

P1	P2	P4	P3	P1
1	4	5	8	12

Turn Around Time = Completion Time – Arrival Time

Avg Turnaround Time = (12 + 3 + 6+ 1)/4 = 5.50

Q53: A uni-processor computer system only has two processes, both of which alternate 10ms CPU bursts with 90ms 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 utilization (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

Q54: Which combination of the following features will suffice to characterize an OS as a multi-programmed OS?

(a) 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 program terminates, another program is immediately scheduled for execution.

(A) a

(B) a and b

(C) a and c

(D) a, b and c

Q55: Which of the following does not interrupt a running process?

(A) A device

(B) Timer

(C) Scheduler process

(D) Power failure

Q56: Which of the following actions is/are typically not performed by the operating system when switching context from process A to process B?

(A) Saving current register values and restoring saved register values for process B.

(B) Changing address translation tables.

(C) Swapping out the memory image of process A to the disk.

(D) Invalidating the translation look-aside buffer.

Q57: Four jobs to be executed on a single processor system arrive at time 0 in the order A, B, C, D. Their burst CPU time requirements are 4, 1, 8, 1 time units respectively. The completion time of A under round robin scheduling with time slice of one time unit is

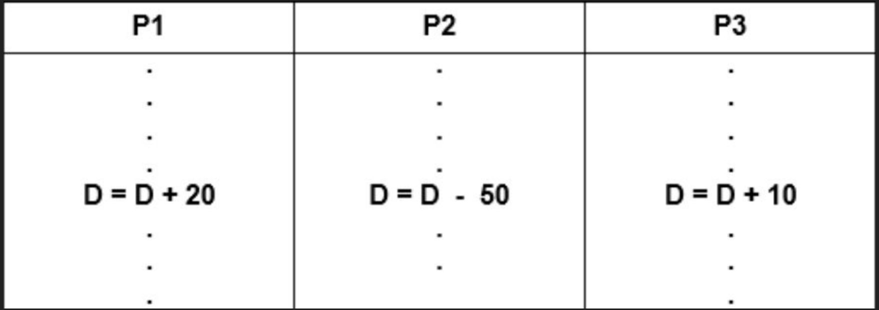
(A) 10

(B) 4

(C) 8

(D) 9

Q58: Consider three concurrent processes P1, P2 and P3 as shown below, which access a shared variable D that has been initialized to 100.



P1	P2	P3
.	.	.
.	.	.
.	.	.
.	.	.
D = D + 20	D = D - 50	D = D + 10
.	.	.
.	.	.
.	.	.

The process are executed on a uniprocessor system running a time-shared operating system. If the minimum and maximum possible values of D after the three processes have completed execution are X and Y respectively, then the value of Y–X is _____.

Note: This is a Numerical Type question.

(A) 80

(B) 130

(C) 50

(D) None of these

>> Minimum value (X) of D will possible when,

P2 reads D=100, preempted.

P1 executes D=D+20, D=120.

P3 executes D=D+10, D=130.

Now, P2 has D=100, executes, D = D-50 = 100-50 = 50. P2 writes D=50 final value.

Q59: A file is organized so that the ordering of data records is the same as or close to the ordering of data entries in some index. Then that index is called

(A) Dense

(B) Sparse

(C) Clustered

(D) Unclustered

Q60: The following two functions P1 and P2 that share a variable B with an initial value of 2 execute concurrently.

```
P1()
{  C = B – 1;
   B = 2*C;  }
P2()
{D = 2 * B;
   B = D - 1; }
```

The number of distinct values that B can possibly take after the execution is

(A) 3

(B) 2

(C) 5

(D) 4

13 May

Q61: 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)

(B) 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)

Q62: 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;

Method Used by P2

while (S1 != S2) ;

Critical Section

S2 = not (S1);

Which one of the following statements describes the properties achieved?

(A) Mutual exclusion but not progress

(B) Progress but not mutual exclusion

(C) Neither mutual exclusion nor progress

(D) Both mutual exclusion and progress

Q63: Where does the swap space reside?

(A) RAM

(B) Disk

(C) ROM

(D) On-chip cache

>>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 process are put in swap space of disk.

Q64: Let m[0]…m[4] be mutexes (binary semaphores) and P[0] …. P[4] be processes.

Suppose each process P[i] executes the following:

```
  wait (m[i]); wait(m[(i+1) mode 4]);
  release (m[i]); release (m[(i+1)mod 4]);
```

This could cause:

(A) Thrashing

(B) Deadlock

(C) Starvation, but not deadlock

(D) None of the above

Q65: When the result of a computation depends on the speed of the processes involved, there is said to be

(A) cycle stealing

(B) race condition

(C) a time lock

(D) a deadlock

Q66: A counting semaphore was initialized to 10. Then 6 P (wait) operations and 4 V (signal) operations were completed on this semaphore. The resulting value of the semaphore__________.

>> 8

Q67: 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 reads 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

(D)2

Q68: Each of a set of n processes executes the following code using two semaphores a and b initialized to 1 and 0, respectively. Assume that count is a shared variable initialized to 0 and not used in CODE SECTION P.

CODE SECTION P

wait(a); count=count+1;

if (count==n) signal (b);

signal (a): wait (b) ; signal (b);

CODE SECTION Q

What does the code achieve ?

(A) It ensures that no process executes CODE SECTION Q before every process has finished CODE SECTION P

(B) It ensures that two processes are in CODE SECTION Q at any time

(C) It ensures that all processes execute CODE SECTION P mutually exclusively

(D) It ensures that at most n−1 processes are in CODE SECTION P at any time

Q69: A certain computation generates two arrays a and b such that a[i]=f(i) for 0 ≤ i < n and b[i]=g(a[i]) for 0 ≤ i < n. Suppose this computation is decomposed into two concurrent processes X and Y such that X computes the array a and Y computes the array b. The processes employ two binary semaphores R and S, both initialized to zero. The array a is shared by the two processes. The structures of the processes are shown below.

Process X:

```
private i;
for (i=0; i < n; i++) {
   a[i] = f(i);
   ExitX(R, S);
}
```

Process Y:

```
private i;
for (i=0; i < n; i++) {
   EntryY(R, S);
      b[i]=g(a[i]);
}
```

Which one of the following represents the CORRECT implementations of ExitX and EntryY?

```
(A)ExitX(R, S) {
  P(R);
  V(S);}
EntryY (R, S) {
  P(S);
  V(R);}
```

```
(B)ExitX(R, S) {
  V(R);
  V(S);}
EntryY(R, S) {
  P(R);
  P(S);}
```

(C)ExitX(R, S) {

```
  P(S);
  V(R);}
```

EntryY(R, S) {

```
  V(S);
  P(R);}
```

```
D)ExitX(R, S) {
  V(R);
  P(S);}
EntryY(R, S) {
  V(S);
  P(R);}
```

Q70: Fetch_And_Add(X,i) is an atomic Read-Modify-Write instruction that reads the value of memory location X, increments it by the value i, and returns the old value of X. It is used in the pseudocode shown below to implement a busy-wait lock. L is an unsigned integer shared variable initialized to 0. The value of 0 corresponds to lock being available, while any non-zero value corresponds to the lock being not available.

```
AcquireLock(L){
    while (Fetch_And_Add(L,1))
        L = 1; }
ReleaseLock(L){
    L = 0;  }
```

This implementation

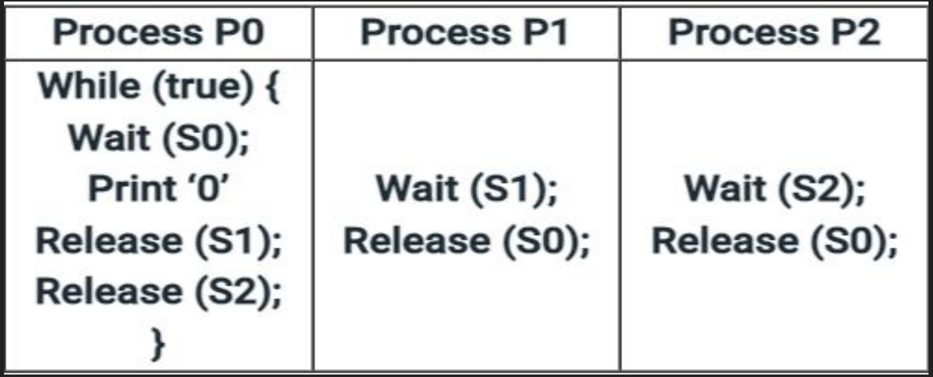
(A) fails as L can overflow

(B) fails as L can take on a non-zero value when the lock is actually available

(C) works correctly but may starve some processes

(D) works correctly without starvation

Q71: 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’	Wait (S1);	Wait (S2);
Release (S1);	Release (S0);	Release (S0);
Release (S2);		
}		

How many times will process P0 print ‘0’?

(A) At least twice

(B) Exactly twice

(C) Exactly thrice

(D) Exactly once

Q72: 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 is TRUE?

(A) I only

(B) I and II

(C) II and III

(D) IV only

Q73: The P and V operations on counting semaphores, where s is a counting semaphore, 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 Pb and Vb the wait and signal operations on binary semaphores are provided. Two binary semaphores Xb and Yb are used to implement the semaphore operations P(s) and V(s) as follows:

```
P(s) : Pb(Xb);
  s = s - 1;
  if (s < 0) { Vb(Xb) ;
   Pb(Yb) ;}
  else Vb(Xb);
V(s) : Pb(Xb) ;
  s = s + 1;
  if (s <= 0) Vb(Yb) ;
  Vb(Xb) ;
```

The initial values of Xb and Yb are respectively

(A) 0 and 0

(B) 0 and 1

(C) 1 and 0

(D) 1 and 1

Q74: Two processes, P1 and P2, need to access a critical section of code. Consider the following synchronization construct used by the processes:Here, wants1 and wants2 are shared variables, which are initialized to false. Which one of the following statements is TRUE about the above construct?

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

(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

Two processes, P1 and P2, need to access a critical section of code. Here, wants1 and wants2 are shared variables, which are initialized to false.

Now, when both wants1 and wants2 become true, both process p1 and p2 enter in while loop and waiting for each other to finish. This while loop run indefinitely which leads to deadlock.

Now, Assume P1 is in critical section (it means wants1=true, wants2 can be anything, true or false). So this ensures that p2 won't enter in critical section and vice versa. This satisfies the property of mutual exclusion.

Here bounded waiting condition is also satisfied as there is a bound on the number of process which gets access to critical section after a process request access to it.

Q75: A process in the context of computing is?

(A)A set of instructions to be executed on a computer

(B)A program in execution

(C)A piece of hardware that executes a set of instructions

(D)The main procedure of a program

Q76: Which technique was introduced because a single job could not keep both CPU and IO devices busy?

(A) Real time

(B)Spooling

(C)Preemptive scheduling

(D)Multiprogramming

Q77: The atomic fetch-and-set x, y instruction unconditionally sets the memory location x to 1 and fetches the old value of x n 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_semaphore *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

Q78: 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 left.

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

The variables 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.

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

Q79: 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:

```
while (1) {
W:
   print '0';
   print '0';
X:
}
```

Process Q:

```
while (1) {
Y:
   print '1';
   print '1';
Z:
}
```

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

Which of the following will ensure that the output string never contains a substring of the form 01^n0 or 10^n1 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(S) at Y, V(S) at Z, S initially 1

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

Q80: 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 = i

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

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

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

Q81: Each Process Pi, i= 1…….9 is coded as follows

```
 repeat
    P(mutex)
    {Critical section}
    V(mutex)
 forever
```

The code for P10 is identical except it uses V(mutex) in place of P(mutex). What is the largest number of processes that can be inside the critical section at any moment?

(A) 1

(B) 2

(C) 3

(D) None of above

Q82: A solution to the Dining Philosophers Problem which avoids deadlock is:

(A) ensure that all philosophers pick up the left fork before the right fork

(B) ensure that all philosophers pick up the right fork before the left fork

(C) ensure that one particular philosopher picks up the left fork before the right fork, and that all other philosophers pick up the right fork before the left fork

(D) None of the above

Q83: At a particular time of computation the value of a counting semaphore is 7. Then 20 P operations and 15 V operations were completed on this semaphore. The resulting value of the semaphore is :

(A) 42

(B) 2

(C) 7

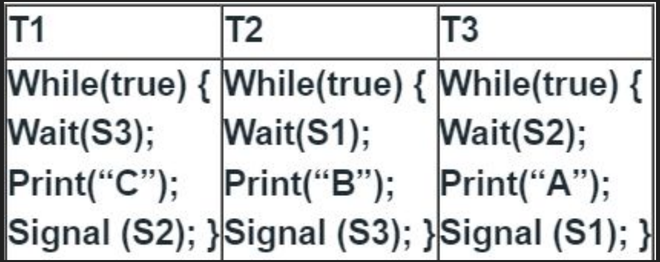
(D) 12

>>Value of a counting semaphore = 7

After 20 P operations value of semaphore = 7 – 20 = -13

After 15 V operations value of semaphore = -13 + 15 = 2

Q84: Consider the following threads, T1, T2, and T3 executing on a single processor, synchronized using three binary semaphore variables, S1, S2, and S3, operated upon using standard wait() and signal(). The threads can be context switched in any order and at any time.



T1	T2	T3
While(true) {	While(true) {	While(true) {
Wait(S3);	Wait(S1);	Wait(S2);
Print(“C”);	Print(“B”);	Print(“A”);
Signal (S2); }	Signal (S3); }	Signal (S1); }

Which initialization of the semaphores would print the sequence BCABCABCA….?

(A)S1 = 1; S2 = 1; S3 = 1

(B)S1 = 1; S2 = 1; S3 = 0

(C)S1 = 1; S2 = 0; S3 = 0

(D)S1 = 0; S2 = 1; S3 = 1

Q85: Which of the following statements is/are true with respect to deadlocks?

A)Circular wait is a necessary condition for the formation of deadlock.

B)In a system where each resource has more than one instance, a cycle in its wait-for graph indicates the presence of a deadlock.

IC)f the current allocation of resources to processes leads the system to unsafe state, then deadlock will necessarily occur.

D) In the resource-allocation graph of a system, if every edge is an assignment edge, then the system is not in deadlock state.

Q86: Consider a system with 3 processes that share 4 instances of the same resource type. Each process can request a maximum of K instances. Resource instances can be requested and released only one at a time. The largest value of K that will always avoid deadlock is ____ .

(A) 1

(B) 2

(C) 3

(D) 4

>>Since deadlock-free condition is:

$R \geq P(N - 1) + 1$

Where R is total number of resources,
P is the number of processes, and
N is the max need for each resource.

$4 \geq 3(N - 1) + 1$
$3 \geq 3(N - 1)$
$1 \geq (N - 1)$
$N \leq 2$

Therefore, the largest value of K that will always avoid deadlock is 2

Q87: A system has 6 identical resources and N processes competing for them. Each process can request atmost 2 resources. Which one of the following values of N could lead to a deadlock?

(A) 1

(B) 2

(C) 3

(D) 4

Q88: A computer has six tape drives, with n processes competing for them. Each process may need two drives. What is the maximum value of n for the system to be deadlock free?

(A) 6

(B) 5

(C) 4

(D) 3

>>Given tape drive = 6 and each process may need 2 drive.

When we give 1 drive to 1 process then total process will be 6 but in this case there will definitely deadlock occur because every process contain 1 drive and waiting for another drive which is hold by other process therefore when we reduce 1 process then system to be deadlock free.

Hence maximum value of n = 6 – 1 = 5.

Q89: Consider the following pseudocode, where S is a semaphore initialized to 5 in line #2 and counter is a shared variable initialized to 0 in line #1. Assume that the increment operation in line #7 is not atomic.

```
1.  int counter =0;
2.  Semaphore S= init(5);
3.  void parop(void)
4.  {
5.        wait(S);
6.        wait(S);
7.        counter++;
8.        signal(S);
9.        signal(S);
10. }
```

If five threads execute the function parop concurrently, which of the following program behavior(s) is/are possible?

(A) The value of counter is 5 after all the threads successfully complete the execution of parop

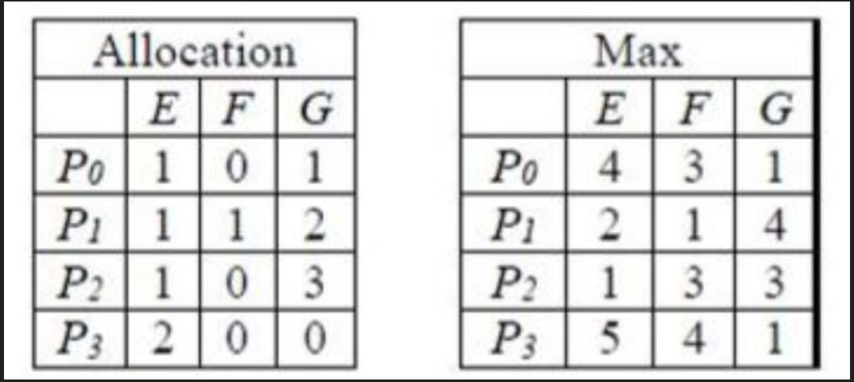
(B) The value of counter is 1 after all the threads successfully complete the execution of parop

(C) The value of counter is 0 after all the threads successfully complete the execution of parop

(D) There is a deadlock involving all the threads

Q90: 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

Q91: From the perspective of deadlock avoidance, which one of the following is true?

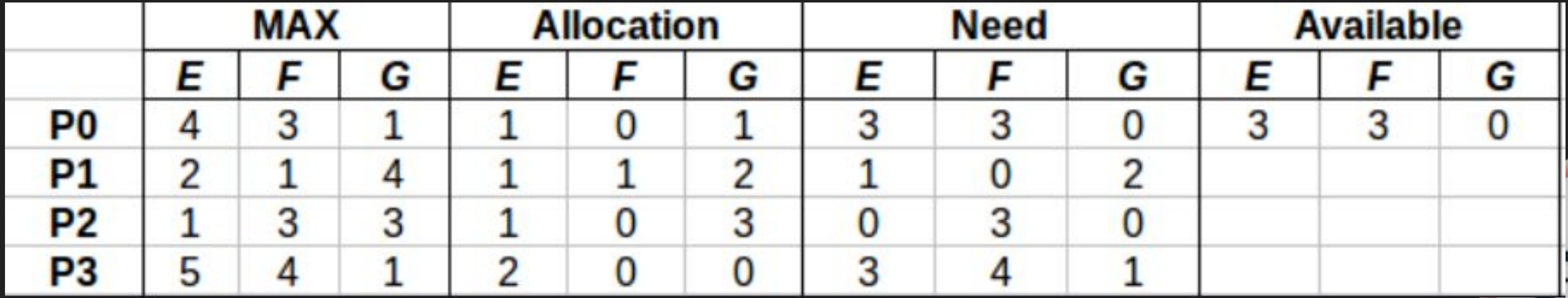
(A) The system is in safe state

(B) The system is not in safe state, but would be safe if one more instance of E were available

(C) The system is not in safe state, but would be safe if one more instance of F were available

(D) The system is not in safe state, but would be safe if one more instance of G were available

>>



	MAX			Allocation			Need			Available		
	E	F	G	E	F	G	E	F	G	E	F	G
P0	4	3	1	1	0	1	3	3	0	3	3	0
P1	2	1	4	1	1	2	1	0	2			
P2	1	3	3	1	0	3	0	3	0			
P3	5	4	1	2	0	0	3	4	1			

Q92: Consider a non-negative counting semaphore S. The operation P(S) decrements S, and V(S) increments S. During an execution, 20 P(S) operations and 12 V(S) operations are issued in some order. The largest initial value of S for which at least one P(S) operation will remain blocked is _____.

(A) 7

(B) 8

(C) 9

(D) 10

Q93: Consider the following policies for preventing deadlock in a system with mutually exclusive resources.

I. Processes should acquire all their resources at the beginning of execution. If any resource is not available, all resources acquired so far are released.

II. The resources are numbered uniquely, and processes are allowed to request for resources only in increasing resource numbers.

III. The resources are numbered uniquely, and processes are allowed to request for resources only in decreasing resource numbers.

IV. The resources are numbered uniquely. A process is allowed to request only for a resource with resource number larger than its currently held resources.

Which of the above policies can be used for preventing deadlock?

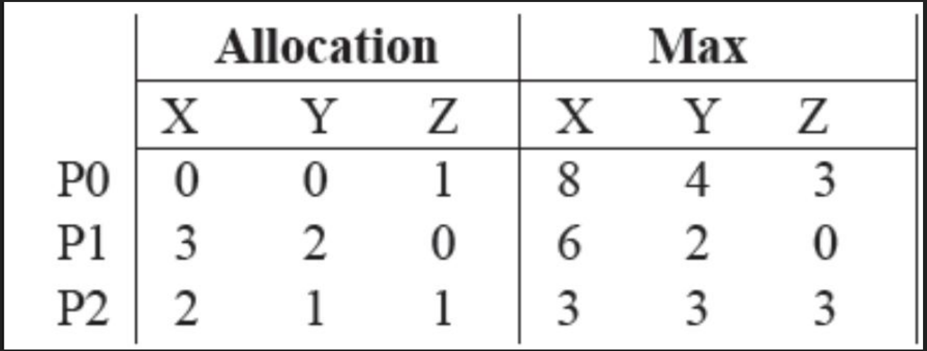
(A) Any one of I and III but not II or IV

(B) Any one of I, III and IV but not II

(C) Any one of II and III but not I or IV

(D) Any one of I, II, III and IV

Q94: Can 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:

REQ1: P0 requests 0 units of X,

0 units of Y and 2 units of Z

REQ2: P1 requests 2 units of X,

0 units of Y and 0 units of Z

Which one of the following is TRUE?

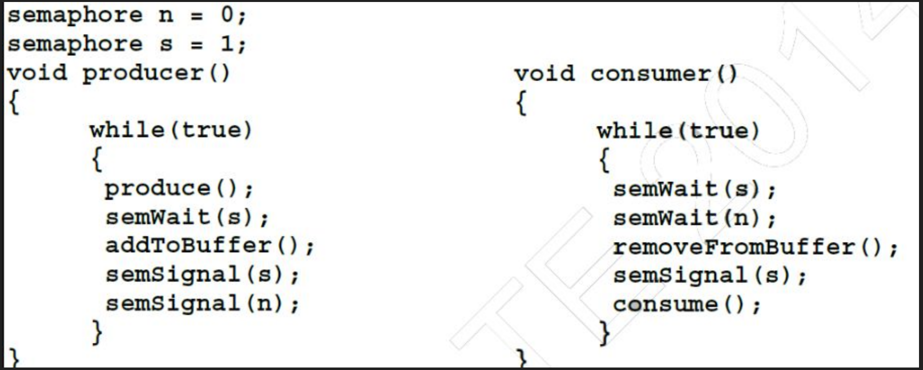
(A) Only REQ1 can be permitted.

(B) Only REQ2 can be permitted.

(C) Both REQ1 and REQ2 can be permitted.

(D) Neither REQ1 nor REQ2 can be permitted

Q95: Consider the procedure below for the Producer-Consumer problem which uses semaphores:



```
semaphore n = 0;
semaphore s = 1;
void producer()                         void consumer()
{                                       {
      while(true)                             while(true)
      {                                       {
       produce();                              semWait(s);
       semWait(s);                             semWait(n);
       addToBuffer();                          removeFromBuffer();
       semSignal(s);                           semSignal(s);
       semSignal(n);                           consume();
      }                                       }
}                                       }
```

Which one of the following is TRUE?

(A) The producer will be able to add an item to the buffer, but the consumer can never consume it.

(B) The consumer will remove no more than one item from the buffer.

(C) Deadlock occurs if the consumer succeeds in acquiring semaphores when the buffer is empty.

(D) The starting value for the semaphore n must be 1 and not 0 for deadlock-free operation.

Initially, there is no element in the buffer.

Semaphore s=1 and semaphore n=0.

We assume that initially control goes to the consumer when buffer is empty.

semWait(s) decrements the value of semaphore ‘s’ . Now, s = 0 and semWait(n) decrements the value of semaphore ‘n’.

Since, the value of semaphore ‘n’ becomes less than 0 , the control stucks in while loop of function semWait() and a deadlock arises.

Thus, deadlock occurs if the consumer succeeds in acquiring semaphore s when the buffer is empty.

Q96: A system has n resources R0,.....,Rn-1, and k processes P0,.....,Pk-1. The implementation of the resource request logic of each process Pi, is as follows:

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

In which situation is a deadlock possible?

>>n =21 k =12

Q97: Consider a system with 4 types of resources R1 (3 units), R2 (2 units), R3 (3 units), R4 (2 units). A non-preemptive resource allocation policy is used. At any given instance, a request is not entertained if it cannot be completely satisfied. Three processes P1, P2, P3 request the sources as follows if executed independently.

Process P1:

t=0: requests 2 units of R2

t=1: requests 1 unit of R3

t=3: requests 2 units of R1

t=5: releases 1 unit of R2 and 1 unit of R1.

t=7: releases 1 unit of R3

t=8: requests 2 units of R4

t=10: Finishes

Process P2:

t=0: requests 2 units of R3

t=2: requests 1 unit of R4

t=4: requests 1 unit of R1

t=6: releases 1 unit of R3

t=8: Finishes

Process P3:

t=0: requests 1 unit of R4

t=2: requests 2 units of R1

t=5: releases 2 units of R1

t=7: requests 1 unit of R2

t=8: requests 1 unit 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 a deadlock.

(D) All three processes will be in deadlock

Q98: 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 result state is safe

(C) Deadlock avoidance is less restrictive than deadlock prevention

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

Deadlock Prevention: Deadlocks can be prevented by preventing at least one of the four required conditions:

1. Mutual Exclusion – not required for sharable resources; must hold for non-sharable resources.

2. Hold and Wait – must guarantee that whenever a process requests a resource, it does not hold any other resources. Require process to request and be allocated all its sources before it begins execution, or allow process to request resources only when the process has none. Low resource utilization; starvation possible. Restrain the ways request can be made.

3. No Pre-emption – If a process that is holding some resources requests another resource that cannot be immediately allocated to it, and then all resources currently being held are released. Pre-empted resources are added to the list of resources for which the process is waiting. Process will be restarted only when it can regain its old resources, as well as the new ones that it is requesting.

4. Circular Wait – impose a total ordering of all resource types, and require that each process requests resources in an increasing order of enumeration.

Q99: 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	1 2 1	1 0 3
P1	2 0 1	0 1 2
P2	2 2 1	1 2 0

(A) P0

(B) P!

(C) P2

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

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.

So, the resource instances which are left being unallocated = { unallocated resources= total resources-allocated resources }

Q100: Consider the following snapshot of a system running n processes. Process i is holding Xi instances of a resource R, 1 <= i <= n. currently, all instances of R are occupied. Further, for all i, process i has placed a request for an additional Yi instances while holding the Xi instances it already has. There are exactly two processes p and q such that Yp = Yq = 0. Which one of the following can serve as a necessary condition to guarantee that the system is not approaching a deadlock?

(A) min (Xp, Xq) < max (Yk) where k != p and k != q

(B) Xp + Xq >= min (Yk) where k != p and k != q

(C) max (Xp, Xq) > 1

(D) min (Xp, Xq) > 1

>>Necessary condition to guarantee no deadlock which means without satisfying this condition, no deadlock is possible. Both the process p and q have no additional requirement; they both can be finished releasing Xp + Xq resources without asking for any additional resource. Using this, we can finish one more process only if condition B is satisfied. If the resources released by p and q are sufficient for another process waiting for Yk resources, then system is not approaching deadlock.
i.e Xp+Xq > min (Yk) where k != p and k != q

Q101: 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 Si, where Si > 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) C]

(D) D

Q102: Which of the following is NOT a valid deadlock prevention scheme?

(A) Release all resources before requesting a new resource

(B) Number the resources uniquely and never request a lower numbered resource than the last one requested.

(C) Never request a resource after releasing any resource

(D) Request and all required resources be allocated before execution.

Q103: Consider a system having ‘m’ resources of the same type. These resources are shared by three processes P1, P2and P3 which have peak demands of 2, 5 and 7 resources respectively. For what value of ‘m’ deadlock will not occur ?

(A) 70

(B) 14

(C) 13

(D) 7

To avoid deadlock ‘m’ >= peak demands(P1 + P2 + P3)
i.e. m >= peak demands(2 + 5 + 7)
m >= peak demands(14)

Q104: A computer has six tape drives, with n processes competing for them. Each process may need two drives. What is the maximum value of n for the system to be deadlock free?

(A) 6

(B) 5

(C) 4

(D) 3

Q105: Which one of the following statements is FALSE?

(A)The TLB performs an associative search in parallel on all its valid entries using the page number of the incoming virtual addresses.

(B)If the virtual address of a word given by the CPU has a TLB hit, but the subsequent search for the word results in a cache miss, then the word will always be present in the main memory

(C)The memory access time using a given inverted page table is always the same for all incoming virtual addresses.

(D)In a system that uses hashed page tables, if two distinct virtual addresses V1 and V2 map to the same value while hashing, then the memory access time of these addresses will not be the same.

Q106: Consider a demand paging system with four-page frames (initially empty) and an LRU page replacement policy. For the following page reference string 7, 2,7,3, 2,5,3, 4,6,7,7,1,5,6,1 the page fault rate, defined as the ratio of number of page faults to the number of memory accesses (rounded off to one decimal place) is______.

(A)1.5

(B)0.5

(C)0.6

(D)0.8

>> 7,2,7,3,2,5,3,4,6,7,7,1,5,6,1

7

2

3

5

Total page fault=9

Total access=15

thus page fault rate =9/15=0.6

Q107: Consider allocation of memory to a new process. Assume that none of the existing holes in the memory will exactly fit the process's memory requirement. Hence, a new hole of smaller size will be created if allocation is made in any of the existing holes. Which one of the following statement is TRUE ?

(A) The hole created by first fit is always larger than the hole created by next fit.

(B) The hole created by worst fit is always larger than the hole created by first fit.

(C) The hole created by best fit is never larger than the hole created by first fit.

(D) The hole created by next fit is never larger than the hole created by best fit.

>> (A) May not be correct if size of partition in first fit smaller than size of partition in next fit.

(B) May not be correct if both worst fit and first fit in same partition.

(C) Always correct, because best fit always has minimum hole.

(D) Absolutely false.

Q108: In which one of the following page replacement algorithms it is possible for the page fault rate to increase even when the number of allocated frames increases?

(A) LRU (Least Recently Used)

(B) OPT (Optimal Page Replacement)

(C) MRU (Most Recently Used)

(D) FIFO (First In First Out)

>>In some situations FIFO page replacement gives more page faults when increasing the number of page frames. This situation is Belady’s anomaly.
Belady’s anomaly proves that it is possible to have more page faults when increasing the number of page frames while using the First in First Out (FIFO) page replacement algorithm. For example, if we consider reference string 3 2 1 0 3 2 4 3 2 1 0 4 and 3 slots, we get 9 total page faults, but if we increase slots to 4, we get 10 page faults.

Q109: Consider a system with byte-addressable memory, 32 bit logical addresses, 4 kilobyte page size and page table entries of 4 bytes each. The size of the page table in the system in megabytes is ______.

(A) 2

(B) 4

(C) 8

(D) 16

>>Number of entries in page table = 232 / 4Kbyte

= 232 / 212

= 220

Size of page table = (No. page table entries)*(Size of an entry)

= 220 * 4 bytes

= 222

= 4 Megabytes

Q110: A computer system implements a 40 bit virtual address, page size of 8 kilobytes, and a 128-entry translation look-aside buffer (TLB) organized into 32 sets each having four ways. Assume that the TLB tag does not store any process id. The minimum length of the TLB tag in bits is ____

(A) 20

(B) 10

(C) 11

(D) 22

>>Total virtual address size = 40

Since there are 32 sets, set offset = 5

Since page size is 8kilobytes, word offset = 13

Minimum tag size = 40 - 5- 13 = 22

Q111:A system uses 3 page frames for storing process pages in main memory. It uses the Least Recently Used (LRU) page replacement policy. Assume that all the page frames are initially empty. What is the total number of page faults that will occur while processing the page reference string given below?

4, 7, 6, 1, 7, 6, 1, 2, 7, 2

(A) 4

(B) 5

(C) 6

(D) 7

Q112: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

Q113:In which one of the following page replacement policies, Belady’s anomaly may occur?

(A) FIFO

(B) Optimal

(C) LRU

(D) MRU

>> Belady’s anomaly proves that it is possible to have more page faults when increasing the number of page frames while using the First in First Out (FIFO) page replacement algorithm.

Q114: Page information in memory is also called as Page Table. The essential contents in each entry of a page table is/are ____.

(A) Page Access information

(B) Virtual Page number

(C) Page Frame number

(D) Both virtual page number and Page Frame Number

>> Belady’s anomaly proves that it is possible to have more page faults when increasing the number of page frames while using the First in First Out (FIFO) page replacement algorithm

Q115: How many 32K x 1 RAM chips are needed to provide a memory capacity of 256K-bytes?

(A) 8

(B) 32

(C) 64

(D) 128

>> We need 256 Kbytes, i.e., 256 x 1024 x 8 bits.
We have RAM chips of capacity 32 Kbits = 32 x 1024 bits.
(256 * 1024 * 8)/(32 * 1024) = 64

Q116: What is the swap space in the disk used for?

(A) Saving temporary html pages

(B) Saving process data

(C) Storing the super-block

(D) Storing device drivers

Q117: Consider a program P that consists of two source modules M1 and M2 contained in two different files. If M1 contains a reference to a function defined in M2, the reference will be resolved at

(A) Edit-time

(B) Compile-time

(C) Link-time

(D) Load-time

Static linking is done at link-time, dynamic linking or shared libraries are brought in only at runtime.

(A) Edit-time : Function references can never be given/determined at edit-time or code-writing time. Function references are different from function names. Function names are used at edit-time and function references are determined at linker-time for static libraries or at runtime for dynamic libraries.

(B) Compile-time : Compile time binding is done for functions present in the same file or module.

(C) Link-time : Link time binding is done in the linker stage, where functions present in separate files or modules are referenced in the executable.

(D) Load-time : Function referencing is not done at load-time.

Q118: 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

Q119: 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

Q120: The optimal page replacement algorithm will select the page that

(A)Has been used least number of times.

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

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

(D)Has been used most number of times.

>> The optimal page replacement algorithm will select the page whose next occurrence will be after the longest time in future. For example, if we need to swap a page and there are two options from which we can swap, say one would be used after 10s and the other after 5s, then the algorithm will swap out the page that would be required 10s later.

Q121: The process of assigning load addresses to the various parts of the program and adjusting the code and date in the program to reflect the assigned addresses is called

(A) Assembly

(B) Parsing

(C) Relocation

(D) Symbol resolution

Q122: 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

Q123: In a resident- OS computer, which of the following system software must reside in the main memory under all situations?

(A) Assembler

(B) Linker

(C) Loader

(D) Compiler

Q124: Locality of reference implies that the page reference being made by a process

(A) will always be to the page used in the previous page reference

(B) is likely to be to one of the pages used in the last few page references

(C) will always be to one of the pages existing in memory

(D) will always lead to a page fault

Q125: The principal of the locality of reference justifies the use of

(A) virtual memory

(B) interrupts

(C) main memory

(D) cache memory

Q126: A linker is given object modules for a set of programs that were compiled separately. What information need to be included in an object module?

A) Object code

B) Relocation bits

C) Names and locations of all external symbols defined in the object module

D) Absolute addresses of internal symbols

Q127: Which page replacement policy sometimes leads to more page faults when size of memory is increased?

A] Optimal

B] LRU

C] FIFO

D] None of these.

Q128: In the context of operating systems, which of the following statements is/are correct with respect to paging?

(A) Paging helps solve the issue of external fragmentation

(B) Page size has no impact on internal fragmentation

(C) Paging incurs memory overheads

(D) Multi-level paging is necessary to support pages of different sizes

ept

Q129: Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and TLB lookup takes 20 ns. Each page transfer to/from the disk takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. Assume that for 20% of the total page faults, a dirty page has to be written back to disk before the required page is read from disk. TLB update time is negligible.

The average memory access time in ns (round off to 1 decimal places) is ______ .

(A) 154.5

(B) 155

(C) 755

(D) 725

M = 100 ns

T = 20 ns

D = 5000 ns

h = 0.95

p = 0.1,

(1-p) = 0.9

d = 0.2,

(1-d) = 0.8

Therefore, average memory access time,

= h×(T+M) + (1-h)[(1-p)×2M + p[(1-d)[D+M] + d(2D+M)] + T]

= 0.95×(20+100) + (1-0.95)[(1-0.1)×200 + (0.1)[(1-0.2)[5000+100] + (0.2)(10000+100)] + 20]

= 154.5 (in ns)

Q130: Assume that in a certain computer, the virtual addresses are 64 bits long and the physical addresses are 48 bits long. The memory is word addressable. The page size is 8k Band the word size is 4 bytes. The Translation Lookaside Buffer (TLB) in the address translation path has 128 valid entries. At most how many distinct virtual addresses can be translated without any TLB miss?

(A) 16 x 210

(B) 8 x 220

(C) 4 x 220

(D) 256 x 210

Number of words in a page

= Page size / word size

= 8 KB / 4 B

= 2 K

= 2 * 210

Since TLB can hold 128 valid entries, therefore, TLB can translate

= 128 * number of words in page

= 128 * 2 * 210

= 256 * 210 addresses with TLB hit

Q131: Consider a process executing on an operating system that uses demand paging. The average time for a memory access in the system is M units if the corresponding memory page is available in memory, and D units if the memory access causes a page fault. It has been experimental measured that the average time taken for a memory access in the process is X units.

Which one of the following is the correct expression for the page fault rate experienced by the process?

(A) (D – M) / (X – M)

(B) (X – M) / (D – M)

(C) (D – X) / (D – M)

(D) (X – M) / (D – X)

Q132: Consider a computer system with 40-bit virtual addressing and page size of sixteen kilobytes. If the computer system has a one-level page table per process and each page table entry requires 48 bits, then the size of the per-process page table is ______megabytes.

(A) 384

(B) 48

(C) 192

(D) 96

Q133: Consider a computer system with ten physical page frames. The system is provided with an access sequence a1, a2, …, a20, a1, a2, …, a20), where each ai number. The difference in the number of page faults between the last-in-first-out page replacement policy and the optimal page replacement policy is _____

(A) 0

(B) 1

(C) 2

(D) 3

Q134: A Computer system implements 8 kilobyte pages and a 32-bit physical address space. Each page table entry contains a valid bit, a dirty bit three permission bits, and the translation. If the maximum size of the page table of a process is 24 megabytes, the length of the virtual address supported by the system is ______ bits

(A)36

(B)32

(C)28

(D)40

>> Max size of virtual address can be calculated by calculating maximum number of page table entries.

Maximum Number of page table entries can be calculated using given maximum page table size and size of a page table entry.

Given maximum page table size = 24 MB

Let us calculate size of a page table entry.

A page table entry has following number of bits.

1 (valid bit) +

1 (dirty bit) +

3 (permission bits) +

x bits to store physical address space of a page.

Value of x = (Total bits in physical address) -

(Total bits for addressing within a page)

Since size of a page is 8 kilobytes, total bits needed within

a page is 13.

So value of x = 32 - 13 = 19

Putting value of x, we get size of a page table entry =

1 + 1 + 3 + 19 = 24bits.

Number of page table entries

= (Page Table Size) / (An entry size)

= (24 megabytes / 24 bits)

= 223

Virtual address Size

= (Number of page table entries) * (Page Size)

= 223 * 8 kilobits

= 236

Therefore, length of virtual address space = 36

Q135: Consider six memory partitions of size 200 KB, 400 KB, 600 KB, 500 KB, 300 KB, and 250 KB, where KB refers to kilobyte. These partitions need to be allotted to four processes of sizes 357 KB, 210 KB, 468 KB and 491 KB in that order. If the best fit algorithm is used, which partitions are NOT allotted to any process?

(A) 200 KB and 300 KB

(B) 200 KB and 250 KB

(C) 250 KB and 300 KB

(D) 300 KB and 400 KB

Q136: Assume that there are 3 page frames which are initially empty. If the page reference string is 1, 2, 3, 4, 2, 1, 5, 3, 2, 4, 6, the number of page faults using the optimal replacement policy is_____.

(A) 5

(B) 6

(C) 7

(D) 8

Q137: Consider a paging hardware with a TLB. Assume that the entire page table and all the pages are in the physical memory. It takes 10 milliseconds to search the TLB and 80 milliseconds to access the physical memory. If the TLB hit ratio is 0.6, the effective memory access time (in milliseconds) is ____.

(A) 120

(B) 122

(C) 124

(D) 118

Q138: A computer has twenty physical page frames which contain pages numbered 101 through 120. Now a program accesses the pages numbered 1, 2, …, 100 in that order, and repeats the access sequence THRICE. Which one of the following page replacement policies experiences the same number of page faults as the optimal page replacement policy for this program?

(A) Least-recently-used

(B) First-in-first-out

(C) Last-in-first-out

(D) Most-recently-used

>> The optimal page replacement algorithm swaps out the page whose next use will occur farthest in the future

Q139: 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

(B) OPTIMAL < FIFO < LRU

(C) OPTIMAL = LRU

(D) OPTIMAL = FIFO

Q140: 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 a memory location.

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

(C) It is required by the translation lookaside buffer.

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

Q141: 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.

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

Q142: 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 programs do not exhibit locality of reference.

Which one of the following is TRUE?

(A)P is false, but Q is true

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

(C)Both P and Q are true, and Q is the reason for P

(D)Both P and Q are false

Q143: 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

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

>> Optimal replacement policy looks forward in time to see which frame to replace on a page fault.

1 23-> 1,2,3 //page faults

173 ->7

143 ->4

153 -> 5

163 -> 6

Total=7

Q144: 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 memory 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

Q145: 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 90%, 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

>> 100 + 2×(0.9×(0)+0.1×(2×150)) + 2×150 + 1 /10000 × 8 × 106

= 100 + 60 + 300 + 800

= 1260 ns

So, none option is correct. option D is close

Q146: 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

Q147: Dynamic linking can cause security concerns because:

(A) Security is dynamic

(B) The path for searching dynamic libraries is not known till runtime

(C) Linking is insecure

(D) Cryptographic procedures are not available for dynamic linking

Q149: Consider a machine with 64 MB physical memory and a 32-bit virtual address space. If the page size is 4KB, what is the approximate size of the page table?

(A)16 MB

(B)8 MB

(C)2 MB

(D)24 MB

Physical Address Space = 64MB = 226B
Virtual Address = 32-bits, ∴ Virtual Address Space = 232B
Page Size = 4KB = 212B
Number of pages = 232/212 = 220 pages.

Number of frames = 226/212 = 214 frames.

∴ Page Table Size = 220×14-bits ≈ 220×16-bits ≈ 220×2B = 2MB

Q150: Suppose the time to service a page fault is on the average 10 milliseconds, while a memory access takes 1 microsecond. Then a 99.99% hit ratio results in average memory access time of

(A) 1.9999 milliseconds

(B) 1 millisecond

(C) 9.999 microseconds

(D) 1.9999 microseconds

Q150: Which of the following is/are advantages of virtual memory?

a) Faster access to memory on an average.

b) Processes can be given protected address spaces.

c) Linker can assign addresses independent of where the program will be loaded in physical memory.

d) Programs larger than the physical memory size can be run.

(A) a and b

(B) b and c

(C) b and d

(D) All of the above

Q151: If an instruction takes i microseconds and a page fault takes an additional j microseconds, the effective instruction time if on the average a page fault occurs every k instructions is:

(A) i + j/k

(B) i + j* k

(C) (i + j)/ k

(D) (i + j)* k

Q151: A 1000 Kbyte memory is managed using variable partitions but no compaction. It currently has two partitions of sizes 200 Kbytes and 260 Kbytes respectively. The smallest allocation request in Kbytes that could be denied is for

(A) 151

(B) 181

(C) 231

(D) 541

Q152: Suppose a disk has 201 cylinders, numbered from 0 to 200. At some time the disk arm is at cylinder 100, and there is a queue of disk access requests for cylinders 30, 85, 90, 100, 105, 110, 135 and 145. If Shortest-Seek Time First (SSTF) is being used for scheduling the disk access, the request for cylinder 90 is serviced after servicing _____ number of requests.

(A) 1

(B) 2

(C) 3

(D) 4

>> In Shortest-Seek-First algorithm, request closest to the current position of the disk arm and head is handled first.

In this question, the arm is currently at cylinder number 100. Now the requests come in the queue order for cylinder numbers 30, 85, 90, 100, 105, 110, 135 and 145.

The disk will service that request first whose cylinder number is closest to its arm. Hence 1st serviced request is for cylinder no 100 (as the arm is itself pointing to it), then 105, then 110, and then the arm comes to service request for cylinder 90. Hence before servicing request for cylinder 90, the disk would had serviced 3 requests.

Q153: A FAT (file allocation table) based file system is being used and the total overhead of each entry in the FAT is 4 bytes in size. Given a 100 x 106 bytes disk on which the file system is stored and data block size is 103 bytes, the maximum size of a file that can be stored on this disk in units of 106 bytes is _____.

(A) 99.55 to 99.65

(B) 100.5 to 101.4

(C) 97.2 to 98.5

(D) 89.1 to 91.2

>> Here block size is 10^3 B.
No. of entries in the FAT = Disk capacity/ Block size
= 10^8/10^3
= 10^5
Total space consumed by FAT = 10^5 *4B
= 0.4*10^6B
Max. size of file that can be stored = 100*10^6-0.4*10^6
= 99.6*10^6B.

Q154: The data blocks 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

Q155: Consider a disk pack with 16 surfaces, 128 tracks per surface and 256 sectors per track. 512 bytes of data are stored in a bit serial manner in a sector. The capacity of the disk pack and the number of bits required to specify a particular sector in the disk are respectively:

(A) 256 Mbyte, 19 bits

(B) 256 Mbyte, 28 bits

(C) 512 Mbyte, 20 bits

(D) 64 Gbyte, 28 bit

>> The Unix file system uses an extension of indexed allocation. It uses direct blocks, single indirect blocks, double indirect blocks and triple indirect blocks

Q156: Normally user programs are prevented from handling I/O directly by I/O instructions in them. For CPUs having explicit I/O instructions, such I/O protection is ensured by having the I/O instructions privileged. In a CPU with memory mapped I/O, there is no explicit I/O instruction. Which one of the following is true for a CPU with memory mapped I/O?

(A) I/O protection is ensured by operating system routine (s)

(B) I/O protection is ensured by a hardware trap

(C) I/O protection is ensured during system configuration

(D) I/O protection is not possible

>> User applications are not allowed to perform I/O in user mode – All I/O requests are handled through system calls that must be performed in kernel mode.

Q157: 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 First (SSTF), claimed by the vendor to give 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%

>> Since Operating System can execute a single sequential user process at a time, the disk is accessed in FCFS manner always. The OS never has a choice to pick an IO from multiple IOs as there is always one IO at a time

Q158: 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

>> Using larger block size makes disk utilization poorer as more space would be wasted for small data in a block. It may make throughput better as the number of blocks would decrease.

Q159: Which of the following requires a device driver?

(A) Register

(B) Cache

(C) Main memory

(D) Disk

>> A disk driver is software which enables communication between internal hard disk (or drive) and computer.

It allows a specific disk drive to interact with the remainder of the computer.

Q160: Listed below are some operating system abstractions (in the left column) and the hardware components or mechanism (in the right column) that they are abstractions of. Which of the following matching of pairs is correct?

A. Thread	1. Interrupt
B. Virtual address space	2. Memory
C. File system	3. CPU
D. Signal	4. Disk

(A) A-2, B-4, C-3, D-1

(B) A-1, B-2, C-3, D-4

(C) A-3, B-2, C-4, D-1

(D) A-4, B-1, C-2, D-3

>> Thread is handled by CPU.
Virtual address space (it is a set of virtual memory addresses that a process can use) is associated with memory management.
File System is used for disk management.
Interrupt is a type of signal

Q162: The correct matching for the following pairs is

(A) DMA I/O	(1) High speed RAM
(B) Cache	(2) Disk
(C) Interrupt I/O	(3) Printer
(D) Condition Code Register	(4) ALU

Codes:

	A	B	C	D
a	4	3	1	2
b	2	1	3	4
c	4	3	2	1
d	2	3	4	1

(A) a

(B) b

(C) c

(D) d

Q163:In FIFO page replacement algorithm, when a page must be replaced

____

a) oldest page is chosen

b) newest page is chosen

c) random page is chosen

d) none of the mentioned

Q163: Which of the following disk strategies is likely to give the best throughput?

(A) Farthest cylinder next

(B) Nearest cylinder next

(C) First come first served

(D) Elevator algorithm

>> Nearest cylinder next is also known as shortest seek time first and in this algorithm many pages can be accessed in less amount of time.

So throughput will be high so this is optimal algorithm.

Q164: The correct matching for the following pairs is

(A) Disk Scheduling	(1) Round robin
(B) Batch Processing	(2) SCAN
(C) Time sharing	(3) LIFO
(D) Interrupt processing	(4) FIFO

Codes:

	A	B	C	D
a	3	4	2	1
b	4	3	2	1
c	2	4	1	3
d	3	4	3	2

(A) a

(B) b

[✓ (C) c]

(D) d

>> Round-Robin is also called Time-sharing.

Disk Scheduling Algorithms are used to reduce the total seek time of any request. SCAN is one of the Algorithms.

Interrupt processing is LIFO because when we are processing an interrupt, we disable the interrupts originating from lower priority devices so lower priority interrupts cannot be raised. If an interrupt is detected then it means that it has higher priority than currently executing interrupt so this new interrupt will preempt the current interrupt so, LIFO.

Q165: Consider the following five disk five disk access requests of the form (request id, cylinder number) that are present in the disk scheduler queue at a given time.

(P, 155), (Q, 85), (R, 110), (S, 30), (T, 115)

Assume the head is positioned at cylinder 100. The scheduler follows Shortest Seek Time First scheduling to service the requests.

Which one of the following statements is FALSE ?

(A) T is serviced before P

(B) Q is serviced after S, but before T

(C) The head reverses its direction of movement between servicing of Q and P

(D) R is serviced before P

Q166: Consider a storage disk with 4 platters (numbered as 0, 1, 2 and 3), 200 cylinders (numbered as 0, 1, … , 199), and 256 sectors per track (numbered as 0, 1, … 255). The following 6 disk requests of the form [sector number, cylinder number, platter number] are received by the disk controller at the same time:

[120, 72, 2], [180, 134, 1], [60, 20, 0], [212, 86, 3], [56, 116, 2], [118, 16, 1]

Currently head is positioned at sector number 100 of cylinder 80, and is moving towards higher cylinder numbers. The average power dissipation in moving the head over 100 cylinders is 20 milliwatts and for reversing the direction of the head movement once is 15 milliwatts. Power dissipation associated with rotational latency and switching of head between different platters is negligible.

The total power consumption in milliwatts to satisfy all of the above disk requests using the Shortest Seek Time First disk scheduling algorithm is __ .

(A) 45

(B) 80

(C) 85

(D) None of these

>> Head starts at 80.

gate_cs_2018_36(1)

Total Head movements in SSTF = (86-80) + (86-72) + (134-72) + (134-16) = 200

Power dissipated by 200 movements : P1 = 0.2 * 200 = 40 mW

Power dissipated in reversing head direction once = 15 mW

Number of time Head changes its direction = 3

Power dissipated in reversing head direction: P2 = 3 * 15 = 45 mW

Total power consumption (in mW) is P1 + P2 = 40 mW + 45 mW = 85 mW

Q167: Consider a disk queue with requests for I/O to blocks on cylinders 47, 38, 121, 191, 87, 11, 92, 10. The C-LOOK scheduling algorithm is used. The head is initially at cylinder number 63, moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is:

(A) 346

(B) 165

(C) 154

(D) 173

>> The head movement would be:

63 => 87 24 movements
87 => 92 5 movements
92 => 121 29 movements
121 => 191 70 movements
191 --> 10 181 movement
10 => 11 1 movement
11 => 38 27 movements
38 => 47 9 movements
Total head movements = 346

Q168: Suppose the following disk request sequence (track numbers) for a disk with 100 tracks is given: 45, 20, 90, 10, 50, 60, 80, 25, 70. Assume that the initial position of the R/W head is on track 50. The additional distance that will be traversed by the R/W head when the Shortest Seek Time First (SSTF) algorithm is used compared to the SCAN (Elevator) algorithm (assuming that SCAN algorithm moves towards 100 when it starts execution) is ___ tracks

(A) 8

(B) 9

(C) 10

(D) 11

>> In Shortest seek first (SSTF), closest request to the current position of the head, and then services that request next.

In SCAN (or Elevator) algorithm, requests are serviced only in the current direction of arm movement until the arm reaches the edge of the disk. When this happens, the direction of the arm reverses, and the requests that were remaining in the opposite direction are serviced, and so on.

Q169: Consider a main memory with five page frames and the following sequence of page references: 3, 8, 2, 3, 9, 1, 6, 3, 8, 9, 3, 6, 2, 1, 3. Which one of the following is true with respect to page replacement policies First-In-First Out (FIFO) and Least Recently Used (LRU)?

(A) Both incur the same number of page faults

(B) FIFO incurs 2 more page faults than LRU

(C) LRU incurs 2 more page faults than FIFO

(D) FIFO incurs 1 more page faults than LRU

3, 8, 2, 3, 9, 1, 6, 3, 8, 9, 3, 6, 2, 1, 3

In both FIFO and LRU, we get following after considering 3, 8, 2, 3, 9, 1, 3 8 2 9 1

FIFO

6 replaces 3

8 2 9 1 6

3 replaces 8

2 9 1 6 3

8 replaces 2

9 1 6 3 8

2 replaces 9

1 6 3 8 2

No more page faults

LRU

6 replaces 8

3 2 9 1 6

8 replaces 2

3 9 1 6 8

2 replaces 1

3 9 6 8 2

1 replaces 8

3 9 6 2 1

Q170: A system contains three programs and each requires three tape units for its operation. The minimum number of tape units which the system must have such that deadlocks never arise is ____.

(A) 6

(B) 7

(C) 8

(D) 9

>> If 6 resources are there then it may be possible that all three process have 2 resources and waiting for 1 more resource. Therefore, all of them will have to wait indefinitely. If 7 resources are there, then atleast one must have 3 resources so deadlock can never occur.

Q171: A file system with 300 GByte disk uses a file descriptor with 8 direct block addresses, 1 indirect block address and 1 doubly indirect block address. The size of each disk block is 128 Bytes and the size of each disk block address is 8 Bytes. The maximum possible file size in this file system is

(A) 3 Kbytes

(B) 35 Kbytes

(C) 280 Bytes

(D) Dependent on the size of the disk

Q172: An application loads 100 libraries at start-up. Loading each library requires exactly one disk access. The seek time of the disk to a random location is given as 10 ms. Rotational speed of disk is 6000 rpm. If all 100 libraries are loaded from random locations on the disk, how long does it take to load all libraries? (The time to transfer data from the disk block once the head has been positioned at the start of the block may be neglected)

(A) 0.50 s

(B) 1.50 s

(C) 1.25 s

(D) 1.00 s

Q173: 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 1ms 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

>> 4, 34, 10, 7, 19, 73, 2, 15, 6, 20
Since shortest seek time first policy is used, head will first move to 34. This move will cause 16*1 ms. After 34, head will move to 20 which will cause 14*1 ms. And so on. So cylinders are accessed in following order 34, 20, 19, 15, 10, 7, 6, 4, 2, 73 and total time will be (16 + 14 + 1 + 4 + 5 + 3 + 1 + 2 + 2 + 71)*1 = 119 ms.

Q174: A hard disk system has the following parameters :

Number of tracks = 500

Number of sectors/track = 100

Number of bytes /sector = 500

Time taken by the head to move from one track to adjacent track = 1 ms

Rotation speed = 600 rpm.

What is the average time taken for transferring 250 bytes from the disk ?

(A) 300.5 ms

(B) 255.5 ms

(C) 255.0 ms

(D) 300.0 ms

>> Avg. time to transfer = Avg. seek time + Avg. rotational delay + Data transfer time

Avg Seek Time – time taken to move from 1st track to 1sr track : 0ms, 1st to 2nd : 1ms, 2ms, 3ms,….499ms
Avg Seek time =(∑0+1+2+3+…+499)/500 = 249.5 ms
Avg Rotational Delay – RMP : 600 , 600 rotations in 60 sec (one Rotation = 60/600 sec = 0.1 sec) So, Avg Rotational Delay = 0.1/2= 50 ms
Data Transfer Time: In One 1 Rotation we can read data on one track = 100 * 500 = 50,000 B data is read in one rotation. 250 bytes -> 0.1 * 250 / 50,000 = 0.5 ms
Therefore ATT = 249.5+50+0.5 = 300 ms

Q175: Consider a disk pack with a seek time of 4 milliseconds and rotational speed of 10000 rotations per minute (RPM). It has 600 sectors per track and each sector can store 512 bytes of data. Consider a file stored in the disk. The file contains 2000 sectors. Assume that every sector access necessitates a seek, and the average rotational latency for accessing each sector is half of the time for one complete rotation. The total time (in milliseconds) needed to read the entire file is ___.

(A) 14020

(B) 14000

(C) 25030

(D) 15000

Seek time (given) = 4ms

RPM = 10000 rotation in 1 min [60 sec]

So, 1 rotation will be =60/10000 =6ms [rotation speed]

Rotation latency= 1/2 * 6ms=3ms

# To access a file,

total time includes =seek time + rot. latency +transfer time

TO calc. transfer time, find transfer rate

Transfer rate = bytes on track /rotation speed

so, transfer rate = 600*512/6ms =51200 B/ms

transfer time= total bytes to be transferred/ transfer rate

so, Transfer time =2000*512/51200 = 20ms

Given as each sector requires seek tim + rot. latency

= 4ms+3ms =7ms

Total 2000 sector takes = 2000*7 ms =14000 ms

To read entire file ,total time = 14000 + 20(transfer time)= 14020 ms

Seek time (given) = 4ms

RPM = 10000 rotation in 1 min [60 sec]

So, 1 rotation will be =60/10000 =6ms [rotation speed]

Rotation latency= 1/2 * 6ms=3ms

# To access a file,

total time includes =seek time + rot. latency +transfer time

TO calc. transfer time, find transfer rate

Transfer rate = bytes on track /rotation speed

so, transfer rate = 600*512/6ms =51200 B/ms transfer time= total bytes to be transferred/ transfer rate

so, Transfer time =2000*512/51200 = 20ms

Given as each sector requires seek tim + rot. latency

= 4ms+3ms =7ms

Total 2000 sector takes = 2000*7 ms =14000 ms

To read entire file ,total time = 14000 + 20(transfer time)

= 14020 ms

Q176: Consider a paged memory system with logical address space of 256Mbytes and physical address space of 8Gbytes. The page size is 4Kbytes. The total page table size is 256Kbytes. The number of protection bits stored in each page table entry apart from translation is ____ bits?

>> 11

Q177: Which of the following statements is correct?

1.Deadlock recovery using deadlock detection is very fast process

2.Deadlock avoidance algorithm may allow system to be in unsafe state

3.Deadlock prevention is stricter than deadlock avoidance

4.All of the abve

>> All of the above

Q178: In paged memory management, the page sharing is implemented by?

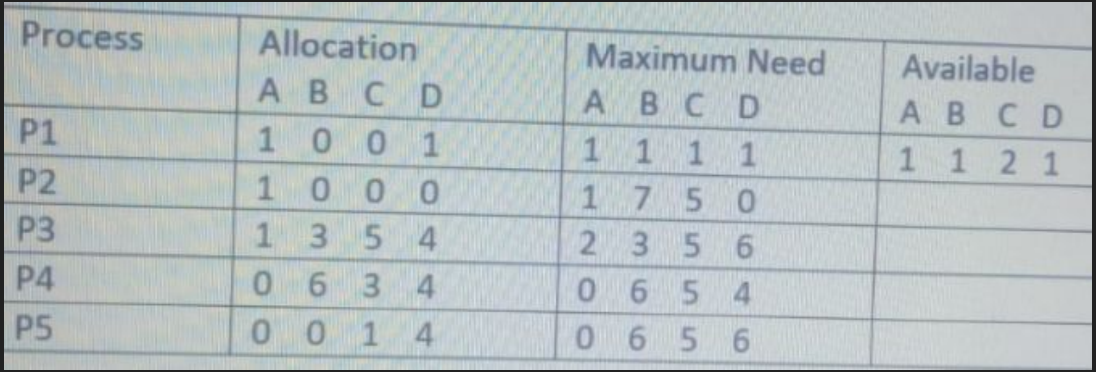
1.Sharing the page tables

2.Loading separate copy of shared page for each process

3.Virtual Memory technique

4.Having some page table entries pointing to same frame in main memory

Q179: A computer system uses Banker's algorithm for deadlock avoidance. The current state of the system is shown as the table below, where P1, P2, P3, P4 and P5 are processes; and A, B, C, D are the resources.



Process	Allocation				Maximum Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P1	1	0	0	1	1	1	1	1	1	1	2	1
P2	1	0	0	0	1	7	5	0				
P3	1	3	5	4	2	3	5	6				
P4	0	6	3	4	0	6	5	4				
P5	0	0	1	4	0	6	5	6				

Which of the following is/are true for the safety of the system?

1.System is in safe state and safe sequence is P1, P3, P4, P5, P2

2.System is in safe state and safe sequence is P1, P3, P4, P2, P5

3.System is in safe state and safe sequence is P4, P5, P3, P2, P1

4.System is in safe state and safe sequence is P4, P1, P3, P5, P2

Q178: Consider a paged memory management system with page size of 2Kbytes. The system uses a 2-way set associative TLB(translation Lookaside Buffer) which can store a total of 4096 page table entries. The TLB controller stores 11-bit tag for each TLB entry. The logical address space is ____ Gbytes?

>> 8

Q180: Consider a paged memory system with TLB. The TLB has hit rate of 90% and access time of 15ns. The main memory access time is 300ns. The effective memory access time is __ns?

>> 345

Q181: Consider a segmented memory management system with logical address of size 32 bits. The segment size may vary from 16 Kbytes to 2Mbytes. The maximum number of segments a process can have is ___ * 2^{10} ?

>> 2

Q182: Which of the following can help in reducing amount of internal fragmentation in paging?

1.Increasing the page size

2.Increasing the allocated number of frames to a process

3.Decreasing the page size

4.Decreasing the allocated number of frames to a process

Q183: Consider a paging system with page size of 4 kbytes(1k=2^10) and virtual address space of 128 Mbytes. Assume that CPU generates the following virtual addresses (all given addresses in decimal):

1256, 5486, 1245, 9721, 9086, 13288, 12167, 14367, 16378, 10221

Consider that system has only 1 frame(which is initially empty) for this process with local page allocation. Number of page faults experienced by the system to serve all above request in given sequence is ____?

>> 8

Q184: A particular disk unit uses a bit string to record the occupancy or vacancy of its disk blocks with ‘1’ denoting vacant block and ‘0’ denoting occupied block. A 1024-bits string contains one time C, three times A, 6 times E, 7 times B and remaining all 0 in hexadecimal representation. The percentage of occupied blocks on the disk for this part (to closest integer) is __%?

>> 95

Q185: Disk requests are made for a disk drive for cylinders 5, 25, 18, 3,39, 8 and 35 in that order. A seek takes 3 milliseconds for 1 disk cylinder movement. Assume that the arm is current at cylinder 20 and the above all requests are serviced using shortest seek time first (SSTF). The seek time needed is __ milliseconds?

>> 177

Q186: Consider a paged virtual memory environment with 4 page-frames in main memory (all initially empty). The system uses FIFO page replacement policy. The system stores dirty bit for each page in main memory to keep a track of modified pages in main memory; and only the page which has dirty bit set, is actually written back to secondary storage if replaced otherwise not. The CPU executes a process in which 70% of total page references have a hit in main memory, 15% of total memory references having page fault with no page replacement needed, 10% of total memory references having page fault with page replacement needed of a non-dirty page and remaining references having page fault with page replacement needed of a dirty page. The memory access time for a reference when the page is available in memory is 200ns. The memory access time is 2000ns when there is a page fault without page replacement. The memory access time is 2500ns when there is a page fault with page replacement of a non-dirty page. The memory access time is 5000ns when there is a page fault with page replacement of a dirty page. The effective memory access time for the process is __ ns?

>> 940

Q187: A main memory has 3 page frames and initially all of them are empty. Consider the following page reference sequence:

1, 2, 1, 9, 4, 1, 5, 4, 3, 9, 6, 3

If the paged system uses FIFO replacement policy, the page hit ratio h for service of the above requests will be ____ % (rounded to nearest integer)?

>> 25

Q188: Which of the following statements is/are correct about file allocation methods?

1.Contiguous file allocation method suffers from external and internal fragmentation both, but linked file allocation method suffers from only internal fragmentation

2.Indexed file allocation method suffers from external and internal fragmentation both, but linked file allocation method suffers from only internal fragmentation

3.Indexed file allocation method can enable file access sequential and random file access both, but linked file allocation can enable sequential file access only

4.Indexed file allocation method can enable file access sequential and random file access both, but contiguous file allocation can enable sequential file access

Q189: A single level directory on a system uses starting 8 blocks to store the directory entries, indexes and other file system content. The disk has total 512 blocks each of size 2Kbyes. The maximum size of any file can be ____ Kbytes?

>> 1008

Q190:If the system has 2 processes in ready state: one is IO bound process and one is CPU bound process. Then which of the following is true?

1.The CPU bound process should be scheduled first for better resource utilization.

2.The IO bound process should be scheduled first for better resource utilization

3.Any process can be scheduled first for better resource utilization.

4.None can be said

Q191: Consider a process scenario with 4 processes A,B,C and D with their respective arrival times and burst times in milliseconds. Processes A,B,C and D arriving at times 0,1,2 and 3 respectively. Processes A,B,C and having their CPU burst time 6,4,3 and 1 respectively. Consider preemptive shortest remaining time first algorithm for scheduling. Use arrival time to break tie if 2 processes have some remaining time. The scheduling happens only on arrival of a new process or on completion of a running process. Each scheduling overhead takes 0.1ms. Further note that scheduling is required only when at least one process is there in ready queue. The total time required to run given 4 processes on single CPU is _____ milliseconds.

>> 14.7

Q192: Consider 4 tasks T1, T2, T3 and T4 and each of them consist an infinite sequence of instances. A new instance of all processes arrive every 16ms. Each instance of T1,T2,T3 and T4 takes 6ms, 5ms, 2ms and 3ms time for execution on CPU. No any task and instance has any IO requirement. The OS uses Non-preemptive SJF policy to run the processes on CPU. The first instance of all process arrive at time 0. The throughput of the CPU for this execution is______instances/ second?

>> 250

Q193: Which of the following are CPU scheduling algorithms?

1. Priority scheduling
2. Round Robin
3. Shortest Job First
4. **All of the above**

Q194: Operating systems :

1.Provides a layer so as to act as a user-friendly interface that enables the programmer to draw a flow chart

2.Links the program with subroutines

3.Helps to create a flow chart of the programs

4.All of these

Q195: A process which is copied from main memory to secondary memory on the basis of requirement is known as

1. **Demand Paging**
2. Paging
3. Threads
4. Segmentation

Q196: FIFO scheduling is a type of:

1. Pre-emptive scheduling
2. **Non Pre-Emptive scheduling**
3. Deadline scheduling
4. None of the above

Q197: Which of the type of OS reads and reacts in terms of actual time?

1. Quick Sharing OS
2. Time Sharing OS
3. **Real time OS**
4. Batch OS

Q198: A systematic procedure for moving the CPU to new process is known as-

1. Synchronization
2. Deadlock
3. Starvation
4. **Context Switching**

Q199: UNIX is written in which language?

1) C#

2) C++

3) C

4) .NET

Q200: Thread is a

1) Light weight process

2) Heavy weight process

3) Multi-process

4) I/0 process

Q201: OS classifies the threads as

1) Mainframe and motherboard level

2) Kernel and User level

3) Security and Memory level

4) OS and CPU level

Q202: Among the following CPU scheduling algorithms, which of these allocated the CPU first to the process that requests the CPU first?

1) FCFS

2) SJF

3) Priority scheduling

4) None

Q203: What are the two types of operating modes of AT?

1) Virtual mode, dedicated mode

2) Private mode, public mode

3) Real mode, protected mode

4) Direct mode, indirect mode

Q204: Which of the following schedules threads?

1) Virtual memory

2) Operating system

3) CPU

4) Input

Q205: What is meant by ready state of a process?

1) When the process is scheduled to run after some execution

2) When the process is currently using the CPU

3) When the process is dependent of the execution time of some other process.

4) None of these

Q206: Among the following, which is an example of a spooled device?

1) A line printer that prints the output of a number of jobs.

2) A terminal that inputs user data

3) A I/O device to display graphics.

4) None

Q207: Main memory of a computer system is?

1) Non-volatile

2) Volatile

3) Restricted

4) Unrestricted

Q208: For which of the following purposes in Banker’s algorithm is used?

1) Preventing deadlock

2) Solving deadlock

3) Recover from deadlock

4) None

Q209: Device driver required in?

1) Register

2) Main memory

3) Disk

4) Cache

Q210: When are the register context and stack of thread deallocated?

1) when the thread terminates

2) when the thread blocks

3) when the thread unblocks

4) when the thread spawns

ept

Q211: Threads is not shared among which of the following?

1) stack

2) program counter

3) both program counter and stack

4) none

Q212: For which of the following is the jacketing technique used?

1) to construct a new thread

2) to communicate between threads

3) convert a blocking system call into non blocking system call

4) None

Q213: For which of the following is resource sharing used?

1) an application having several threads of activity all within the same address space.

2) share the memory and resources of the process to which the threads belong

3) Compress the address space a process can use

4) all of the mentioned

Q214: Many to One model is at an advantage in which of the following conditions?

1) When the program needs to be multi-threaded

2) When there is a single processor present

3) When the program does not need multithreading

4) None

Q215: Identify the system calls that on termination does not return control to the calling point.

1) exec

2) fork

3) longjmp

4) ioctl

Q216: Consider the following program:

```
main()
{
     if(fork()>0)
sleep(100);
}
```

1) infinite process

2) orphan process

3) zombie process

4) none

Q217: The output of the following C program is?

```
int main(){
     fork();
    fork();
    printf("code ");
}
```

1) code code code code

2) code code code

3) code code

4) code

Q218 Identify the call which never returns an error?

1) fork

2) getpid

3) ioctl

4) open

Q219: What of the following defines Thread cancellation?

1) The process of terminating a thread process before its execution

2) The process of removing a thread after its work is executed

3) The process of destroying the thread after its work is executed

4) none

Q220: When a thread terminates some target thread immediately, it is known as?

1) Immediate Termination

2) Asynchronous termination

3) Synchronous termination

4) Deferred cancellation

Q221: Signals of some given type are

1) sent together

2) queued

3) stacked

4) none

Q222: Which of the following commands in UNIX is used to send a signal?

1) send

2) kill

3) sigsend

4) none

Q223. Because of virtual memory, the memory can be shared among

____

a) processes

b) threads

c) instructions

d) none of the mentioned

Q224:______ is the concept in which a process is copied into the main memory from the secondary memory according to the requirement.

a) Paging

b) Demand paging

c) Segmentation

d) Swapping

Q225:When a program tries to access a page that is mapped in address space but not loaded in physical memory, then

____

a) segmentation fault occurs
b) fatal error occurs
c) page fault occurs
d) no error occurs

ept

Q226:Effective access time is directly proportional to ________________

a) page-fault rate

b) hit ratio

c) memory access time

d) none of the mentioned

Q227:Virtual memory allows ________________

a) execution of a process that may not be completely in memory

b) a program to be smaller than the physical memory

c) a program to be larger than the secondary storage

d) execution of a process without being in physical memory

Q228:The ability to execute a program that is only partially in memory has benefits like ______________

a) The amount of physical memory cannot put a constraint on the program

b) Programs for an extremely large virtual space can be created

c) Throughput increases

d) All of the mentioned

ept

Q229: If the memory access time is denoted by ‘ma’ and ‘p’ is the probability of a page fault (0 <= p <= 1). Then the effective access time for a demand paged memory is _______________

a) p x ma + (1-p) x page fault time
b) ma + page fault time
c) (1-p) x ma + p x page fault time
d) none of the mentioned

Q230:Locality of reference implies that the page reference being made by a process ________________
a) will always be to the page used in the previous page reference
b) is likely to be one of the pages used in the last few page references
c) will always be one of the pages existing in memory
d) will always lead to page faults

Q231:A process refers to 5 pages, A, B, C, D, E in the order : A, B, C, D, A, B, E, A, B, C, D, E. If the page replacement algorithm is FIFO, the number of page transfers with an empty internal store of 3 frames is?

a) 8

b) 10

c) 9

d) 7

Q232: A process refers to 5 pages, A, B, C, D, E in the order : A, B, C, D, A, B, E, A, B, C, D, E. If the page replacement algorithm is FIFO, the number of page frames is increased to 4, then the number of page transfers ______________

a) decreases

b) increases

c) remains the same

d) none of the mentioned

Q233: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 programs do not exhibit locality of reference
Which of the following is TRUE?
a) Both P and Q are true, 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

Q234:The aim of creating page replacement algorithms is to ______________

a) replace pages faster

b) increase the page fault rate

c) decrease the page fault rate

d) to allocate multiple pages to processes

Q235:What is the Optimal page – replacement algorithm?
a) Replace the page that has not been used for a long time
b) Replace the page that has been used for a long time
c) Replace the page that will not be used for a long time
d) None of the mentioned

Optimal page – replacement algorithm is difficult to implement, because ________________

a) it requires a lot of information

b) it requires future knowledge of the reference string

c) it is too complex

d) it is extremely expensive

Q236:For 3 page frames, the following is the reference string:
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
How many page faults does the LRU page replacement algorithm produce?
a) 10
b) 15
c) 11
d) 12

Q237:What are the two methods of the LRU page replacement policy that can be implemented in hardware?

a) Counters

b) RAM & Registers

c) Stack & Counters

d) Registers

Q238:Applying the LRU page replacement to the following reference string.

1 2 4 5 2 1 2 4

The main memory can accommodate 3 pages and it already has pages 1 and 2. Page 1 came in before page 2.

How many page faults will occur?

a) 2

b) 3

c) 4

d) 5

Q239:Increasing the RAM of a computer typically improves performance because ________________

a) Virtual memory increases

b) Larger RAMs are faster

c) Fewer page faults occur

d) None of the mentioned

Q240:What is the reason for using the MFU page replacement algorithm?

a) an actively used page should have a large reference count

b) a less used page has more chances to be used again

c) it is extremely efficient and optimal

d) all of the mentioned

Q241:The algorithm in which we split m frames among n processes, to give everyone an equal share, m/n frames is known as ________________

a) proportional allocation algorithm

b) equal allocation algorithm

c) split allocation algorithm

d) none of the mentioned

Q242:In the working set model, for:
2 6 1 5 7 7 7 7 5 1 6 2 3 4 1 2 3 4 4 4 3 4 3 4 4 4 1 3 2 3
if DELTA = 10, then the working set at time t1 (….7 5 1) is?

a) {1, 2, 4, 5, 6}
b) {2, 1, 6, 7, 3}
c) {1, 6, 5, 7, 2}
d) {1, 2, 3, 4, 5}

Q243:Consider the following page reference string.
1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6
For LRU page replacement algorithm with 4 frames, the number of page faults is?

a) 10
b) 14
c) 8
d) 11

Q244:Consider the following page reference string.
1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6
For Optimal page replacement algorithms with 3 frames, the number of page faults is?

a) 16
b) 15
c) 14
d) 11

Q245:To create a file _______________

a) allocate the space in file system

b) make an entry for new file in directory

c) allocate the space in file system & make an entry for new file in directory

d) none of the mentioned

Q246:Which file is a sequence of bytes organized into blocks understandable by the system’s linker?

a) object file

b) source file

c) executable file

d) text file

Q247:Which one of the following explains the sequential file access method?

a) random access according to the given byte number

b) read bytes one at a time, in order

c) read/write sequentially by record

d) read/write randomly by record

Q248:When will file system fragmentation occur?

a) unused space or single file are not contiguous

b) used space is not contiguous

c) unused space is non-contiguous

d) multiple files are non-contiguous

Q249:Management of metadata information is done by ______________

a) file-organisation module

b) logical file system

c) basic file system

d) application programs

Q250:What will happens when a process closes the file?

a) per-process table entry is not removed

b) system wide entry’s open count is decremented

c) all of the mentioned

d) none of the mentioned

Q251:In which type of allocation method each file occupy a set of contiguous block on the disk?

a) contiguous allocation

b) dynamic-storage allocation

c) linked allocation

d) indexed allocation

Q252:In which type of allocation method each file occupy a set of contiguous block on the disk?

a) contiguous allocation

b) dynamic-storage allocation

c) linked allocation

d) indexed allocation

Q253:File attributes consist of ______________

a) name

b) type

c) identifier

d) all of the mentioned

Q254:The information about all files is kept in

____

a) swap space
b) operating system
c) seperate directory structure
d) none of the mentioned

Q256:The operating system keeps a small table containing information about all open files called

____

a) system table
b) open-file table
c) file table
d) directory table

Q257:What will happen in the two level directory structure?

a) each user has his/her own user file directory

b) the system doesn’t its own master file directory

c) all of the mentioned

d) none of the mentioned

Q258:What will happen in the single level directory?

a) all directories must have unique names

b) all files must have unique names

c) all files must have unique owners

d) all of the mentioned

Q259: When a user job starts in a two level directory system, or a user logs in ________________

a) the users user file directory is searched

b) the system’s master file directory is not searched

c) the master file directory is indexed by user name or account number, and each entry points to the UFD for that user

d) all of the mentioned

Q260:What is the disadvantage of the two level directory structure?
a) it does not solve the name collision problem
b) it solves the name collision problem
c) it does not isolate users from one another
d) it isolates users from one another

Q261:Which of the following are the types of Path names?

a) absolute & relative

b) local & global

c) global & relative

d) relative & local

Q262:In a tree structure, when deleting a directory that is not empty?

a) The contents of the directory are safe

b) The contents of the directory are also deleted

c) contents of the directory are not deleted

d) none of the mentioned

Q263:A situation where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which access takes place is called

____

a) data consistency
b) race condition
c) aging
d) starvation

Q264:The segment of code in which the process may change common variables, update tables, write into files is known as _______________

a) program

b) critical section

c) non – critical section

d) synchronizing

Q265:Mutual exclusion implies that ______________

a) if a process is executing in its critical section, then no other process must be executing in their critical sections

b) if a process is executing in its critical section, then other processes must be executing in their critical sections

c) if a process is executing in its critical section, then all the resources of the system must be blocked until it finishes execution

d) none of the mentioned

Q266:Bounded waiting implies that there exists a bound on the number of times a process is allowed to enter its critical section ______________

a) after a process has made a request to enter its critical section and before the request is granted

b) when another process is in its critical section

c) before a process has made a request to enter its critical section

d) none of the mentioned

Q267:TestAndSet instruction is executed ______________

a) after a particular process
b) periodically
c) atomically
d) none of the mentioned

Q268:Semaphore is a/an _________ to solve the critical section problem.

a) hardware for a system

b) special program for a system

c) integer variable

d) none of the mentioned

Q269:The signal operation of the semaphore basically works on the basic _________ system call.

a) continue()

b) wakeup()

c) getup()

d) start()

Q270:If the semaphore value is negative

____

a) its magnitude is the number of processes waiting on that semaphore

b) it is invalid

c) no operation can be further performed on it until the signal operation is performed on it

d) none of the mentioned

Q271:Each process Pi, i = 0,1,2,3,……,9 is coded as follows.

```
repeat P(mutex)
{Critical Section}
V(mutex)
forever
```

The code for P10 is identical except that it uses V(mutex) instead of P(mutex). What is the largest number of processes that can be inside the critical section at any moment (the mutex being initialized to 1)?

a) 1

b) 2

c) 3

d) None of the mentioned

>>Any one of the 9 processes can get into critical section after executing P(mutex) which decrements the mutex value to 0. At this time P10 can enter critical section by incrementing the value to 1. Now any of the 9 processes can enter the critical section by again decrementing the mutex value to 0. None of the remaining processes can get into their critical sections.

Q272:At a particular time of computation the value of a counting semaphore is 7.Then 20 P operations and 15 V operations were completed on this semaphore. The resulting value of the semaphore is?

a) 42

b) 2

c) 7

d) 12

>>P represents Wait and V represents Signal. P operation will decrease the value by 1 every time and V operation will increase the value by 1 every time.

Q273:The program follows to use a shared binary semaphore T.

```
Process A int Y;
 A1: Y = X*2;
A2: X = Y;
signal(T);
Process B
int Z;
B1: wait(T);
B2: Z = X+1;
X = Z;
```

T is set to 0 before either process begins execution and, as before, X is set to 5.
Now, how many different values of X are possible after both processes finish executing?

a) one
b) two
c) three
d) four

>>The semaphore T ensures that all the statements from A finish execution before B begins. So now there is only one way in which statements from A and B can be interleaved:

A1 A2 B1 B2: X = 11.

Q274:In the bounded buffer problem, there are the empty and full semaphores that _______________

a) count the number of empty and full buffers

b) count the number of empty and full memory spaces

c) count the number of empty and full queues

d) none of the mentioned

Q275:The bounded buffer problem is also known as ____________

a) Readers – Writers problem
b) Dining – Philosophers problem
c) Producer – Consumer problem
d) None of the mentioned

Q276:A deadlock free solution to the dining philosophers problem ________________

a) necessarily eliminates the possibility of starvation

b) does not necessarily eliminate the possibility of starvation

c) eliminates any possibility of any kind of problem further

d) none of the mentioned

Q277:All processes share a semaphore variable **mutex**, initialized to 1. Each process must execute wait(mutex) before entering the critical section and signal(mutex) afterward.
Suppose a process executes in the following manner.

```
wait(mutex);
 .....
critical section
 .....
wait(mutex);
```

a) a deadlock will occur
b) processes will starve to enter critical section
c) several processes maybe executing in their critical section
d) all of the mentioned

Q278:The number of resources requested by a process

____

a) must always be less than the total number of resources available in the system
b) must always be equal to the total number of resources available in the system
c) must not exceed the total number of resources available in the system
d) must exceed the total number of resources available in the system

Q279:For a deadlock to arise, which of the following conditions must hold simultaneously?

a) Mutual exclusion

b) No preemption

c) Hold and wait

d) All of the mentioned

Q280:For a Hold and wait condition to prevail ______________

a) A process must be not be holding a resource, but waiting for one to be freed, and then request to acquire it

b) A process must be holding at least one resource and waiting to acquire additional resources that are being held by other processes

c) A process must hold at least one resource and not be waiting to acquire additional resources

d) None of the mentioned

Q281:Each request requires that the system consider the ________________ to decide whether the current request can be satisfied or must wait to avoid a future possible deadlock.

a) resources currently available

b) processes that have previously been in the system

c) resources currently allocated to each process

d) future requests and releases of each process

Q282:A system has 12 magnetic tape drives and 3 processes : P0, P1, and P2. Process P0 requires 10 tape drives, P1 requires 4 and P2 requires 9 tape drives.

Process P0

P1

P2

Maximum needs (process-wise: P0 through P2 top to bottom) 10

4

9

Currently allocated (process-wise)

5

2

2

Which of the following sequence is a safe sequence?

a) P0, P1, P2

b) P1, P2, P0

c) P2, P0, P1

d) P1, P0, P2

Q283:The Banker’s algorithm is ________________ than the resource allocation graph algorithm.

a) less efficient

b) more efficient

c) equal

d) none of the mentioned

Q284:What is the disadvantage of invoking the detection algorithm for every request?

a) overhead of the detection algorithm due to consumption of memory

b) excessive time consumed in the request to be allocated memory

c) considerable overhead in computation time

d) all of the mentioned

Q285:A computer system has 6 tape drives, with ‘n’ processes competing for them. Each process may need 3 tape drives. The maximum value of ‘n’ for which the system is guaranteed to be deadlock free is?

a) 2

b) 3

c) 4

d) 1

Q286:A system has 3 processes sharing 4 resources. If each process needs a maximum of 2 units then, deadlock _______________

a) can never occur

b) may occur

c) has to occur

d) none of the mentioned

Q287:‘m’ processes share ‘n’ resources of the same type. The maximum need of each process doesn’t exceed ‘n’ and the sum of all their maximum needs is always less than m+n. In this setup, deadlock _______________

a) can never occur

b) may occur

c) has to occur

d) none of the mentioned

Q288:Those processes should be aborted on occurrence of a deadlock, the termination of which?

a) is more time consuming

b) incurs minimum cost

c) safety is not hampered

d) all of the mentioned

Q289:The process to be aborted is chosen on the basis of the following factors?

a) priority of the process

b) process is interactive or batch

c) how long the process has computed

d) all of the mentioned

Q290:What is the solution to starvation?

a) the number of rollbacks must be included in the cost factor

b) the number of resources must be included in resource preemption

c) resource preemption be done instead

d) all of the mentioned

Q291:If the process can be moved during its execution from one memory segment to another, then binding must be _______________

a) delayed until run time

b) preponed to compile time

c) preponed to load time

d) none of the mentioned

Q292:In a system that does not support swapping

____

a) the compiler normally binds symbolic addresses (variables) to relocatable addresses
b) the compiler normally binds symbolic addresses to physical addresses
c) the loader binds relocatable addresses to physical addresses
d) binding of symbolic addresses to physical addresses normally takes place during execution

Q293:If a higher priority process arrives and wants service, the memory manager can swap out the lower priority process to execute the higher priority process. When the higher priority process finishes, the lower priority process is swapped back in and continues execution. This variant of swapping is sometimes called?

a) priority swapping

b) pull out, push in

c) roll out, roll in

d) none of the mentioned

Q294:Which of the following is TRUE?
a) Overlays are used to increase the size of physical memory
b) Overlays are used to increase the logical address space
c) When overlays are used, the size of a process is not limited to the size of the physical memory
d) Overlays are used whenever the physical address space is smaller than the logical address space

Q295:CPU fetches the instruction from memory according to the value of _______________

a) program counter

b) status register

c) instruction register

d) program status word

Q296:Memory management technique in which system stores and retrieves data from secondary storage for use in main memory is called?
a) fragmentation
b) paging
c) mapping
d) none of the mentioned

Q297:The address of a page table in memory is pointed by _______________

a) stack pointer

b) page table base register

c) page register

d) program counter

Q298:The operating system and the other processes are protected from being modified by an already running process because ______________

a) they are in different memory spaces
b) they are in different logical addresses
c) they have a protection algorithm
d) every address generated by the CPU is being checked against the relocation and limit registers

Q299:A solution to the problem of external fragmentation is ______________

a) compaction

b) larger memory space

c) smaller memory space

d) none of the mentioned

Q300:When the memory allocated to a process is slightly larger than the process, then ________________

a) internal fragmentation occurs

b) external fragmentation occurs

c) both internal and external fragmentation occurs

d) neither internal nor external fragmentation occurs

Q301.Which of the following actions is/are typically not performed by the operating system when switching context from process to process ?

A. Saving current register values and restoring saved register values for process .

B. Changing address translation tables.

C. Swapping out the memory image of process to the disk.

D. Invalidating the translation look-aside buffer.

Q302.Which of the following need not necessarily be saved on a context switch between

A. General purpose registers

B. Translation look-aside

C. Program counter

D. All of the above

Q 303.Let the time taken to switch from user mode to kernel mode of execution be T1 while time taken to switch between two user processes be T2 . Which of the following is correct?

A.T1>T2

B. T1<T2

C.T1=T2

D. Nothing can be said about the relation between and

Q304. System calls are usually invoked by using

A. a software interrupt
B. polling
C. an indirect jump
D. a privileged instruction

Q305.A CPU has two modes -- privileged and non-privileged. In order to change the mode from privileged to non-privileged

A. a hardware interrupt is needed
B. a software interrupt is needed
C. a privileged instruction (which does not generate an interrupt) is needed
D. a non-privileged instruction (which does not generate an interrupt) is needed

Q306. A user level process in Unix traps the signal sent on a Ctrl-C input, and has a signal handling routine that saves appropriate files before terminating the process. When a Ctrl-C input is given to this process, what is the mode in which the signal handling routine executes?

A. User mode
B. Kernel mode
C. Supervisor mode
D. Privileged mode

Q307.Which of the following does not interrupt a running process?

A. A device
B. Timer
C. Scheduler process
D. Power failure

Q308.Which combination of the following features will suffice to characterize an OS as a multi-programmed OS?

a. 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)

Q309. Which of the following standard library functions will always invoke a system call when executed from a single-threaded process in a operating system?

A.exit() B.malloc() C.sleep() D.strlen()

Q310. Which one of the following statements is FALSE?

A. The TLB performs an associative search in parallel on all its valid entries using page number of incoming virtual address.

B. If the virtual address of a word given by CPU has a TLB hit, but the subsequent search for the word results in a cache miss, then the word will always be present in the main memory.

C. The memory access time using a given inverted page table is always same for all incoming virtual addresses.

D. In a system that uses hashed page tables, if two distinct virtual addresses and map to the same valuewhile hashing, then the memory access time of these addresses will not be the same.

Q311.Dynamic linking can cause security concerns because

A. Security is dynamic
B. The path for searching dynamic libraries is not known till runtime
C. Linking is insecure
D. Cryptographic procedures are not available for dynamic linking

Q312. I/O redirection

A. implies changing the name of a file
B. can be employed to use an existing file as input file for a program
C. implies connecting programs through a pipe
D. None of the above