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

(a) $\min(x_p, x_q) < \max_{k \neq p, q} y_k$
(b) $x_p + x_q \geq \max_{k \neq p, q} y_k$
(c) $\min(x_p, x_q) < 1$
(d) $\min(x_p, x_q) > 1$

[2006 : 2 M]

3.8 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) P1
(c) P2
(d) None of the above, since the system is in a deadlock

[2007 : 2 M]

3.9 The following is a code with two threads, producer and consumer, that can run in parallel. Further, S and Q are binary semaphores equipped with the standard P and V operations.

```
semaphore S = 1, Q = 0;
integer x ;
producer :                consumer :
while (true) do           While (true) do
P(S) ;                    P(Q) ;
x = produce( ) ;          consume (x);
V(Q) ;                    V(S) ;
done                      done
```

Which of the following is TRUE about the program above?

(a) The process can deadlock
(b) One of the threads can starve
(c) Some of the items produced by the producer may be lost
(d) Values generated and stored in 'x' by the producer will always be consumed before the producer can generate a new value

[2008 : 2 M]

3.10 An operating system implements a policy that requires a process to release all resources before making a request for another resource. Select the TRUE statement from the following:

(a) Both starvation and deadlock can occur
(b) Starvation can occur but deadlock cannot occur
(c) Starvation cannot occur but deadlock can occur
(d) Neither starvation nor deadlock can occur

[2008 : 2 M]

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

(a) In deadlock prevention, the request for resources is always granted if the resulting state is safe
(b) In deadlock avoidance, the request for resources is always granted if the resulting state is safe
(c) Deadlock avoidance is less restrictive than deadlock prevention
(d) Deadlock avoidance requires knowledge of resource requirements a priori

[2008 : 2 M]

3.12 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 resources as follows if executed independently.

Process P1 :	Process P2 :	Process P3 :
t = 0 : requests 2 units of R2	t = 0 : requests 2 units of R3	t = 0 : requests 1 unit of R4
t = 1 : requests 1 unit of R3	t = 2 : requests 1 unit of R4	t = 2 : requests 2 units of R1
t = 3 : requests 2 units of R1	t = 4 : requests 1 unit of R1	t = 5 : releases 2 units of R1
t = 5 : releases 1 unit of R2 and 1 unit of R1	t = 6 : releases 1 unit of R3	t = 7 : requests 1 unit of R2
t = 7 : releases 1 unit of R3	t = 8 : Finishes	t = 8 : requests 1 unit of R3
t = 8 : requests 2 units of R4		t = 9 : Finishes
t = 10 : Finishes		

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

(a) All processes will finish without any deadlock
(b) Only P1 and P2 will be in deadlock
(c) Only P1 and P3 will be in deadlock
(d) All three processes will be in deadlock

[2009 : 2 M]

3.13 A system has n resources $R_0, R_1, \ldots, R_{n-1}$, and k processes $P_0, P_1, \ldots, P_{k-1}$. The implementation of the resource request logic of each process P_i, is as follows:

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

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

(a) n = 40, k = 26 (b) n = 21, k = 12
(c) n = 20, k = 10 (d) n = 41, k = 19

[2010 : 2 M]

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

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

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

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.

[2014 (Set-1) : 2 M]

3.15 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 ___.

[2014 (Set-3) : 2 M]

3.16 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

[2015 (Set-2) : 1 M]

3.17 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

[2015 (Set-3) : 2 M]

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

Process	Current Allocation	Maximum Requirement
P1	3	7
P2	1	6
P3	3	5

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

(a) Safe, Deadlocked
(b) Safe, Not Deadlocked
(c) Not Safe, Deadlocked
(d) Not Safe, Not Deadlocked

[2017 (Set-2) : 2 M]

3.19 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 _____. **[2018 : 1 M]**

3.20 In a system, there are three types of resources: E, F and G. Four processes P_0, P_1, P_2 and P_3 execute concurrently. At the outset, the processes have declared their maximum resource requirements using a matrix named Max as given below. For example Max[P_2, F] is the maximum number of instances of F that P_2 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

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

[2018 : 2 M]

3.21 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.

(c) If 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.

[2022 : 1 M]

Answers Deadlock

3.1	(c)	3.2	(b)	3.3	(c)	3.4	(a)	3.5	(c)	3.6	(a)	3.7	(b)	3.8	(c)	3.9	(d)
3.10	(b)	3.11	(a)	3.12	(a)	3.13	(b)	3.14	(b)	3.15	(7)	3.16	(*)	3.17	(d)	3.18	(b)
3.19	(2)	3.20	(a)	3.21	(a, d)												

Explanations Deadlock

3.1 (c)

There are 3 processes, and demands '2' resources. If any one gets equal resources as it's demand, deadlock won't occur, and according to this question 4 resources are enough.

3.2 (b)

Every process needs 2 resources. With 5 resources atleast one can get whole 2 and executes. So after execution it frees 2 resources which can be used by other processes.

P_1	P_2	P_3	P_4	P_5
1	1	1	1	1

3.3 (c)

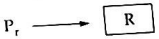
It is not a valid deadlock prevention scheme.

3.4 (a)

Given a deadlock prevention scheme, where each process is assigned a unique time stamp and if killed, it get restarted with same time stamp.

P_h : Process holding resource R.

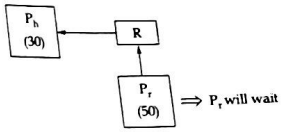
P_r : Process requesting a resource R.



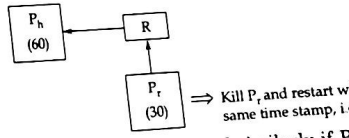
Now, consider the following scenario, where P_1, P_2, ... P_n are n-processes with given time stamp.

Process	P_1	P_2	P_3	P_4	P_5	.....	P_n
Time stamp	10	20	30	40	50	.....	.....

Case-I: Let P_3 is holding resource R
($\therefore P_h = P_3$) and P_5 is requesting for resource R
(so, $P_r = P_5$)
$\therefore$ Condition
if $T(P_r) < T(P_h)$
i.e. $T(P_5) < T(P_3)$
$50 < 30$ // False
Hence P_5 will wait for P_3.



Case-II: Now consider P_3 is requesting for resource R currently hold by process P_5, so checking condition.
if $T(P_r) < T(P_h)$
i.e. $T(P_3) < T(P_5)$
$30 < 50$ // True
$\therefore$ Kill P_r
So, P_3 will be killed and P_5 get resource R now suppose after sometime.
Process P_3 again requesting for resource R let now R is hold by P_6
Again checking
if $T(P_r) < T(P_h)$
$30 < 60$ // True
$\Rightarrow$ P_3 will get killed



P_3 will be killed again and similarly if P_7, P_8 ..... so on keep on arriving and holding resource R then innocent process P_3 may go to indefinite waiting which lead to starvation.
So, option (a) is answer.

3.5 (c)

Total number of resources shared by process P_1, ... P_n = m
Maximum resource requirement of process $P_i = S_i$
i.e.

Process	Requirement
P_1	S_1
P_2	S_2
P_n	S_n

So the worst case allocation of resources for deadlock prevention. So, that deadlock do not occur is:
[Allot the resources to the processes, one less than the actual requirement (worst case deadlock will occur) + 1 (for no deadlock)]
So , we have
$(S_1 - 1) + (S_2 - 1) \ldots (S_n - 1) + 1 \le m$

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

$$\sum_{i=1}^{n} S_i - \sum_{i=1}^{n} 1 + 1 \le m$$

$$\sum_{i=1}^{n} S_i - n + 1 \le m$$

$$\sum_{i=1}^{n} S_i \le m + n - 1$$

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

3.6 (a)

Suppose the slots are full → F = 0.
Now, if Wait(F) and Wait(S) are interchanges and Wait(S) succeeds, The producer will wait for Wait(F) which is never going to succeed as Consumer would be waiting for Wait(S).
So, deadlock can happen.
If Signal(S) and Signal(F) are interchanged in Consumer, deadlock won't happen.
It will just give priority to a producer compared to the next consumer waiting.

3.7 (b)

Total number of processes = n
Process i is holding x_i instances of a resource R for $1 \le i \le n$.
Process i holding resource x_i and additional required y_i. There are two processes p and q such that $y_p = y_q = 0$.
For no dead lock can occur

$$x_p + x_q \ge (\max_{k \ne p,q} y_k)$$

3.8 (c)

Consider the matrix

	alloc XYZ	request XYZ
P_0	121	103
P_1	201	012
P_2	221	120

So the available matrix is (0, 1, 2) and the requirement of process P_1 is (0, 1, 2) so P_1 can use the available resources after releasing the resources the matrix becomes:

	alloc XYZ	request XYZ
P_0	121	103
P_1	000	000
P_2	221	120

In this time available matrix is (2 1 3) and required matrix of P_0 is (1 0 3) so P_0 uses the resources of the releasing the resource. The available matrix is (3, 3, 4) so P_2 will use the resources, P_2 will finish last.

3.9 (d)

Consumer can consume only once the producer has produced the item, and producer can produce (except the first time) only once the consumer has consumed the item.

3.10 (b)

Releasing all resources before making a request for another resource is a technique of deadlock prevention. So, deadlock cannot occur.
However, we cannot say anything about starvation, it may occur or may not.

3.11 (a)

In deadlock prevention, the request for a resource may not be granted even if the resulting state is safe.

3.12 (a)

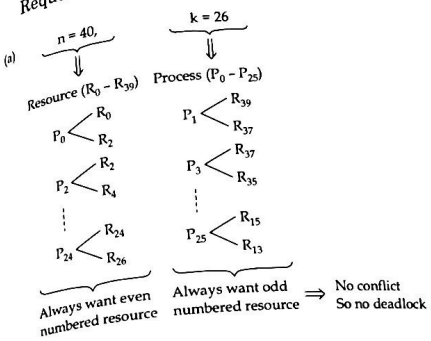
t	R1(3)	R2(2)	R3(3)	R1(2)
0	3	0	1	1
1	3	0	0	1
2	1	0	0	0
3	1	0	0	0
4	0	0	0	0
5	1	1	0	0
6	1	1	1	0
7	1	0	2	0
8	2	0	2	1
9	2	1	3	0
10				

3.13 (b)

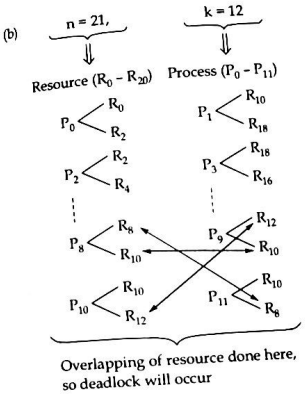
Logic in question:
If both odd and even type of process need same resource, then deadlock may occur.

For even processes:
Request : R_i and R_{i+2}
For odd processes:
Request : R_{n-1} and R_{n-i-2}

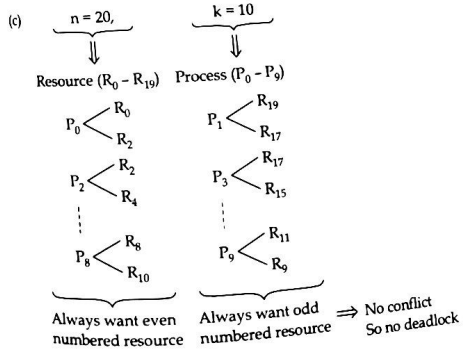
(a)



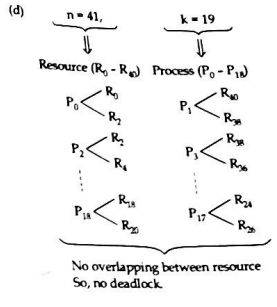
(b)



(c)



(d)



3.14 (b)

Allocation	Max	Need	Need with additional requests
X Y Z	X Y Z	X Y Z	X Y Z
P0 0 0 1	8 4 3	8 4 2	8 4 4 (REQ1)
P1 3 2 0	6 2 0	3 0 0	5 0 0 (REQ2)
P2 2 1 1	3 3 3	1 2 2	1 2 2

Available: X Y Z
3 2 2 → P2 can satisfy it needs
2 1 1 → P2 releases its allocation
5 3 3 → P1 satisfies it needs
3 2 0 → P1 releases its allocation
8 5 3 → P0 can not satisfy with additional requests

∴ Only REQ2 can be permitted.

3.15 (7)

Number of programs = 3
(P_1, P_2, P_3)
Tape required per program = 3
Allocating tape units, so that deadlock does not occur:

$$\underbrace{(3-1)}_{P_1} + \underbrace{(3-1)}_{P_2} + \underbrace{(3-1)}_{P_3} + 1$$

$$\Downarrow$$

7

Answer is 7.

3.16 (*)

N is number of processes, 6 identical resources
$N \ge 6 \times (2 - 1)$
If $N \ge 6 \Rightarrow$ It could lead to a deadlock
No option matches with $N \ge 6$

3.17 **(d)**

I. Dissatisfying hold and wait for deadlock prevention.
II. Dissatisfying circular wait for deadlock prevention.
III. Dissatisfying circular wait for deadlock prevention.
IV. Dissatisfying circular wait for deadlock prevention.

3.18 **(b)**

P.id	Current Allocation	Maximum Requirement	Remaining need	Current available
P1	3	7	4	9 – 7 = 2
P2	1	6	5	5
P3	3	5	2	8
				9

Safe sequence ⇒ P3 → P1 → P2
Safe and Not deadlocked

3.19 **(2)**

P_1 P_2 P_3
| | |
|

No deadlock

Maximum each process can request for 2 resources so, that there will not be any deadlock, because we, have only 4 resource available.
So, K value is '2'.

3.20 **(a)**

	Max need			Current allocation			Current available			Remaining need		
	E	F	G	E	F	G	E(3)	F(3)	G(0)	E	F	G
P_0	4	3	1	1	0	1	4	3	1	3	3	0
P_1	2	1	4	1	1	2	5	3	4	1	0	2
P_2	1	3	3	1	0	3	6	4	6	0	3	0
P_3	5	4	1	2	0	0	8	4	6	3	4	1

Safe sequence: P_0, P_2, P_1, P_3
Hence, system is in safe state.

3.21 **(a, d)**

(a) Circular wait is one of the necessary condition for the formation of deadlock hence it is true.
(b) If the resource has more than one instance then cycle is just a necessary condition but not the sufficient condition, hence it is false.
(c) Unsafe state will not always leads to deadlock because it depends on behaviour of the processes, hence it is false.
(d) If every edge is only allocation, it means there is no requirement of the resources, so deadlock not possible, hence it is true.

■■■■

4 Memory Management and Virtual Memory

4.1 In a paged segmented scheme of memory management, the segment table itself must have a page table because
(a) the segment table is often too large to fit in one page
(b) each segment is spread over a number of pages
(c) segment tables point to page table and not to the physical locations of the segment
(d) the processor's description base register points to a page table **[1995 : 1 M]**

4.2 The capacity of a memory unit is defined by the number of words multiplied by the number of bits/word. How many separate address and data lines are needed for a memory of 4 K × 16?
(a) 10 address, 16 data lines
(b) 11 address, 8 data lines
(c) 12 address, 16 data lines
(d) 12 address, 12 data lines **[1995 : 2 M]**

4.3 The address sequence generated by tracing a particular program executing in a pure demand paging system with 100 records per page with 1 free main memory frame is recorded as follows. What is the number of page faults?
0100, 0200, 0430, 0499, 0510, 0530, 0560, 0120, 0220, 0240, 0260, 0320, 0370
(a) 13 (b) 8
(c) 7 (d) 10 **[1995 : 2 M]**

4.4 In a virtual memory system the address space specified by the address lines of the CPU must be ____ than the physical memory size and ____ than the secondary storage size.
(a) smaller, smaller (b) smaller, larger
(c) larger, smaller (d) larger, larger **[1995 : 2 M]**

4.5 A ROM is used to store the table for multiplication of two 8-bit unsigned integers. The size of ROM required is
(a) 256 × 16 (b) 64 K × 8
(c) 4K × 16 (d) 64 K × 16 **[1996 : 1 M]**

4.6 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 **[1996 : 2 M]**

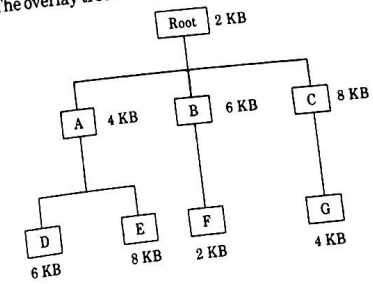
4.7 Locality of reference implies that the page reference being made by a process
(a) will always be to the page used in the previous page reference
(b) is likely to be to one of the pages used in the last few page references
(c) will always be to one of the pages existing in memory
(d) will always lead to a page fault **[1997 : 1 M]**

4.8 Thrashing
(a) Reduces page I/O
(b) Decreases the degree of multipro-gramming
(c) Implies excessive page I/O
(d) Improve the system performance **[1997 : 1 M]**

4.9 Dirty bit for a page in a page table
(a) Helps avoid unnecessary writes on paging device
(b) Helps maintain LRU information
(c) Allows only read on a page
(d) None of the above **[1997 : 1 M]**

4.10 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 **[1998 : 1 M]**

4.11 The overlay tree for a program is as shown below:



What will be the size of the partition (in physical memory) required to load (and run) this program?
(a) 12 KB (b) 14 KB
(c) 10 KB (d) 8 KB
[1998 : 2 M]

4.12 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 instruction is
(a) $i + \frac{j}{k}$ (b) i + j * k
(c) $\frac{i+j}{k}$ (d) (i + j) * k
[1998 : 2 M]

4.13 Which of the following is/are advantage 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. **[1999 : 2 M]**

4.14 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 **[2000 : 2 M]**

4.15 Consider a machine with 64 MB physical memory and a 32-bit virtual address space. If the page size of 4 KB, what is the approximate size of the page table?
(a) 16 MB (b) 8 MB
(c) 2 MB (d) 24 MB
[2001 : 2 M]

4.16 Which of the following statements is/are false?
(a) Virtual memory implements the translation of a program's address space in to physical memory address space.
(b) Virtual memory allows each program to exceed the size of the primary memory.
(c) Virtual memory increases the degree of multiprogramming.
(d) Virtual memory reduces the context switching overhead. **[2001 : 1 M]**

4.17 The process of assigning load addresses to the various parts of the program and adjusting the code and data in the program to reflect the assigned addresses is called
(a) assembly (b) parsing
(c) relocation (d) symbol resolution
[2001 : 1 M]

4.18 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
[2001 : 1 M]

4.19 Which of the following is not a form of memory?
(a) Instruction cache
(b) Instruction register
(c) Instruction opcode
(d) Translation look a side buffer
[2002 : 1 M]

4.20 The optimal page replacement algorithm will select the page that
(a) Has not been used for the longest time in the past
(b) Will not be used for the longest time in the future
(c) Has been used least number of times
(d) Has been used most number of times
[2002 : 1 M]

4.21 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 **[2002 : 2 M]**

4.22 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 **[2003 : 1 M]**

Common Data for Q. 4.23 & 4.24:
A processor uses 2-level page tables for virtual to physical address translation. Page tables for both levels are stored in the main memory. Virtual and physical addresses are both 32 bits wide. The memory is byte addressable. For virtual to physical address translation, the 10 most significant bits of the virtual address are used as index into the first level page table while the next 10 bits are used as index into the second level page table. The 12 least significant bits of the virtual address are used as offset within the page. Assume that the page table entries in both levels of page tables are 4 bytes wide. Further, the processor has a translation look-aside buffer (TLB), with a hit rate of 96%. The TLB caches recently used virtual page numbers and the corresponding physical page numbers. The processor also has a physically addressed cache with a hit ratio of 90%. Main memory access time is 10 ns, cache access time is 1 ns, and TLB access time is also 1 ns.

4.23 Assuming that no page faults occur, the average time taken to access a virtual address is approximately (to the nearest 0.5 ns)
(a) 1.5 ns (b) 2 ns
(c) 3 ns (d) 4 ns
[2003 : 2 M]

4.24 Suppose a process has only the following pages in its virtual address space: two contiguous code pages starting at virtual address 0 × 00000000, two contiguous data pages starting at virtual address 0 × 00400000, and a stack page starting at virtual address 0 × FFFFF000. The amount of memory required for storing the page tables of this process is
(a) 8 KB (b) 12 KB
(c) 16 KB (d) 20 KB
[2003 : 2 M]

4.25 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 given 50% better benchmark results, what is the expected improvement in the I/O performance of user programs?
(a) 50% (b) 40%
(c) 25% (d) 0% **[2004 : 1 M]**

4.26 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
[2004 : 1 M]

4.27 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) 1260 nanoseconds
[2004 : 2 M]

4.28 Consider a fully associative cache with 8 cache blocks (numbered 0-7) and the following sequence of memory block requests: 4, 3, 25, 8, 19, 6, 25, 8, 16, 35, 45, 22, 8, 3, 16, 25, 7.
If LRU replacement policy is used, which cache block will have memory block 7?
(a) 4 (b) 5
(c) 6 (d) 7 **[2004 : 2 M]**

4.29 In a virtual memory system, size of virtual address is 32-bit, size of physical address is 30-bit, page size is 4 Kbyte and size of each page table entry is 32-bit. The main memory is byte addressable. Which one of the following is the maximum number of bits that can be used for storing protection and other information in each page table entry?
(a) 2 (b) 10
(c) 12 (d) 14 **[2004 : 2 M]**

4.30 For each of the four processes P_1, P_2, P_3 and P_4. The total size in kilobytes (KB) and the number of segments are given below.

Process	Total size (in KB)	Number of segments
P_1	195	4
P_2	254	5
P_3	45	3
P_4	364	8

The page size is 1 KB. The size of an entry in the page table is 4 bytes. The size of an entry in the segment table is 8 bytes. The maximum size of a segment is 256 KB. The paging method for memory management uses two-level paging, and

its storage overhead is P. The storage overhead for the segmentation method is S. The storage overhead for the segmentation and paging method is T. What is the relation among the overheads for the different methods of memory management in the concurrent execution of the above four processes?

(a) P < S < T (b) S < P < T
(c) S < T < P (d) T < S < P **[2006 : 2 M]**

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

(a) 11 bits (b) 13 bits
(c) 15 bits (d) 20 bits **[2006 : 2 M]**

4.32 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 **[2006 : 2 M]**

4.33 Let a memory have four free blocks of sizes 4k, 8k, 20k, 2k. These blocks are allocated following the best-fit strategy. The allocation requests are stored in a queue as shown below:

Request No.	Request sizes	Usage Time
J1	2k	4
J2	14k	10
J3	3k	2
J4	6k	8
J5	6k	4
J6	10k	1
J7	7k	8
J8	20k	6

The time at which the request for J7 will be completed will be

(a) 16 (b) 19
(c) 20 (d) 37 **[2007 : 1 M]**

4.34 The address sequence generated by tracing a particular program executing in a pure demand paging system with 100 bytes per page is 0100, 0200, 0430, 0499, 0510, 0530, 0560, 0120, 0220, 0240, 0260, 0320, 0410.
Suppose that the memory can store only one page and if x is the address which causes a page fault then the bytes from addresses x to x + 99 are loaded on to the memory. How many page faults will occur?

(a) 0 (b) 4
(c) 7 (d) 8 **[2007 : 1 M]**

4.35 A demand paging system takes 100 time units to service a page fault and 300 time units to replace a dirty page. Memory access time is 1 time unit. The probability of a page fault is p. In case of a page fault, the probability of page being dirty is also p. It is observed that the average access time is 3 time units. Then the value of p is

(a) 0.194 (b) 0.233
(c) 0.514 (d) 0.981 **[2007 : 2 M]**

Common Data for Q. 4.36 & Q. 4.37:

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

4.36 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 **[2007 : 2 M]**

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

(a) 0 (b) 1
(c) 2 (d) 3 **[2007 : 2 M]**

4.38 A paging scheme uses a Translation Look-aside Buffer (TLB). A TLB-access takes 10 ns and a main memory access takes 50 ns. What is the effective access time(in ns) if the TLB hit ratio is 90% and there is no page-fault?

(a) 54 (b) 60
(c) 65 (d) 75 **[2008 : 1 M]**

4.39 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
- bits 0-11 are used as offset within the page

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

(a) 20, 20 and 20 (b) 24, 24, and 24
(c) 24, 24 and 20 (d) 25, 25 and 24 **[2008 : 2 M]**

4.40 Assume that a main memory with only 4 pages, each of 16 bytes, is initially empty. The CPU generates the following sequence of virtual addresses and uses the Least Recently Used (LRU) page replacement policy.

0, 4, 8, 20, 24, 36, 44, 12, 68, 72, 80, 84, 28, 32, 88, 92

How many page faults does this sequence cause? What are the page numbers of the pages present in the main memory at the end of the sequence?

(a) 6 and 1, 2, 3, 4 (b) 7 and 1, 2, 4, 5
(c) 8 and 1, 2, 4, 5 (d) 9 and 1, 2, 3, 5 **[2008 : 2 M]**

4.41 Match the following flag bits used in the context of virtual memory management on the **List-I (Name of the bit)** with the different purposes on the **List-II (Purpose)** of the table below:

List-I	List-II
I. Dirty	a. Page initialization
II. R/W	b. Write-back policy
III. Reference	c. Page protection
IV. Valid	d. Page replacement policy

Codes:

(a) I-d, II-a, III-b, IV-c (b) I-b, II-c, III-a, IV-d
(c) I-c, II-d, III-a, IV-b (d) I-b, II-c, III-d, IV-a **[2008 : 2 M]**

4.42 In which one of the following page replacement policies, Bleady's anomaly may occur?

(a) FIFO (b) Optimal
(c) LRU (d) MRU **[2009 : 1 M]**

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

(a) virtual page number
(b) page frame number
(c) both virtual page number and page frame number
(d) access right information **[2009 : 1 M]**

4.44 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 **[2010 : 1 M]**

4.45 Let the page fault service time be 10 ms in a computer with average memory access time being 20 ns. If one page fault is generated for every 10^6 memory accesses, what is the effective access time for the memory?

(a) 21 ns (b) 30 ns
(c) 23 ns (d) 35 ns **[2011 : 1 M]**

4.46 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 computer system that has main memory size of 3 page frames which are initially empty. Let LRU, FIFO and OPTIMAL denote the number of page faults under the corresponding page replacement policy. Then

(a) OPTIMAL < LRU < FIFO
(b) OPTIMAL < FIFO < LRU
(c) OPTIMAL = LRU
(d) OPTIMAL = FIFO **[2012 : 2 M]**

Linked Answer Questions 4.47 and 4.48:

A computer uses 46-bit virtual address, 32-bit physical address, and a three-level paged page table organization. The page table base register stores the base address of the first-level (T_1), which occupies exactly one page. Each entry of T_1 stores the base address of a page of the second-level table (T_2). Each entry of T_2 stores the base address of a page of the third-level table (T_3). Each entry of T_3 stores a page table entry (PTE). The PTE is 32 it in size. The processor used in the computer has a 1 MB 16-way-set associative virtually indexed physical tagged cache the cache block size is 64 bytes.

4.47 What is the size of a page in KB in this computer?

(a) 2 (b) 4
(c) 8 (d) 16 **[2013 : 2 M]**

4.48 What is the minimum number of page colours needed to guarantee that no two synonyms map to different sets in the processor cache of this computer?
(a) 2 (b) 4
(c) 8 (d) 16 **[2013 : 2 M]**

4.49 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 _____. **[2014 (Set-1) : 2 M]**

4.50 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
[2014 (Set-2) : 2 M]

4.51 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
[2014 (Set-3) : 1 M]

4.52 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 _____. **[2014 (Set-3) : 2 M]**

4.53 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 _____. **[2015 (Set-1) : 1 M]**

4.54 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
[2015 (Set-1) : 2 M]

4.55 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 _____.
[2015 (Set-2) : 1 M]

4.56 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 **[2015 (Set-2) : 2 M]**

4.57 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.
[2015 (Set-2) : 2 M]

4.58 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. **[2016 (Set-1) : 2 M]**

4.59 Consider a computer system with ten physical page frames. The system is provided with an access sequence $(a_1, a_2, \ldots, a_{20}, a_1, a_2, \ldots, a_{20})$, where each a_i is a distinct virtual page 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 _____. **[2016 (Set-1) : 2 M]**

4.60 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) **[2016 (Set-2) : 1 M]**

4.61 Recall that Belady's anomaly is that the page-fault rate may increase as the number of allocated frames increases. Now, consider the following statements:
S1: Random page replacement algorithm (where a page chosen at random is replaced) suffers from Belady's anomaly
S2: LRU page replacement algorithm suffers from Belady's anomaly
Which of the following is CORRECT?
(a) S1 is true, S2 is true
(b) S1 is true, S2 is false
(c) S1 is false, S2 is true
(d) S1 is false, S2 is false **[2017 (Set-1) : 2 M]**

4.62 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 experimentally 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) $\frac{(D-M)}{(X-M)}$ (b) $\frac{(X-M)}{(D-M)}$
(c) $\frac{(D-X)}{(D-M)}$ (d) $\frac{(X-M)}{(D-X)}$
[2018 : 1 M]

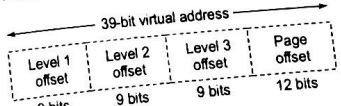
4.63 Assume that in a certain computer, the virtual addresses are 64 bits long and the physical addresses are 48 bits long. The memory is word addressable. The page size is 8 kB and the word size is 4 bytes. The Translation Look-aside Buffer (TLB) in the address translation path has 128 valid entries. At most how many distinct virtual addresses can be translated without any TLB miss?
(a) 16×2^{10} (b) 8×2^{20}
(c) 4×2^{20} (d) 256×2^{10} **[2019 : 2 M]**

4.64 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 statements is TRUE?
(a) The hole created by next fit is never larger than the hole created by best fit.
(b) The hole created by first fit is always larger than the hole created by next fit.
(c) The hole created by best fit is never larger than the hole created by first fit.
(d) The hole created by worst fit is always larger than the hole created by first fit.
[2020 : 1 M]

4.65 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 in from disk. TLB update time is negligible. The average memory access time in ns (round off to 1 decimal places) is _____. **[2020 : 2 M]**

4.66 In the context of operating systems, which of the following statements is/are correct with respect to paging?
(a) Paging incurs memory overheads.
(b) Multi-level paging is necessary to support pages of different sizes.
(c) Page size has no impact on internal fragmentation.
(d) Paging helps solve the issue of external fragmentation. **[2021 (Set-1) : 1 M]**

4.67 Consider a three-level page table to translate a 39-bit virtual address to a physical address as shown below:



The page size is 4 KB (1 KB = 2^{10} bytes) and page table entry size at every level is 8 bytes. A process P is currently using 2 GB (1 GB = 2^{30} bytes) virtual memory which is mapped to 2 GB of

physical memory. The minimum amount of memory required for the page table of P across all levels is _____ KB.

[2021 (Set-2) : 2 M]

4.68 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 V1 and V2 map to the same value while hashing, then the memory access time of these addresses will not be the same.

[2022 : 2 M]

4.69 Consider a demand paging system with four page frames (initially empty) and 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 _____.

[2022 : 2 M]

4.70 Consider the following two-dimensional array D in the C programming language, which is stored in row-major order:

int D[128][128];

Demand paging is used for allocating memory and each physical page frame holds 512 elements of the array D. The Least Recently Used (LRU) page-replacement policy is used by the operating system. A total of 30 physical page frames are allocated to a process which executes the following code snippet:

```
for (int i = 0; i < 128; i++)
    for (int j = 0; j < 128; j++)
        D[j][i] *= 10;
```

The number of page faults generated during the execution of this code snippet is _____.

[2023 : 2 M]

4.71 Consider a computer system with 57-bit virtual addressing using multi-level tree-structured page tables with L levels for virtual to physical address translation. The page size is 4 KB (1 KB = 1024 bytes) and a page table entry at any of the levels occupies 8 bytes.

The value of L is _____.

[2023 : 2 M]

4.72 Consider a memory management system that uses a page size of 2 KB. Assume that both the physical and virtual addresses start from 0. Assume that the pages 0, 1, 2, and 3 are stored in the page frames 1, 3, 2, and 0, respectively. The physical address (in decimal format) corresponding to the virtual address 2500 (in decimal format) is _____.

[2024 (Set-1) : 2 M]

4.73 Which of the following tasks is/are the responsibility/responsibilities of the memory management unit (MMU) in a system with paging-based memory management?

(a) Allocate a new page table for a newly created process.

(b) Translate a virtual address to a physical address using the page table.

(c) Raise a trap when a process tries to write to a page marked with read-only permission in the page table.

(d) Raise a trap when a virtual address is not found in the page table.

[2024 (Set-2) : 1 M]

4.74 Consider a 32-bit system with 4 KB page size and page table entries of size 4 bytes each. Assume 1 KB = 2^{10} bytes. The OS uses a 2-level page table for memory management, with the page table containing an outer page directory and an inner page table. The OS allocates a page for the outer page directory upon process creation. The OS uses demand paging when allocating memory for the inner page table, i.e., a page of the inner page table is allocated only if it contains at least one valid page table entry.

An active process in this system accesses 2000 unique pages during its execution, and none of the pages are swapped out to disk. After it completes the page accesses, let X denote the minimum and Y denote the maximum number of pages across the two levels of the page table of the process.

The value of $X + Y$ is _____.

[2024 (Set-2) : 2 M]

Answers Memory Management and Virtual Memory

4.1	(a)	4.2	(c)	4.3	(c)	4.4	(c)	4.5	(d)	4.6	(b)	4.7	(b)	4.8	(c)	4.9	(a)
4.10	(c)	4.11	(b)	4.12	(a)	4.13	(d)	4.14	(d)	4.15	(c)	4.16	(d)	4.17	(c)	4.18	(c)
4.19	(c)	4.20	(b)	4.21	(b)	4.22	(c)	4.23	(d)	4.24	(c)	4.25	(d)	4.26	(a)	4.27	(d)
4.28	(b)	4.29	(d)	4.30	(b)	4.31	(c)	4.32	(a)	4.33	(b)	4.34	(c)	4.35	(a)	4.36	(a)
4.37	(c)	4.38	(c)	4.39	(b)	4.40	(b)	4.41	(d)	4.42	(a)	4.43	(b)	4.44	(a)	4.45	(b)
4.46	(b)	4.47	(c)	4.48	(c)	4.49	(7)	4.50	(d)	4.51	(6)	4.52	(122)	4.53	(4)	4.54	(a)
4.55	(22)	4.56	(a)	4.57	(36)	4.58	(384)	4.59	(1)	4.60	(d)	4.61	(b)	4.62	(b)	4.63	(d)
4.64	(c)	4.65	(154.5)	4.66	(a, d)	4.67	(4108)	4.68	(c)	4.69	(0.6)	4.70	(4096)	4.71	(5)	4.72	(6596)
4.73	(b, c, d)	4.74	(1028)														

Explanations Memory Management and Virtual Memory

4.1 **(a)**

Segment table is often too large to fit in one page. So segment table itself must have a page table in a paged segmented scheme.

4.2 **(c)**

$$\text{Size of memory} = 4\text{K} \times 16 \text{ bits} = 2^{12} \times 16 \text{ bits}$$

$\Rightarrow$ 12 address lines and 16 data lines are needed.

4.3 **(c)**

Given that only one frame is available which can hold 100 records at one time (0-99).

100	200	430	499	510	530
×	×	×	✓	✓	×
560	120	220	240	260	320
✓	×	×	✓	✓	×
370					
✓					

4.4 **(c)**

In virtual memory system, Address Space specified by the address lines of CPU must be larger than the physical memory size and smaller than the secondary storage size.

4.5 **(d)**

n-bit multiplication storage required $= 2^{(n+n)} \times (n + n)$

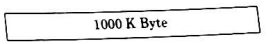
For 8-bit data $= 2^{(8+8)} \times (8 + 8)$ bit

$= 2^{16} \times 16$ bit

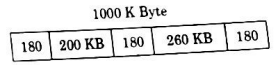
$= 64\text{ K} \times 16$ bit

4.6 **(b)**

Initially memory available:



After allocation 200 K byte and 260 K byte in worst case.



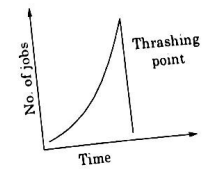
So, in worst case partition left with size 180 KB that are not contiguous. So, request for allocation request 181 will be denied because it is asking smallest request.

4.7 **(b)**

Locality of reference is a phenomenon describing the same value, related storage location being frequently accessed.

4.8 **(c)**

Thrashing: After reaching the maximum number of jobs allowed in main memory. CPU efficiency drastically getting down this situation is called Trashing.



Thrashing implies excessive page I/O.

4.9 (a)

Which page has Dirty bit only those pages being written back in main memory, so it avoids unnecessary writing.

4.10 (c)

Loader is used to load the program into main memory.

4.11 (b)

Maximum space that can be needed at a time is 14 kB (using E). The maximum space to load the program is decided by the maximum size taken by rootine call sequence from the root. i.e., Root → A → E.

4.12 (a)

EMAT = p × s + (1 − p) × m

p: page fault rate

s: page fault service time

m: main memory access time

$$\Rightarrow \frac{(i+j)}{k}+\left(1-\frac{1}{k}\right)\times i = i+\frac{j}{k}$$

4.13 (d)

Virtual memory allows the user can run the programs larger than the physical memory size.

4.14 (d)

Average memory access time = [(% of page miss) * (time of service a page fault) + (% of page hit) * (memory access time)]/100

So, average memory access time in microseconds is

$$\frac{(99.99*1+0.01*10*1000)}{100}$$

$$= \frac{(99.99+100)}{1000}$$

$$= \frac{199.99}{1000} = 1.9999\ \mu\text{sec}$$

4.15 (c)

Number of entries in page table = $\frac{2^{32}}{2^{12}} = 2^{20}$

Frame size = $\frac{2^{26}}{2^{12}} = 2^{14}$

∴ PT have to be stored in one frame so entry size must be 2 bytes, hence size of PT

$= 2^{20} \times 2 = 2^{21} = 2$ MB

4.16 (d)

In a system with virtual memory context switch includes extra overhead in switching of address spaces.

4.17 (c)

Relocation can change the assigned address of data and code in the program.

4.18 (c)

Bleady's Anamoly increasing number of frames = increase in page faults.

4.19 (c)

Instruction cache, instruction register and TLB are memories, but instruction opcode is the part of instruction that specify the operation to perform.

4.20 (b)

Optimal page replacement algorithm select a page that will not be used in the longest time in future.

4.21 (b)

In dynamic linking path for searching dynamic libraries is not known till runtime i.e. keep on changing, which can create security concern.

4.22 (c)

Given, Page Size = 1 KB

$= 2^{10}$ bytes

Virtual address is 32 bit long

So required number of pages

$$= \frac{2^{32}}{2^{10}} = 2^{22}$$

2^{22} is a very large so the large memory overhead in maintaining page tables.

4.23 (d)

Average time taken to access a virtual address

$$= \left[\left(\frac{96}{100}\right)(1+0.9\times 1+0.1\times(1+10))\right]$$

$$+\left[(0.04)(21+0.9\times 1+0.1\times(1+10))\right]$$

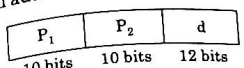
$$= [(.96)(1+0.9+0.1\times 11)]$$

$$+ [(0.04)(21+0.9+0.1\times 11)]$$

= 3.87 ns = 4 ns

4.24 (c)

Logical address = 32 bits

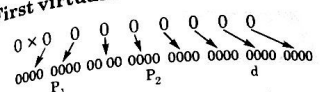


P_1= 1st level page table

P_2= 2nd level page table

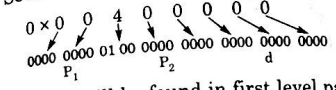
d = Page offset

First virtual address given



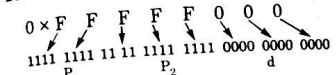
Address will be found in first level page table first block.

Second virtual address given

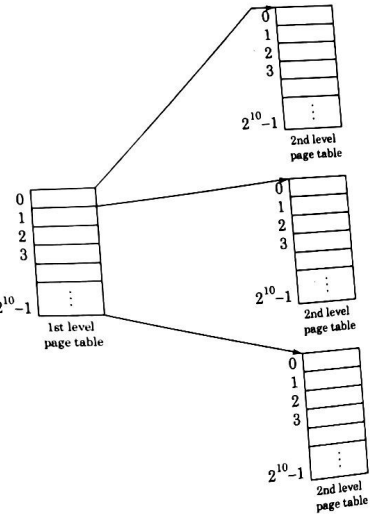


Address will be found in first level page table second block.

Third virtual address given



Address will be found in first level page table last block.



To execute the process we need to bring all the above 4 page tables in main memory.

$4 \times 2^{10} \times 4$ B = 16 kB

4.25 (d)

The I/O performance of a user program determined by many input and output devices not only by the disk. When we uses FCFS and replace it by SSTF it improves only disk driver performance not the entire I/O performance. So I/O improvement performance of user program is 0%.

4.26 (a)

Minimum number of page frames allocated to a running process is determined by the instruction set architecture.

4.27 (d)

Effective average instruction execution time = CPU time (100 ns) + Average time lost due to page fault per instruction $\left(=\frac{8\text{ ms}}{10000}\right)$ + Number of memory references × EMAT

100 + 800 + 2 × [0.9 (0 + 150) + 0.1 (0 + 3 × 150)]

= 1260

4.28 (b)

Fully associate cache with 8 block i.e. 1 set with 8 cache blocks.

Set	4	3	25	8	19	6	25	16
	B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7

Since set contain 8 blocks so, request for 45 replacement is needed. When 45 comes then

45	3	25	8	19	6	25	16
B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7

For request 22:

45	22	25	8	19	6	25	16
B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7

For request 8. Since already present, so no replacement.

For request 3:

45	22	25	8	3	6	25	16
B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7

For request 16; since already present, so no replacement.

For request 25; since already present, so no replacement.

For request 7:

45	22	25	8	3	7	25	16
B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7

So block 5 i.e. B_5 contain memory block 7.

4.29 **(d)**

In page table, number of entries is equal to number of pages in virtual memory and page table entry is equal to number of frames and other information i.e.

So, number of frames = $\frac{\text{Physical memory}}{\text{Page size}}$

$$= \frac{2^{30}}{2^{12}} = 2^{18}$$

So, number of bits for frames = $\log_2(2^{18}) = 18$

We know that,

Page table entry = Number of bits for frames + Other information

$$32 = 18 + x$$
$$x = 32 - 18 = 14$$

So, 14 bits can be used for storing protection and other information.

Note: Each page table entry must contain frame number.

4.30 **(b)**

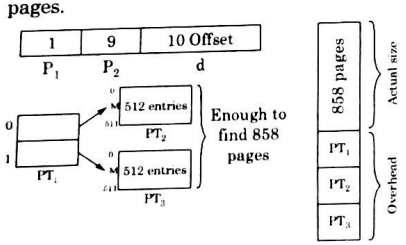
Overhead is considered as memory wasted by maintaining the page tables.

So, we need to find the overhead for every case in the given question.

Case-1: Overhead using two level paging.

Each page size is 1 KB.

Total pages = 195 + 254 + 45 + 364 = 858 KB pages.



Each page table entry = 4 bytes.

Size of page table$_1$ = $2 \times 4 = 8$ bytes

Size of page table$_2$ = $2^9 \times 4 = 2048$ bytes

Size of page table$_3$ = $2^9 \times 4 = 2048$ bytes

∴ Overhead of paging (P) = 8 + 2048 + 2048
= 4104 bytes

Case-2: Overhead using segmentation.

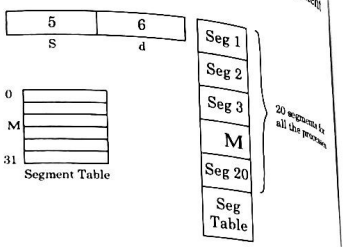
Number of segments = 4 + 5 + 3 + 8 = 20 (5 bits for segment number)

Maximum segment size

$$= \max\left(\frac{195}{4}, \frac{254}{5}, \frac{45}{3}, \frac{364}{8}\right)$$

= max(49,51,15,46) = 51 bytes (helps to find segment offset)

Number of bits required to find the limit (segment offset) = 6



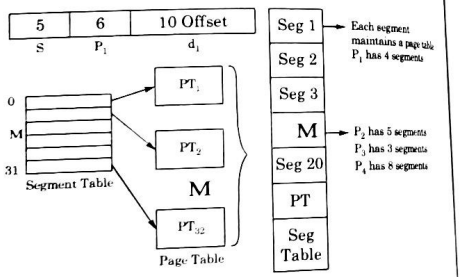
Size of segment table (aprox)
= 32 × 8 = 256 bytes

∴ Overhead using segmentation (S) = 256 bytes

Case-3: Overhead using segmentation and paging.

Number of segments = 20 ⇒ 5 bits required for segment number.

Maximum segment size = 51 bytes ⇒ 6 bits required for page offset.



20 page tables are required.

Each segment needed 1 page table.

Size of segment table = $32 \times 8 = 256$ bytes.

Size of each page table = $64 \times 4 = 256$ bytes.

∴ Segmentation and paging overhead (T)
= 256 B + 20 × 256 B = 5376 B

S: Simple segmentation takes less overhead.

P: Two level paging takes little more overhead because of two page tables.

T: Segmented paging takes more overhead because we have paging on both segment and segment table.

So, S < P < T.

4.31 **(c)**

Number of bits for virtual address = 32

Number of bits for pages = 4 KB
= $\log_2(2^{12}) = 12$

So, number of bits for pages = 32 – 12 = 20

Number of sets = $\frac{128}{4} = 32$

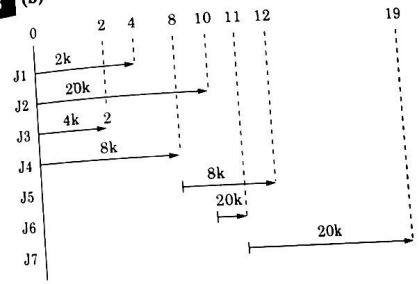
Number of bits for sets = $\log_2(32) = 5$

So, minimum tag size = 20 – 5 = 15

4.32 **(a)**

The main purpose of virtual memory is by allocating less memory to the process, we plan to increase degree of multiprogramming, by bringing more processes into memory. If virtual memory is removed then the efficient implementation of multiprogramming and multi user is no longer possible.

4.33 **(b)**



4.34 **(c)**

Size of page = 100

When page fault occur then from that page to next 99 byte are fetched into memory

Reference address	Address in memory	Hit/Fault
0100	0100 – 0199	Fault
0200	0200 – 0299	Fault
0430	0430 – 0529	Fault
0499	0430 – 0529	Hit
0510	0430 – 0529	Hit
0530	0530 – 0629	Fault
0560	0530 – 0629	Hit
0120	0120 – 0219	Hit
0220	0220 – 0319	Fault
0240	0220 – 0319	Hit
0260	0220 – 0319	Hit
0320	0320 – 0419	Fault
0410	0320 – 0419	Hit
		Total fault = 7

4.35 **(a)**

Average Access Time = $(1-p)(T_{mem}) + p[p(\text{S.T. dirty}) + (1-p)(\text{S.T. not dirty})]$

$3 = (1-p)(1) + p(p \times 300) + (1-p) \times 100)$

$3 = 1 - p + p^2\,300 + p\,100 - p^2\,100$

$3 = 1 + 99\,p + 200\,p^2$

$200\,p^2 + 99\,p - 2 = 0$

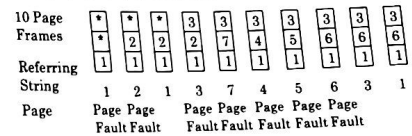
$$p = \frac{b \pm \sqrt{b^2 - 4ac}}{2a} = 0.0194$$

4.36 **(a)**

Number of page frames = 3

Consider the optimal page replacement algorithm as follows:

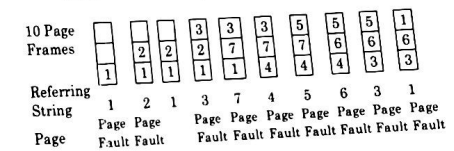
1, 2, 1, 3, 7, 4, 5, 6, 3, 1



So total number of page faults = 7

4.37 **(c)**

Consider the LRU page replacement policy



So total number of page fault = 9

Number of page fault in LRU – Number of page fault in optional = 9 – 7 = 2 more page fault.

4.38 **(c)**

Effective access time = hit ratio * (TLB access time + Memory access time) + miss ratio * (TLB access time + Page table access time + Memory access time)

= 0.9 * (10+50) + 0.1 *(10 + 50 + 50)

= 54 + 11 = 65

4.39 **(b)**

Page tables are stored in the main memory and main memory is divided into frames, it means addressing the page table is same as addressing the frames, so we just need to find, the number of bits required for addressing frames.

Number of bits required for first level = 24
Number of bits required for second level = 24
Number of bits required for third level = 24

4.40 (b)

Main memory is divided in page size = 16 bytes
We know that, page size = frame size
So in virtual address least 4 bit represent page size and remaining bits represents page number.
So, page number request are:

0 = 00 0000 i.e. Page number = 0

4 = 00 0100 i.e. Page number = 0

8 = 00 1000 i.e. Page number = 0

20 = 01 0100 i.e. Page number = 1

24 = 01 1000 i.e. Page number = 1

36 = 100 1000 i.e. Page number = 4

44 = 10 1100 i.e. Page number = 2

12 = 00 1100 i.e. Page number = 0

68 = 100 0100 i.e. Page number = 4

72 = 100 1000 i.e. Page number = 4

80 = 101 0000 i.e. Page number = 5

84 = 101 0100 i.e. Page number = 5

28 = 1 1100 i.e. Page number = 1

32 = 10 0000 i.e. Page number = 2

88 = 101 1000 i.e. Page number = 5

92 = 101 1100 i.e. Page number = 5

Request are:

0	0	0	1	1	4	2	0	4	4	5	5	1	2	5	5
						2	2	2	2	2	2	1	1	1	1
					4	4	4	4	4	4	4	4	4	4	4
			1	1	1	1	1	1	1	5	5	5	5	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2
F	H	H	F	H	F	F	H	H	H	F	H	F	F	H	H

So, at the end pages 1, 2, 4, 5 are in memory and 7 page fault are present.

4.41 (d)

- **Dirty bits** used in write back policy.
- **R/W bits** used in page protection.
- **Reference bit** used in page replacement policy (second chance policy).
- **Valid bit** used when page is initialized.

4.42 (a)

In Bleady's Anomaly if number of frames are increased then number of page faults increases. This behaviour found only with FIFO.

4.43 (b)

Page frame numbers are most important, virtual page number maynot be stored entirely.

4.44 (a)

FIFO policy for page replacement used.
Access 100 distinct pages by taking some example: 2 3 4 5 6 7 8 9
So by loading it get

5
4
3
2

= 4 page fault, | 9 |
|---|
| 8 |
| 7 |
| 6 |

= 4 page fault

and now access these page in reverse so

9 8 7 6 (So no page fault for these) 5 4 3 2 (page fault for these)

9
8
7
6

5
4
3
2

4 page fault

So total = 4 + 4 + 4 = 12 page fault
For 8 pages = 2 × 8 − 4 = 12
So for n pages = 2n − 4
So for 100 pages = 2 × (100) − 4 = 196

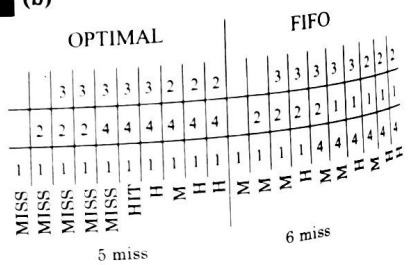
4.45 (b)

Effective access time = [(1 − p)* access time when no page fault + p* access time during page fault]

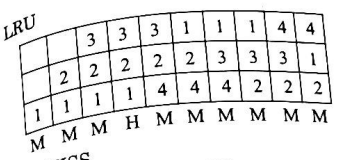
$$= \left[1-\left(\frac{1}{10^6}\right)\right]*20\,\text{ns}+\left(\frac{1}{10^6}\right)*10\,\text{ms}$$

$$= \left(\frac{10^6-1}{10^6}\right)*20\,\text{ns}+\left(\frac{1}{10^6}\right)*10*10^6\,\text{ns} = 30\,\text{ns}$$

4.46 (b)



LRU



⇒ 9 MISS
So optimal < FIFO < LRU

4.47 (c)

Assume Page Size = x bytes
We know that,
1st Page table size = Number of entries in page table × Page table entry size

$$= \frac{2^{46}}{x}\times 4B = \frac{2^{48}}{x}$$

2nd Page table size = Number of entries in 1st page table × Page table entry size

$$= \frac{2^{48}}{x^2}\times 4B = 2^{50}$$

3rd Page table size = Number of entries in 2nd page table × Page table entry size

$$= \frac{2^{50}}{x^3}\times 4B = \frac{2^{52}}{x^3}$$

Since 3rd level page table must fit into single page.

So, $x = \frac{2^{52}}{x^3}$

$x^4 = 2^{52}$

Take log both side

$4(\log_2 x) = 52\,(\log_2 2)$

$(\log_2 x) = 13$

$x = 2^{13} = 8$ KB

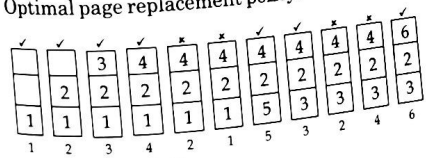
4.48 (c)

Since each cache line can contain 16,648 blocks, assuming best case 1024 bytes can accomodate in one line hence $\frac{2^{13}}{2^{10}} = 2^3$

⇒ Number of colours needed = 8.

4.49 (7)

Optimal page replacement policy:

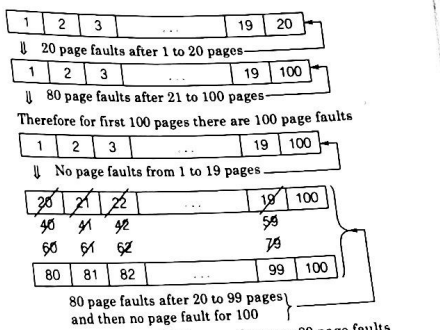


4.50 (d)

Pages:

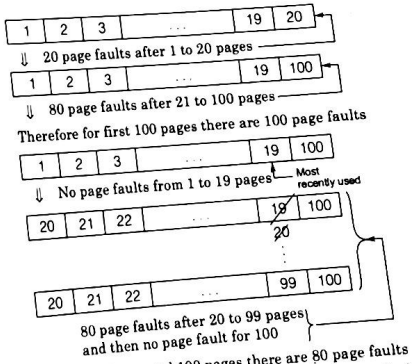
1, 2,...20,...100, 1,...20,...100, 1,...20,...100,

Optimal page replacement:



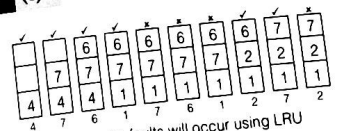
Therefore for second 100 pages there are 80 page faults
Similarly, for third 100 pages there will be 80 page faults.
∴ Total page faults for 300 pages = 100 + 80 + 80 = 260

Most recently used:



Therefore for second 100 pages there are 80 page faults
Similarly, for third 100 pages there will be 80 page faults.
∴ Total page faults for 300 pages = 100 + 80 + 80 = 260

4.51 (6)



4.52 **(122)**

Effective Memory Access Time = H_{TLB} (T_{TLB} + T_M) + (1 – H_{TLB}) (T_{TLB} + T_M + T_M)
= 0.6 (10 + 80) + (1 – 0.6) (10 + 80 + 80)
= 0.6 (90) + 0.4 (170) = 122

Note: In case of TLB miss 1st T_M is page table access time while 2nd T_M is main memory access time.

4.53 **(4)**

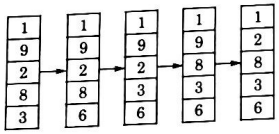
Page table size = Number of page table entries × Entry size
= Number of pages × 4 bytes

$$= \frac{2^{32}}{2^{12}} \times 4 \text{ bytes} = 4 \text{ MB}$$

4.54 **(a)**

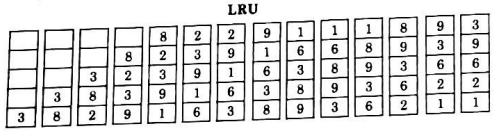
3, 8, 2, 3, 9, 1, 6, 3, 8, 9, 3, 6, 2, 1, 3

FIFO



FIFO 9 page faults.

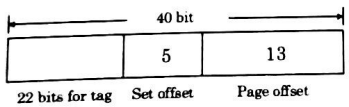
LRU



Here LRU also incurs 9 page faults.
Both FCFS and LRU incurs the same number of page faults.

4.55 **(22)**

Page size = 8 kB = 2^{13} bytes
⇒ 13 bits required for offset.
Number of sets in cache = 32
⇒ 5 bits required for index (set offset).



∴ 22 tag bits are required.

4.56 **(a)**

The best fit algorithm finds the smallest sufficient partition.

	357	491	468		210
200 KB	400 KB	600 KB	500 KB	300 KB	290 KB

∴ The memory partitions 200 KB and 300 KB are not allotted.

4.57 **(36)**

Page size = 8 KB ⇒ 13 bit offset
Number of frame bits = 32 – 13 = 19 bits
Page table entry = Valid + Dirty + Permission bits + Translation (frame bits)
= 1 + 1 + 3 + Frame bits
= 5 + 19 = 24 bits
Page table size = 24 Mbytes
Number of pages = Number of page table entries

$$= \frac{24 \text{ Mbytes}}{24 \text{ Bits}} = 8 \text{ M} = 2^{23} \text{ pages}$$

∴ 23 bits needed for page and 13 bits offset
Length of virtual address = 23 + 13 = 36 bits.

4.58 **(384)**

Page table size = Number of entries in page table × Page table entry size

$$= \left(\frac{2^{40}}{2^{14}}\right) \times 48 \text{ bits} = 2^{26} \times 6 \text{ bytes}$$

= 64 M × 6 B = 384 MB

4.59 **(1)**

This question is little tricks, but not difficult. You can solve the problem by taking smaller example with smaller values.

Example: 1, 2, 3, 4, 1, 2, 3, 4 with '2' frames.

LIFO: 1, 2, 3, 4, 1, 2, 3, 4

	2	3	4	4	2	3	4
1	1	1	1	1	1	1	1
F	F	F	F		F	F	F

Total page faults = 7

Optimal: 1, 2, 3, 4, 1, 2, 3, 4

	2	3	4	4	4	4	4
1	1	1	1	4	2	3	3
F	F	F	F		F	F	

Total page faults = 6
7 – 6 = 1

4.60 **(d)**

Because of Belady's anamoly, it will happen in FIFO phase replacement algorithm.

4.61 **(b)**

Considering each statement:
- Random page replacement algorithm can behave like any algorithm probably FCFS too, hence it can suffer from Belady's anomaly.
- LRU page replacement algorithm doesn't suffer from Belady's anomaly.

4.62 **(b)**

$$EMAT = P \times S + (1 - P) \times M$$
$$X = P \times D + (1 - P) \times M$$
$$X = P \times D + M - P \times M$$
$$X - M = P (D - M)$$
$$P = \frac{X - M}{D - M}$$

4.63 **(d)**

1 word = 4 bytes, Page size = 8 kB = 2^{13} B

Number of words in 1 page = $\frac{2^{13}}{2^2} = 2^{11}$

TLB can hold 128 valid entries so, at most 128 × 2^{11} memory address can be addressed without TLB miss.

$$128 \times 2^{11} = 256 \times 2^{10}$$

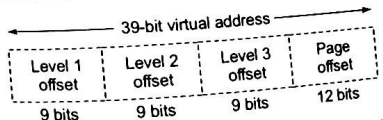
4.64 **(c)**

The hole created by best fit is never larger than the hole created by first fit. This is correct option.

4.65 **(154.5)**

EMAT = 0.95 × (20 + 100) + 0.05 × (0.9 × (20 + 100 + 100) + 0.1 × [0.2 × (20 + 100 + 5000 + 5000) + 0.8 × (20 + 100 + 5000)]
= 154.5 ns

4.67 **(4108)**



Since process P is using 2 GB virtual memory and page size 2^{12} bytes.

So, Number of pages = $\frac{2^{31}}{2^{12}} = 2^{19}$

There are 2^9 entries per page table in 3rd level.

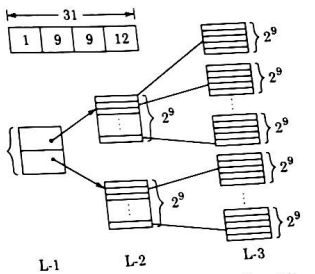
So, we need = $\frac{2^{19}}{2^9} = 2^{10}$ page tables in 3rd level

Now, it means 2^{10} entries in 2nd level so number of page tables in 2nd level

$$= \frac{2^{10}}{2^9} = 2 \text{ page table}$$

Now we have 2 entries in 1st level hence we need only 1 page table in 1st level.
Total number of page table = 2^{10} + 2 + 1
= 1027 per table
Each page table has 2^9 entries and each entry size is 8 bytes.
So, total size of page table in bytes
= 1027 × 2^9 × 8 bytes = 4108 KB

Alternative method:



Total page table = 1 + 2^1 + 2^1 × 2^9 = 1027
Page table size = 1027 × 2^{12} = 4108 KB

4.68 **(c)**

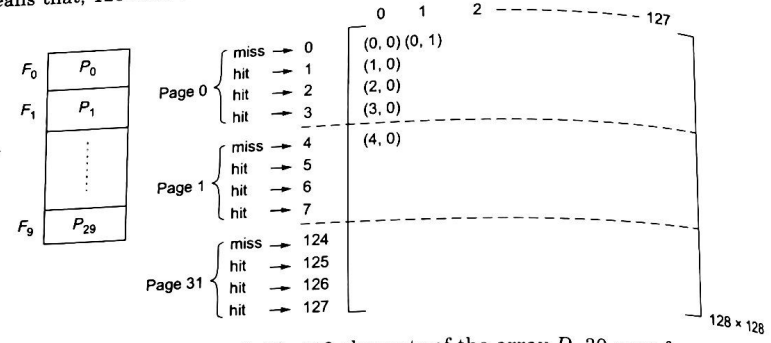
(a) TLB performs parallel search. Hence true
(b) If the TLB hit means, word will surely present in main memory. Hence true
(c) Memory access time using inverted page table is always not same, because there is no indexing applied, we follows no equal linear search. Hence false
(d) If two distinct virtual address map to same value while hashing, they will be resolved using linked list, so memory access time will not be same. Hence true

4.69 **(0.6)**

9 page faults out of total 15 references $\frac{9}{15} = 0.6$

4.70 **(4096)**

Given array dimension is $D[128][128]$ which is stored in row major order.
It means that, 128 rows and 128 columns.



Given that, each physical frame holds 512 elements of the array D. 30 page frames are allocated to a process. Array access is in column major order $[D[j][i]$ *= 10]
As each physical frame holds 512 elements, the 1st 4 rows of the array belongs one page, we can see that, one page fault 4 elements.
Array contains 32 pages, but memory contains 30 page frames.

$$\therefore \quad n_{out} \text{ page faults} = \frac{128 \times 128}{4} = \frac{2^{14}}{2^2} = 4096$$

4.71 **(5)**

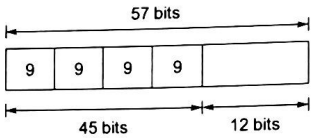
$$\text{Offset} = \log_2(\text{Page size})$$
$$= \log_2(4\ \text{KB}) = 12 \text{ bits}$$
$$\text{PTE} = 8 \text{ bytes}$$

For every level page table should fit into a page.

$$\text{So, number of entries in a page} = \frac{4\ \text{KB}}{8\ \text{B}} = 2^9$$

$$\text{Bits for a level} = \log_2(2^9) = 9 \text{ bits}$$

Following is representation



So, 5 levels required.

4.72 **(6596)**

$$\text{P.A.} \rightarrow (1100111000100)_2 \rightarrow (6596)_{10}$$

5 File System and Device Management

5.1 The correct matching for the following pairs is

List-I	List-II
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) A-4 B-3 C-1 D-2 (b) A-2 B-1 C-3 D-4
(c) A-4 B-3 C-2 D-1 (d) A-2 B-3 C-4 D-1
[1997 : 1 M]

5.2 The correct matching for the following pairs is

List-I	List-I
A. Disk scheduling	1. Round robin
B. Batch processing	2. SCAN
C. Time sharing	3. LIFO
D. Interrupt processing	4. FIFO

Codes:
(a) A-3, B-4, C-2, D-1
(b) A-4, B-3, C-2, D-1
(c) A-2, B-4, C-1, D-3
(d) A-3, B-4, C-3, D-2
[1997 : 1 M]

5.3 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 2 programs through a pipe
(d) None of the above **[1997 : 1 M]**

5.4 When an interrupt occurs, an operating system
(a) ignores the interrupt
(b) always changes state of interrupted process after processing the interrupt
(c) always resumes execution of interrupted process after processing the interrupt
(d) may change state of interrupted process to 'blocked' and schedule another process
[1997 : 1 M]

5.5 Which of the following devices should get higher priority in assigning interrupts?
(a) Hard disk (b) Printer
(c) Keyboard (d) Floppy disk
[1998 : 1 M]

5.6 Which of the following is true?
(a) Unless enabled, a CPU will not be able to process interrupts.
(b) Loop instructions cannot be interrupted till they complete.
(c) A processor checks for interrupts before executing a new instruction.
(d) Only level triggered interrupts are possible on microprocessors **[1998 : 1 M]**

5.7 Listed below are some operating system abstractions (in the left column) and the hardware components (in the right column)?

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 **[1999 : 1 M]**

5.8 Which of the following disk scheduling strategies is likely to give the best throughput?
(a) Farthest cylinder next
(b) Nearest cylinder next
(c) First come first served
(d) Elevator algorithm **[1999 : 1 M]**

5.9 Which of the following requires a device driver?
(a) Register (b) Cache
(c) Main memory (d) Disk **[2001 : 1 M]**

5.10 In the index allocation scheme of blocks to a file, the maximum possible size of the file depends on
(a) the size of the blocks, and the size of the address of the blocks.
(b) the number of blocks used for the index, and the size of the blocks
(c) the size of the blocks, the number of blocks used for the index, and the size of the address of the blocks
(d) None of the above **[2002 : 2 M]**

5.11 Using a larger block size in a fixed block size file system leads to
(a) better disk throughput but poorer disk space utilization

(b) better disk throughput and better disk space utilization
(c) poorer disk throughput but better disk space utilization
(d) poorer disk throughput and poorer disk space utilization **[2003 : 1 M]**

5.12 A Unix-style I-node has 10 direct pointers and one single, one double and one triple indirect pointers. Disk block size is 1 Kbyte, disk block address is 32 bits, and 48-bit integers are used. What is the maximum possible file size?
(a) 2^{24} bytes (b) 2^{32} bytes
(c) 2^{34} bytes (d) 2^{48} bytes **[2004 : 2 M]**

5.13 A disk has 200 tracks (numbered 0 through 199). At a given time, it was servicing the request of reading data from track 120, and at the previous request, service was for track 90. The pending requests (in order of their arrival) are for track numbers.

30 70 115 130 110 80 20 25

How many times will the head change its direction for the disk scheduling policies SSTF (Shortest Seek Time First) and FCFS (First Come Fist Serve)?
(a) 2 and 3 (b) 3 and 3
(c) 3 and 4 (d) 4 and 4 **[2004 : 2 M]**

5.14 In a particular Unix OS, each data block is of size 1024 bytes, each node has 10 direct data block addresses and three additional addresses: one for single indirect block, one for double indirect block and one for triple indirect block. Also, each block can contain addresses for 128 blocks. Which one of the following is approximately the maximum size of a file in the file system?
(a) 512 MB (b) 2 GB
(c) 8 GB (d) 16 GB **[2004 : 2 M]**

5.15 In a computer system, four files of size 11050 bytes, 4990 bytes, 5170 bytes and 12640 bytes need to be stored. For storing these files on disk, we can use either 100 byte disk blocks or 200 byte disk blocks (but can't mix block sizes). For each block used to store a file, 4 bytes of bookkeeping information also needs to be stored on the disk. Thus, the total space used to store a file is the sum of the space taken to store the file and the space taken to store the book keeping information for the blocks allocated for storing the file. A disk block can store either bookkeeping information for a file or data from a file, but not both.
What is the total space required for storing the files using 100 byte disk blocks and 200 byte disk blocks respectively?
(a) 35400 and 35800 bytes
(b) 35800 and 35400 bytes
(c) 35600 and 35400 bytes
(d) 35400 and 35600 bytes **[2005 : 2 M]**

5.16 In the working-set strategy, which of the following is done by the operating system to prevent thrashing?
I. It initiates another process if there are enough extra frames.
II. It selects a process to suspend if the sum of the sizes of the working-sets.
III. Exceeds the total number of available frames.
(a) I only (b) II only
(c) Neither I nor II (d) Both I and II **[2006 : 1 M]**

Directions for Question 5.17 to 5.18:
The head of a hard disk serves requests following the shortest seek time first (SSTF) policy. The head is initially positioned at track number 180.

5.17 Which of the request sets will cause the head to change its direction after servicing every request assuming that the head does not change direction if there is a tie in SSTF and all the requests arrive before the servicing starts?
(a) 11, 139, 170, 178, 181, 184, 201, 265
(b) 10, 138, 170, 178, 181, 185, 201, 265
(c) 10, 139, 169, 178, 181, 184, 201, 265
(d) 10, 138, 170, 178, 181, 185, 200, 265 **[2007 : 2 M]**

5.18 What is the maximum cardinality of the request set, so that the head changes its direction after servicing every request if the total number of tracks are 2048 and the head can start from any track?
(a) 9 (b) 10
(c) 11 (d) 12 **[2007 : 2 M]**

5.19 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) 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 **[2007 : 2 M]**

5.20 The data block of a very large file in the Unix file system are allocated using
(a) Contiguous allocation
(b) Linked allocation
(c) indexed allocation
(d) an extension of indexed allocation **[2008 : 1 M]**

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

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

Assuming that the head is currently at cylinder 50, what is the time taken to satisfy all requests if it takes 1 ms to move from one cylinder to adjacent one and shortest seek time first policy is used?
(a) 95 ms (b) 119 ms
(c) 233 ms (d) 276 ms **[2009 : 2 M]**

5.22 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 look a side buffer
(d) it helps to reduce the number of page faults in page replacement algorithms. **[2009 : 2 M]**

5.23 A file system with 300 GByte disk uses a file descriptor with 8 direct block addresses, I 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 KBytes
(d) dependent on the size of the disk **[2012 : 2 M]**

5.24 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. **[2014 (Set-1) : 1 M]**

5.25 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 100×10^6 bytes disk on which the file system is stored and data block size is 10^3 bytes, the maximum size of a file that can be stored on this disk in units of 10^6 bytes is _____. **[2014 (Set-2) : 1 M]**

5.26 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. **[2015 (Set-1) : 2 M]**

5.27 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 _____. **[2016 (Set-1) : 2 M]**

5.28 In a file allocation system, which of the following allocation scheme(s) can be used if no external fragmentation is allowed?
I. Contiguous II. Linked
III. Indexed
(a) I and III only (b) II only
(c) III only (d) II and III only **[2017 (Set-2) : 1 M]**

5.29 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 the 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 _____. **[2018 : 2 M]**

5.30 The index node (inode) of a Unix-like file system has 12 direct, one single-indirect and one double-indirect pointers The disk block size is 4 kB and the disk block address is 32-bits long. The maximum possible file size is (rounded off to 1 decimal place) _____ GB. **[2019 : 2 M]**

5.31 Consider the following 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) R is serviced before P.
(b) T is serviced before P.
(c) Q is serviced after S but before T.
(d) The head reverses its direction of movement between servicing of Q and P.

[2020 : 2 M]

5.32 Consider a linear list based directory implementation in a file system. Each directory is a list of nodes, where each node contains the file name along with the file metadata, such as the list of pointers to the data blocks. Consider a given directory foo.

Which of the following operations will necessarily require a full scan of foo for successful completion?

(a) Opening of an existing file in foo
(b) Renaming of an existing file in foo
(c) Creation of a new file in foo
(d) Deletion of an existing file from foo

[2021 (Set-1) : 1 M]

5.33 Consider two files systems A and B, that use contiguous allocation and linked allocation respectively. A file of size 100 blocks is already stored in A and also in B. Now, consider inserting a new block in the middle of the file (between 50th and 51st block), whose data is already available in the memory. Assume that there are enough free blocks at the end of the file and that the file control blocks are already in memory. Let the number of disk accesses required to insert a block in the middle of the file in A and B are n_A and n_B, respectively, then the value of $n_A + n_B$ is _____. **[2022 : 2 M]**

5.34 Consider a 512 GB hard disk with 32 storage surfaces. There are 4096 sectors per track and each sector holds 1024 bytes of data. The number of cylinders in the hard disk is _____. **[2024 (Set-1) : 2 M]**

5.35 Consider a disk with the following specifications: rotation speed of 6000 RPM, average seek time of 5 milliseconds, 500 sectors/track, 512 byte sectors. A file has content stored in 3000 sectors located randomly on the disk. Assuming average rotational latency, the total time (in seconds, rounded off to 2 decimal places) to read the entire file from the disk is _____. **[2024 (Set-2) : 2 M]**

Answers File System and Device Management

5.1	(b)	5.2	(c)	5.3	(b)	5.4	(d)	5.5	(a)	5.6	(a, c)	5.7	(c)	5.8	(b)	5.9	(d)
5.10	(c)	5.11	(a)	5.12	(c)	5.13	(c)	5.14	(b)	5.15	(c)	5.16	(b)	5.17	(b)	5.18	(c)
5.19	(b)	5.20	(d)	5.21	(b)	5.22	(b)	5.23	(b)	5.24	(3)	5.25	(99.6)	5.26	(10)	5.27	(346)
5.28	(d)	5.29	(85)	5.30	(4)	5.31	(c)	5.32	(b, c)	5.33	(153)	5.34	(4096)	5.35	(30.06)		

Explanations File System and Device Management

5.1 **(b)**

DMA I/O → Disk
Cache → High speed RAM
Interrupt I/O → Printer
Condition code register → ALU

5.2 **(c)**

Disk scheduling → SCAN
Batch processing → FIFO (Queue)
Time sharing → Round Robin (uses time slice)
Interrupt processing → LIFO (stack)

5.3 **(b)**

I/O redirection helps to use an existing file as input file for a program.

5.4 **(d)**

When an interrupt occurs, may change state of interrupted process to 'blocked' and schedule another process.

5.5 **(a)**

Hard disk should be higher priority compared to printer, keyboard, and floppy disk. Because all user programs and OS stored at hard disk.

5.6 **(a, c)**

(a) When interrupt is enabled, a CPU will be able to process the interrupt.
(b) CPU checks for interrupts before executing a new instruction.

5.7 **(c)**

Thread → CPU
Virtual address space → Memory
File system → Disk
Signal → Interrupt

5.8 **(b)**

Shortest seek time next or nearest cylinder next gives the best throughput.

5.9 **(d)**

Disk requires a device driver in computer system.

5.10 **(c)**

$\frac{\text{DB size}}{\text{DBA}}$ = Number of DBA's possible in one disk block.

5.11 **(a)**

In a fixed block size file system, larger block size gives better disk throughput but poorer disk space utilization.

5.12 **(c)**

The number of disk block pointers that will be fit in one block

$$= \frac{2^{10}\text{ byte}}{32\text{ bit}} = \frac{2^{10}\text{ byte}}{4\text{ byte}} = 256$$

Maximum file size due to single indirection pointer

$= 256 \times 1 = 256$ KB

Maximum file size due to direct pointer

$= 10 \times 1 = 10$ KB

Maximum file size due to double indirection pointer

$= 256 \times 256 \times 1$ KB

Maximum file size due to triple indirection pointer

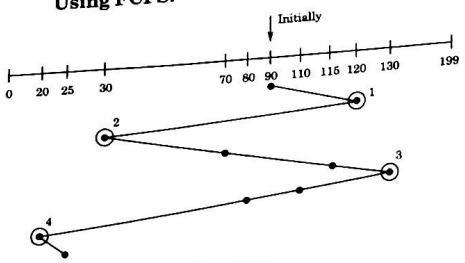
$= 256 \times 256 \times 256 \times 1$ KB

So the maximum file size

$= 256 \times 256 \times 256 \times 1$ KB
$= 2^8 \times 2^8 \times 2^8 \times 2^{10}$ bytes
$= 2^{24} \times 2^{10}$ bytes $= 2^{34}$ bytes

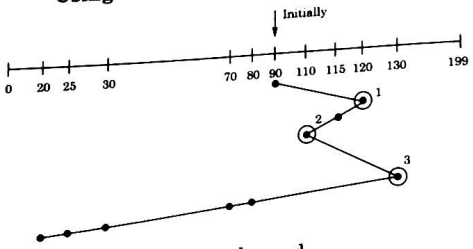
5.13 **(c)**

Using FCFS:



So, 4 head direction change.

Using SRTF:



So, 3 head direction changed.

5.14 **(b)**

Maximum file size = [direct DBA + Number of (Data block size/DBA) + Number of (Data block size/DBA)^2 + Number of (Data block size/DBA)^3]* Data block size.

Data Block size = 1024 byte

(Disk Block size/DBA) = Number of Disk Block address Store inside One Block.

Maximum file size = [10+1*(128) + 1*(128* 128*) + 1*(128*128*128)]*1024 Byte

= Approx 2 GB.

5.15 **(c)**

Using 100 bytes disk block:

100 B

File size	100 B block	Bytes needed for Book keeping	Block for Information
11050	111	111 × 4 = 444	5
4990	50	50 × 4 = 200	2
5170	52	52 × 4 = 208	3
12640	127	127 × 4 = 508	6
Total	**340**		**16**

Space = (340 + 16) × 100 = 35600

Using 200 bytes disk block:

200 B

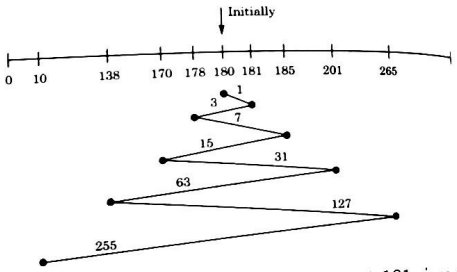
File size	200 B block	Bytes needed for Book keeping	Block for Information
11050	56	56 × 4 = 224	
4990	25	25 × 4 = 100	2
5170	26	26 × 4 = 104	1
12640	64	64 × 4 = 256	1
Total	**171**		2
			6

Space = (171 + 6) × 200 = 35400

5.16 **(b)**

If there are enough extra frames, another process can be initiated. If the sum of the working-set sizes increases, exceeding the total number of available frames, the operating system selects a process to suspend. The process's pages are written out (swapped), and its frames are reallocated to other processes. The suspended process can be restarted later."

5.17 **(b)**



Initially at 180, so need to go at 181 since all option contain it.

Now 181 – 184 = 3 and 181 – 178 = 3 and 181 - 185 = 4.

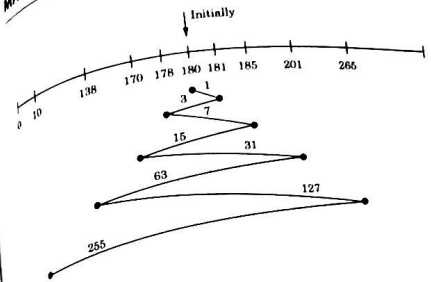
Since on tie direction is not changed. So request service next will be 185. So option (b) and (d) can be answer.

Now after services 185, 185 – 200 = 15 and 185 - 170 = 15 and 185 – 201 = 16. So we need to choose request 201 since with 200 direction will not change.

5.18 **(c)**

If we see the pattern of distance of request, then 1, 3, 7, 15, 31, 63, 127, 255, 511, 1023, 2047 by using, new min distance = (old distance × 2) + 1.

So, maximum 11 request can be satisfied.



5.19 **(b)**

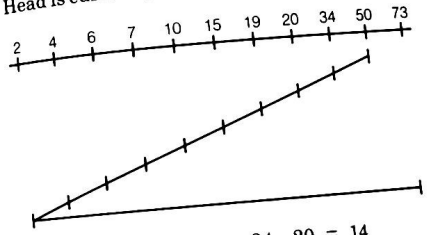
P and Q both are true but Q is not the reason for P because increasing the number of page frames allocated to a process sometimes increases the page fault rate or it is not concern with replacement policy.

5.20 **(d)**

An extension of index allocation is used to allocate the data block of a very large file.

5.21 **(b)**

Head is currently at cylinder 50,



$50-34 = 16$, $34-20 = 14$

$20-19 = 1$, $19-15 = 4$

$15-10 = 5$, $10-7 = 3$

$7-6 = 1$, $6-4 = 2$

$4-2 = 2$, $2-73 = 71$

Total moves = 16 + 14 + 1 + 4 + 5 + 3 + 1 + 2 + 2 + 71 = 119

5.22 **(b)**

Multilevel page table in preferred to reduce the size of page table needed to implement the virtual address space of a process.

5.23 **(b)**

Maximum possible size

$$= \left[\left(\begin{array}{l}\text{address pointed by}\\ \text{doubly indirect block}\end{array}\right)^2 + \left(\begin{array}{l}\text{addres pointed by}\\ \text{single address}\end{array}\right)\right.$$

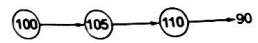
+ Address points by single direct address] × [Block size]

$$= \left[\left(\frac{128}{8}\right)^2 + \left(\frac{128}{8}\right) + 8\right] \times 128\ \text{B}$$

$= [2^8 + 2^4 + 2^3] \times 128\ \text{B}$

$= 32\ \text{kB} + 2\ \text{kB} + 1\ \text{kB} = 35\ \text{kB}$

5.24 **(3)**



90 is serviced after servicing the 3 requests.

5.25 **(99.6)**

Overhead of each entry in FAT = 4 bytes

Block size = 10^3 bytes

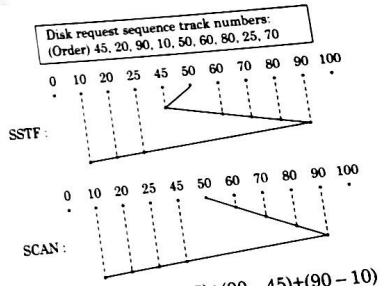
Total size for each entry = 1004 bytes

$$\text{Number of entries in FAT} = \frac{100 \times 10^6}{1004} = 0.099601$$

Maximum size of a file

$= 0.099601 \times 10^3$ bytes $= 99.601 \times 10^6$ bytes

5.26 **(10)**



SSTF distance = (50 – 45)+(90 – 45)+(90 – 10)

= 5 + 45 + 80 = 130

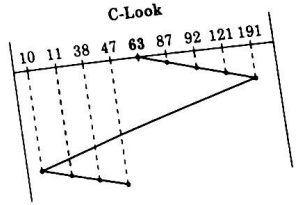
SCAN distance = (90 – 50) + (90 – 10)

= 40 + 80 = 120

∴ SSTF distance – SCAN distance

= 130 – 120 = 10

5.27 **(346)**

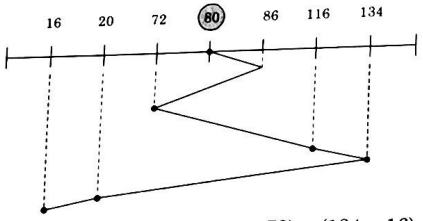


$(84 - 63) + (92 - 87) + (121 - 92) + (191 - 121)$
$+ (191 - 10) + (11 - 10) + (38 - 11) + (47 - 38)$
$= 24 + 5 + 29 + 70 + 181 + 1 + 27 + 9 = 346$

5.28 **(d)**

Linked and indexed allocations are non contigous so, they will not suffer from external fragmentation.

5.29 **(85)**



$(86 - 80) + (86 - 72) + (134 - 72) + (134 - 16)$
$+ 62 + 118 + 14 = 200$
$100 \rightarrow 20$
$200 \rightarrow ?$

$$\frac{200}{100} \times 20 = 40$$

3 direction changes $3 \times 15 = 45$
$40 + 45 = 85$

5.30 **(4)**

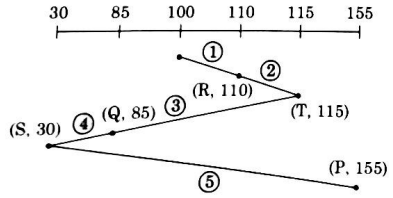
Size of disk block = 4 kB
Disk block address = 32 bit = 4 B

Number of address in one disk block $= \frac{4\text{ kB}}{4\text{ B}} = 2^{10}$

12 direct, 1 single indirect, 1 double indirect pointers.

Maximum possible file size
$= (12 + 2^{10} + 2^{10} \times 2^{10}) \times 4\text{ kB}$
$= 4.00395\text{ GB} \cong 4\text{ GB}$

5.31 **(c)**



- Option (a) is correct as P is serviced last.
- Option (b) is correct as P is serviced last.
- Option (c) is false as Q is serviced before S.
- Option (d) is correct.

5.33 **(153)**

Contiguous Allocation: 50 read operations + 50 write operations + 1 operation to write the newly inserted block = 101 operations (n_A)

Linked Allocation: 50 read operations + 1 operation delete next pointer of 50th element + 1 operation to connect it to the 51st element = 52 operations (n_B)

Total operations:

Required $= n_A + n_B$
$= 101 + 52 = 153$

5.34 **(4096)**

Disk capacity = Number of surfaces × Number of tracks per surface × Number sectors per track × Number of bytes per sector

$512\text{ GB} = 32 \times x \times 4096 \times 1024\text{ B}$
$2^{39}\text{ B} = 2^5 \times x \times 2^{12} \times 2^{10}\text{ B}$
$2^{39}\text{ B} = 2^{27} x\text{ B}$

$$x = \frac{2^{39}\text{ B}}{2^{27}\text{ B}} = 2^{12}$$

$x = 4096$

5.35 **(30.06)**

Average seek time = 5 ms

6000 revolutions — 1 min (60 sec)
1 revolution — ?

$\Rightarrow \left(\frac{60}{6000}\right)$ sec

$\Rightarrow$ 10 ms

Average Rotational latency $= \frac{1}{2} \times 10\text{ ms} = 5\text{ ms}$

1 revolution time — 500 sectors
? — 1 sector

$$= \frac{10\text{ ms}}{500} = 0.02\text{ ms}$$

T_{avg} = Average seek time + Average Rotational latency + Transfer time + Over head

$T_{avg} = (5\text{ ms} + 5\text{ ms} + 0.02\text{ ms} + 0)\ 3000$
$= 30060\text{ ms} = 30.06\text{ sec}$

■■■■

6 Miscellaneous

6.1 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
[1995 : 1 M]

6.2 When an interrupt occurs, an operating system
(a) ignores the interrupt
(b) always changes state of interrupted process after processing the interrupt
(c) always resumes execution of interrupted process after processing the interrupt
(d) may change state of interrupted process to 'blocked' and schedule another process
[1997 : 1 M]

6.3 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) kernel mode (b) super user mode
(c) privileged mode (d) user mode
[2005 : 1 M]

6.4 A student wishes to create symbolic links in a computer system running Unix. Three text files named "file1", "file2" and "file3" exist in her current working directory, and the student has read and write permissions for all three files. Assume that file1 contains information about her hobbies, file2 contains information about her friends and file3 contains information about her courses. The student executes the following sequence of commands from her current working directory

```
ln -s file1 file2
ln -s file2 file3
```

Which of the following types of information would be lost from her file system?
I. Hobbies
II. Friends
III. Courses
(a) I and II only (b) II and III only
(c) II only (d) I and III only
[2005 : 1 M]

6.5 The shell command

```
find -name password -print
```

is executed in /etc directory of a computer system running Unix. Which of the following shell commands will give the same information as the above command when executed in the same directory?
(a) ls passwd
(b) cat passwd
(c) grep name passwd
(d) grep print passwd
[2005 : 1 M]

6.6 Consider n jobs $J_1, J_2, \ldots, J_n$ such that job J_i has execution time t_i and a non-negative integer weight w_i. The weighted mean completion time of the jobs is defined to be $\frac{\sum_{i=1}^{n} w_i T_i}{\sum_{i=1}^{n} w_i}$ where T_i is the completion time of job J_i. Assuming that there is only one processor available, in what order must the jobs be executed in order to minimize the weighted mean completion time of the jobs?
(a) Non-decreasing order of t_i
(b) Non-increasing order of w_i
(c) Non-increasing order of $w_i t_i$
(d) Non-increasing order of w_i / t_i
[2007 : 2 M]

■■■■

Answers Miscellaneous

6.1 (c) 6.2 (d) 6.3 (a) 6.4 (b) 6.5 (a) 6.6 (d)

Explanations Miscellaneous

6.1 (c)

Linker definition: Linker is a program which links the compiled code with library files.

6.2 (d)

An interrupt is a signal from a device attached to a computer or from a program within the computer that causes the main program that operates the computer to stop and figure out what to do next. After the interrupt signal is sensed, it may change state of interrupted process to 'blocked' and schedule another process.

6.3 (a)

When user level process trapping the Ctrl+C signal then the trap signal is going through system call and that's why mode changed to Kernel mode from user mode and then the request is handling.

One more thing kernel mode and privilege mode are same, answer is Kernel mode (privilege mode).

6.4 (b)

In unix system: $ ln – s file 1, file 2 command replace existing file 2 by file 1. If file 2 exist then it is first removed then replace with file 1.

ln – s file 1 file 2

ln – s file 2 file 3

So, file 2 replace file 3 the file replace file 2 i.e. friends and courses are lost and hobbies are remaining.

6.6 (d)

To minimize the weighted mean completion time, jobs must be executed in increasing order of weight/time like FCFS scheduling algorithm. If we go with decreasing order of weight/time then it suffers from convoy effect.

■■■■

UNIT IX

Databases

CONTENTS

1. ER-Model 597
2. Database Design: Functional Dependencies and Normalization 602
3. Structured Query Language (SQL) 616
4. Relational Model: Relational Algebra and Tuple Calculus 632
5. Transactions and Concurrency Control 644
6. File Structures 656