DEADLOCK

By- Aditi mam

June 2014 Paper II

Consider a system with five processes P0 through P4 and three resource types R1, R2 and R3. Resource type R1 has 10 instances, R2 has 5 instances and R3 has 7 instances. Suppose that at time T0, the following snapshot of the system has been taken :

Allocation

	R_1	R_2	R_3
P_0	0	1	0
P_1	2	0	0
P_2	3	0	2
P_3	2	1	1
P_4	0	2	2

Max

R_1	R_2	R_3
7	5	3
3	2	2
9	0	2
2	2	2
4	3	3

Available

R_1	R_2	R_3
3	3	2

Assume that now the process P1 requests one additional instance of type R1 and two instances of resource type R3. The state resulting after this allocation will be

(A) Ready State (B) Safe State

(C) Blocked State (D) Unsafe State

(B) Safe State

By taking one instance of R1 , P1 will fulfill its need of maximum R1 and By taking two instances of R3 , P1 will fulfill its need of maximum R3
Now , it needs two instances of R2 after which it can complete execution and release all of its resources,
hence it is in safe state .

Paper III December 2014

An operating system has 13 tape drives. There are three processes P1, P2 & P3. Maximum requirement of P1 is 11 tape drives, P2 is 5 tape drives and P3 is 8 tape drives. Currently, P1 is allocated 6 tape drives, P2 is allocated 3 tape drives and P3 is allocated 2 tape drives. Which of the following sequences represent a safe state ?

(A) P2 P1 P3 (B) P2 P3 P1

(C) P1 P2 P3 (D) P1 P3 P2

(A) P2 P1 P3

allocated Max need

P1 6 11 5

P2 3 5 2 total resource =13 AND free resources =13-11=2 so clearly these 2 can only be given

P3 2 8 6 to P2 which will be completed and release its all 5 resource then next P1 can take 5 resource and be completed and finally P3

so right ans is choice 1 P2P1P3

Paper II June 2015

Let Pi and Pj be two processes, R be the set of variables read from memory, and W be the set of variables written to memory. For the concurrent execution of two processes Pi and Pj, which of the following conditions is not true?

(A) $R(Pi) \cap W(Pj) = \Phi$
(B) $W(Pi) \cap R(Pj) = \Phi$
(C) $R(Pi) \cap R(Pj) = \Phi$
(D) $W(Pi) \cap W(Pj) = \Phi$

(C) $R(Pi) \cap R(Pj) = \Phi$

For a concurrent execution there should be no read-write ,write-write or write-read conflict.

as given option (a),(b) and (d) depicts the same which true option.

C represent the two simultaneous read operation of the process with no intersection variables which may not be case.

Paper II December 2015

A system has four processes and five allocatable resources. The current allocation and maximum needs are as follows:

	Allocated	Maximum	Available
Process A	1 0 2 1 1	1 1 2 1 3	0 0 x 1 1
Process B	2 0 1 1 0	2 2 2 1 0	
Process C	1 1 0 1 0	2 1 3 1 0	
Process D	1 1 1 1 0	1 1 2 2 1	

The smallest value of x for which the above system in safe state is .....

(A) 1 **(B) 3**

(C) 2 **(D) Not safe for any value of x.**

	Allocated	Maximum	Available	Need
Process A	1 0 2 1 1	1 1 2 1 3	0 0 x 1 1	0 1 0 0 2
Process B	2 0 1 1 0	2 2 2 1 0		0 2 1 0 0
Process C	1 1 0 1 0	2 1 3 1 0		1 0 3 0 0
Process D	1 1 1 1 0	1 1 2 2 1		0 0 1 1 1

For x = 1 process D will execute and free 1 1 2 2 1 instances. Now none of the other process will execute.

let x = 2 then process D will execute and free 1 1 3 2 1 instances. Now process C will execute and free 2 2 3 3 1 instances. With these free instances process B will execute, but process A will not execute because 5 resources needs 2 instances which will never be satisfied. That's why system is not in safe state.

Paper III December 2015

Consider a system with twelve magnetic tape drives and three processes P1, P2 and P3. Process P1 requires maximum ten tape drives, process P2 may need as many as four tape drives and P3 may need upto nine tape drives. Suppose that at time t1, process P1 is holding five tape drives, process P2 is holding two tape drives and process P3 is holding three tape drives. At time t1, system is in:

(A) safe state (B) unsafe state

(C) deadlocked state (D) starvation state

(B) unsafe state

System is in UNSAFE state

P1 holds 5 tapes
P2 holds 2 tapes
P3 holds 3 tapes

Total 10 tapes are allocated and 12-10=2 tapes are free

P1 requires maximum 10 tape drives; It needs 10-5 =5 more tapes
P2 may need as many as 4tape drives. It needs 4-2=2 more tapes.
P3 may need as many as 9 tape drives. It needs 9-3=6 more tapes.

We can allocate 2 free tapes to P2.

P2 will complete execution and release all its resources including 4 tapes.

Even if we allocate 4 tapes to P1 or P3 they cannot complete execution because P1 nees 5 more taps while P3 needs 6 more tapes.

System is in unsafe state.

Paper II July 2016

Suppose there are four processes in execution with 12 instances of a Resource R in a system.
The maximum need of each process and current allocation are given below:

Process	Max. Need	Current Allocation
P1	8	3
P2	9	4
P3	5	2
P4	3	1

With reference to current allocation, is system safe? If so, what is the safe sequence?

(A) No **(B) Yes, P1 P2 P3 P4**
(C) Yes, P4 P3 P1 P2 **(D) Yes, P2 P1 P3 P4**

(C) Yes, P4 P3 P1 P2
P1 holds 3 resources
P2 holds 4 resources
P3 holds 2 resources
P4 holds 1 resource

Total 3+4+2+1 = 10 resources are allocated and 12-10=2 resources are free

P1 requires maximum 8 resources; It needs 8-3 =5 more tapes
P2 requires maximum 9 resources; It needs 9-4 =5 more tapes
P3 requires maximum 5 resources; It needs5-2 =3 more tapes
P4 requires maximum 3 resources; It needs 3-1 = 2 more tapes

We can allocate 2 free resources to P4.
P4 will complete execution and release all its resources. Now total 3 resoucres are free.
We can allocate 3 free resources to P3.
P3 will complete execution and release all its resources. Now total 5 resoucres are free.
We can allocate 5 free resources to P1.
P1 will complete execution and release all its resources. Now total 8 resoucres are free.
We can allocate 5 free resources to P2.
P2 will complete execution and release all its resources.
System is in safe state.

The sequence is P4P3P1P2

Paper II August 2016 (Re-test)

Consider a system with seven processes A through G and six resources R through W. Resource ownership is as follows:

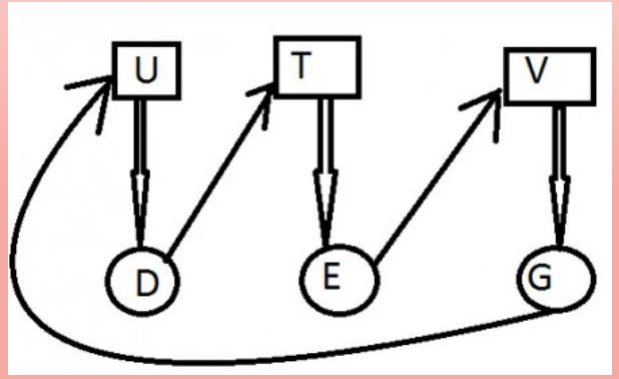
process A holds R and wants T
process B holds nothing but wants T
process C holds nothing but wants S
process D holds U and wants S & T
process E holds T and wants V
process F holds W and wants S
process G holds V and wants U

Is the system deadlocked ? If yes, ... processes are deadlocked.

(A) No
(B) Yes, A, B, C
(C) Yes, D, E, G
(D) Yes, A, B, F

(C) Yes, D, E, G

system is in Deadlock, due to process D,E,G



Paper II August 2016 (Re-test)

Consider a system having ‘m’ resources of the same type. These resources are shared by three processes P1, P2 and P3 which have peak demands of 2, 5 and 7 resources respectively. For what value of ‘m’ deadlock will not occur?

(A) 70
(B) 14
(C) 13
(D) 7

(C) 13

For deadlock free condition , at least one process will be given full no of resouces and rest of processes will be given 1 less their max demands.

So (2-1) + (5-1) + (7-1) + 1 = 1 + 4 + 6 + 1 = 12

Paper II December 2018

Suppose a system has 12 instances of some resource with n processes competing for that resource. Each process may require 4 instances of the resource. The maximum value of n for which the system never enters into deadlock is

(1) 3
(2) 4
(3) 5
(4) 6

(1) 3

Let every process acquired 3 resources now we need 1 more resource to complete one of them so that system will not face deadlock

3n+1<=12

3n<=11

n<=3.66

n=3

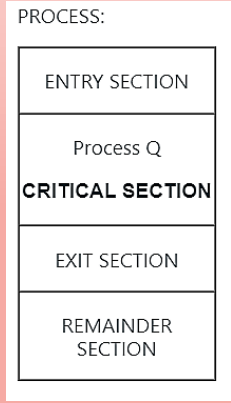
Paper II December 2018

Suppose P, Q and R are co-operating processes satisfying Mutual Exclusion condition. Then, if the process Q is executing in its critical section then

(1) Both ‘P’ and ‘R’ execute in critical section
(2) Neither ‘P’ nor ‘R’ executes in their critical section
(3) ‘P’ executes in critical section
(4) ‘R’ executes in critical section

(2) Neither ‘P’ nor ‘R’ executes in their critical section

A mutual exclusion (mutex) is a program object that prevents simultaneous access to a shared resource. A critical section is a code segment that accesses shared variables and has to be executed as an atomic action. The critical section problem refers to the problem of how to ensure that at most one process is executing its critical section at a given time.



Since process ‘Q’ is in the critical section then other processes P and R cannot be in the critical section. Therefore neither ‘P’ nor ‘R’ executes in their critical section.

Paper II June 2019

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

(a) 5
(b) 4
(c) 3
(d) 6

(a) 5

Each process needs 2 drives

Consider this scenario

P_1	P_2	P_3	P_4	P_5	P_6
1	1	1	1	1	1

This is scenario when a deadlock would happen, as each of the process is waiting for 1 more process to run to completion. And there are no more Resources available as max 6 reached. If we could have provided one more R to any of the process, any of the process could have executed to completion, then released its resources, which further when assigned to other and then other would have broken the deadlock situation.

In case of processes, if there are less than 6 processes, then no deadlock occurs.

Consider the maximum case of 5 processes.

P_1	P_2	P_3	P_4	P_5
1	1	1	1	1

In this case system has 6 resources max,and hence we still have 1 more R left which can be given to any of the processes, which in turn runs to completion, releases its resources and in turn others can run to completion too.

Paper II December 2019

Suppose a system has 12 magnetic tape drives and at time t0 , three processes are allotted tape drives out of their need as given below:

	Maximum Needs	Current Needs
p_0	10	5
p_1	4	2
p_2	9	2

At time t0 , the system is in safe state. Which of the following is safe sequence so that deadlock is avoided?

a) (P0, P1, P2)
b) (P1, P0, P2)
c) (P2, P1, P0)
d) (P0, P2, P1)

b) (P1, P0, P2)

Process	Max Need	Allocated	Need
P0	10	5	5
P1	4	2	2
P2	9	2	7
		9	

Available = 12-9 = 3
3 drives allocated to P1 first then available = 3+2 =5
5 drives allocated to P0 then available = 5+5 = 10
Then 7 drives allocated to P2.
So, safe sequence = P1→P0→P2

Paper II March 2023
Match List I with List II

LIST I		LIST II	
A.	IPC	I.	Resource Allocation
B.	Demand Paging	II.	Computational speedup
C.	Banker's Algorithm	III.	Task Control Block
D.	PCB	IV.	Virtual Memory

Choose the correct answer from the options given below:

1. A-II, B-I, C-IV, D-III
2. A-II, B-IV, C-I, D-III
3. A-I, B-II, C-III, D-IV
4. A-II, B-III, C-I, D-IV

Paper II March 2023
Match List I with List II

LIST I		LIST II	
A.	IPC	I.	Resource Allocation
B.	Demand Paging	II.	Computational speedup
C.	Banker's Algorithm	III.	Task Control Block
D.	PCB	IV.	Virtual Memory

Choose the correct answer from the options given below:

1. A-II, B-I, C-IV, D-III
2. A-II, B-IV, C-I, D-III
3. A-I, B-II, C-III, D-IV
4. A-II, B-III, C-I, D-IV

MISCELLANEOUS

BY- ADITI MAM

June 2014 Paper III

..... maintains the list of free disk blocks in the Unix file system.

(A) I-node
(B) Boot block
(C) Super block
(D) File allocation table

(C) Super block

A super block describes the state of the file system: the total size of the partition, the block size, pointers to a list of free blocks, the inode number of the root directory, magic number, etc.

June 2014 Paper III

A part of Windows 2000 operating system that is not portable is
(A) Device Management
(B) Virtual Memory Management
(C) Processor Management
(D) User Interface

(B) Virtual Memory Management

A part of windows 2000 operating system that is not portable is Virtual Memory management. The design of the VM manager assumes that the underlying hardware supports virtual-to-physical mapping, a paging mechanism, and transparent cache coherence on multiprocessor systems and allows multiple page-table entries to map to the same page frame. The VM manager in Windows uses a page-based management scheme with a page size of 4 KB.

June 2014 Paper III

Match the following with reference to Unix shell scripts :

List – I	List – II
a. $?	i. File name of the current script
b. $#	ii. List of arguments
c. $0	iii. The number of arguments
d. $*	iv. Exit status of last command

Codes :

	a	b	c	d
(A)	iii	ii	i	iv
(B)	ii	iii	i	iv
(C)	iv	iii	i	ii
(D)	i	iii	i	iv

June 2014 Paper III

Match the following with reference to Unix shell scripts :

List – I	List – II
a. $?	i. File name of the current script
b. $#	ii. List of arguments
c. $0	iii. The number of arguments
d. $*	iv. Exit status of last command

Codes :

	a	b	c	d
(A)	iii	ii	i	iv
(B)	ii	iii	i	iv
(C)	iv	iii	i	ii
(D)	i	iii	i	iv

Paper II June 2015

What does the following command do?
grep −vn “abc” x
(A) It will print all of the lines in the file x that match the search string “abc”
(B) It will print all of the lines in file x that do not match the search string “abc”
(C) It will print total number of lines in the file x that match the search string “abc”
(D) It will print the specific line numbers of the file x in which there is a match for string “abc”

Paper II June 2015

What does the following command do?
grep −vn “abc” x
(A) It will print all of the lines in the file x that match the search string “abc”
(B) It will print all of the lines in file x that do not match the search string “abc”
(C) It will print total number of lines in the file x that match the search string “abc”
(D) It will print the specific line numbers of the file x in which there is a match for string “abc”

Paper II June 2015

The Unix Kernel maintains two key data structures related to processes, the process table and the user structure. Which of following information is not the part of user structure?

(A) File descriptor table
(B) System call state
(C) Scheduling parameters
(D) Kernel stack

(C) Scheduling parameters

The Unix Kernel maintains two key data structures related to processes, the process table and the user structure. In user structure there is File descriptor table, System call state and Kernel stack there is no place for scheduling parameters. Process table stores the information of all running process.

Paper III June 2015

In ..... allocation method for disk block allocation in a file system, insertion and deletion of blocks in a file is easy.

(A) Index

(B) Linked

(C) Contiguous

(D) Bit Map

(B) Linked

In linked allocation method for disk block allocation in a file system, insertion and deletion of blocks in a file is easy.

	Contagious Method	Linked Allocation	Indexed Allocation	
1. Pre allocation	Necessary	Possible	Possible	
2. Fixed or variable size portions.	Variable	Fixed	Fixed	Variable
3. Portion size	Large	Small	Small	Medium
4. Allocation frequency	Once	Low to high	High	Low
5. Table size	One entry	One entry	Large	Medium
6. Access type	Random access	Direct access	Direct access	
7. Fragmentation	External	No external fragmentation	NO	

Paper III June 2015

A unix file may be of type:

(A) Regular file
(B) Directory File
(C) Device File
(D) Any one of the above

Paper III June 2015

A unix file may be of type:

(A) Regular file

(B) Directory File

(C) Device File

(D) Any one of the above

Paper II December 2015

In Unix, the login prompt can be changed by changing the contents of the file .....

(A) contrab
(B) init
(C) gettydefs
(D) inittab

Paper II December 2015

In Unix, the login prompt can be changed by changing the contents of the file .....

(A) contrab

(B) init

(C) gettydefs

(D) inittab

Paper III December 2015

In Unix operating system, special files are used to :

(A) buffer data received in its input from where a process reads

(B) provide a mechanism to map physical device to file names

(C) store list of file names plus pointers associated with i-nodes

(D) store information entered by a user application program or utility program

(B) provide a mechanism to map physical device to file names

- provides a mechanism to map physical device to file names
- Windows and DOS also have special files.
- provide simple interfaces between drivers and peripheral devices such as printers and serial ports

Paper III December 2015

Match the following in Unix file system :

List - I	List - II
(a) Boot block	(i) Information about file system
(b) Super block	(ii) Information about file
(c) Inode table	(iii) Storage space
(d) Data block	(iv) Code for making OS ready

Codes :

	(a)	(b)	(c)	(d)
(A)	(iv)	(i)	(ii)	(iii)
(B)	(i)	(iii)	(ii)	(iv)
(C)	(iii)	(i)	(ii)	(iv)
(D)	(iv)	(ii)	(i)	(iii)

Paper III December 2015

Match the following in Unix file system :

List - I	List - II
(a) Boot block	(i) Information about file system
(b) Super block	(ii) Information about file
(c) Inode table	(iii) Storage space
(d) Data block	(iv) Code for making OS ready

Codes :

	(a)	(b)	(c)	(d)
(A)	(iv)	(i)	(ii)	(iii)
(B)	(i)	(iii)	(ii)	(iv)
(C)	(iii)	(i)	(ii)	(iv)
(D)	(iv)	(ii)	(i)	(iii)

Paper III December 2015

In an operating system, indivisibility of operation means :
(A) Operation is interruptible
(B) Race - condition may occur
(C) Processor can not be pre-empted
(D) All of the above

(C) Processor can not be pre-empted

The indivisibility of operation in an operating system means that the processor cannot be pre-empted. Once a process begins to run, it will not be halted or stopped inside the CPU.

Paper II July 2016

In UNIX, ... creates three subdirectories: 'PIS' and two subdirectories 'progs' and ‘data’ from just created subdirectory 'PIS'.

(A) mdkir PIS/progs PIS/data PIS

(B) mkdir PIS progs data

(C) mkdir PIS PIS/progs PIS/data

(D) mkdir PIS/progs data

Paper II July 2016

In UNIX, ... creates three subdirectories: 'PIS' and two subdirectories 'progs' and ‘data’ from just created subdirectory 'PIS'.

(A) mdkir PIS/progs PIS/data PIS

(B) mkdir PIS progs data

(C) mkdir PIS PIS/progs PIS/data

(D) mkdir PIS/progs data

Paper III July 2016

In UNIX operating system, when a process creates a new process using the fork() system call, which of the following state is shared between the parent process and child process?

(A) Heap

(B) Stack

(C) Shared memory segments

(D) Both Heap and Stack

(C) Shared memory segments

The parent and child share only the shared memory segments.

Everything else (like stack , heap etc) is duplicated

Paper III July 2016

Which of the following information about the UNIX file system is not correct?

(A) Super block contains the number of i-nodes, the number of disk blocks, and the start of the list of free disk blocks.

(B) An i-node contains accounting information as well as enough information to locate all the disk blocks that holds the file's data.

(C) Each i-node is 256-bytes long.

(D) All the files and directories are stored in data blocks.

Paper III July 2016

Which of the following information about the UNIX file system is not correct?

(A) Super block contains the number of i-nodes, the number of disk blocks, and the start of the list of free disk blocks.

(B) An i-node contains accounting information as well as enough information to locate all the disk blocks that holds the file's data.

(C) Each i-node is 256-bytes long.

(D) All the files and directories are stored in data blocks.

Paper III July 2016

Which of the following option with reference to UNIX operating system is not correct?

(A) INT signal is sent by the terminal driver when one types <Control-C> and it is a request to terminate the current operation.

(B) TERM is a request to terminate execution completely. The receiving process will clean up its state and exit.

(C) QUIT is similar to TERM, except that it defaults to producing a core dump if not caught.

(D) KILL is a block able signal.

Paper III July 2016

Which of the following option with reference to UNIX operating system is not correct?

(A) INT signal is sent by the terminal driver when one types <Control-C> and it is a request to terminate the current operation.

(B) TERM is a request to terminate execution completely. The receiving process will clean up its state and exit.

(C) QUIT is similar to TERM, except that it defaults to producing a core dump if not caught.

(D) KILL is a block able signal.

Paper III July 2016

A Multicomputer with 256 CPUs is organized as 16x16 grid. What is the worst case delay (in hops) that a message might have to take?

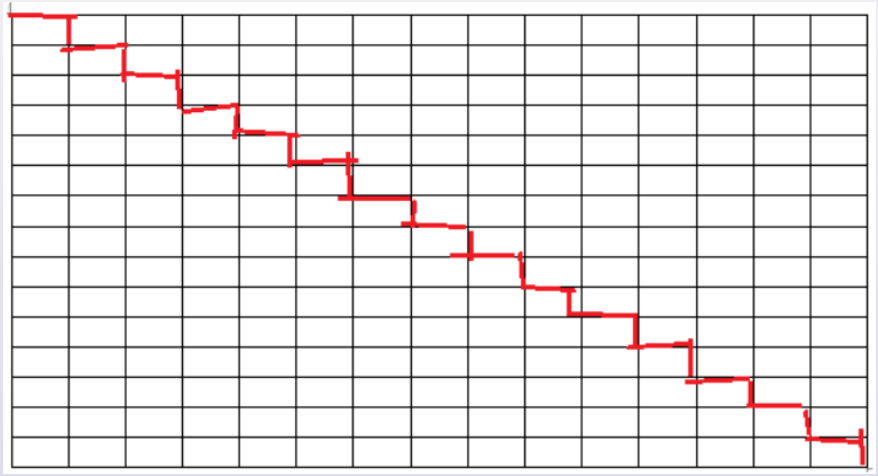
(A) 16

(B) 15

(C) 32

(D) 30

(D) 30



In the 16X16 CPU case, the worst-case delay happens when message passes through longest path which is the diagonal of the upper right corner to lower left corner or upper left corner to lower right corner.

Longest path is indicated in red color.

It passes through 2(N-1) = 2(16-1) = 30 hops

Paper III July 2016

Suppose that the time to do a null remote procedure call (RPC) (i.e, 0 data bytes) is 1.0 msec, with an additional 1.5 msec for every 1K of data. How long does it take to read 32 K from the file server as 32 1K RPCs?

(A) 49 msec

(B) 80 msec

(C) 48 msec

(D) 100 msec

(B) 80 msec

A single 32K RPC takes 1.5*32 + 1.0 = 49.0 msec

32 1K RPCs take 1.5*32 + 1.0*32 = 80.0 msec

Paper III August 2016 (Re-test)

The Unix Operating System Kernel maintains two key data structures related to processes, the process table and the user structure. Now, consider the following two statements:

I. The process table is resident all the time and contain information needed for all processes, even those that are not currently in memory.

II. The user structure is swapped or paged out when its associated process is not in memory, in order not to waste memory on information that is not needed.

Which of the following options is correct with reference to above statements ?

(A) Only (I) is correct.

(B) Only (II) is correct.

(C) Both (I) and (II) are correct.

(D) Both (I) and (II) are wrong.

Paper III August 2016 (Re-test)

The Unix Operating System Kernel maintains two key data structures related to processes, the process table and the user structure. Now, consider the following two statements:

I. The process table is resident all the time and contain information needed for all processes, even those that are not currently in memory.

II. The user structure is swapped or paged out when its associated process is not in memory, in order not to waste memory on information that is not needed.

Which of the following options is correct with reference to above statements ?

(A) Only (I) is correct.

(B) Only (II) is correct.

(C) Both (I) and (II) are correct.

(D) Both (I) and (II) are wrong.

Paper III August 2016 (Re-test)

Consider a file currently consisting of 50 blocks. Assume that the file control block and the index block is already in memory. If a block is added at the end (and the block information to be added is stored in memory), then how many disk I/O operations are required for indexed (single-level) allocation strategy ?

(A) 1

(B) 101

(C) 27

(D) 0

(A) 1

Since file control block and the index block is already in memory we need not fetch them from disc.

We need to copy the new block to be added from disc (1 I/O operation). that's it

Paper III January 2017

Match the following for Unix rue system:

List-I	List-II
a. Boot block	i. Information about file system, free block list, free inode list etc.
b. Super block	ii. Contains operating system files as well as program and data files created by users.
c. Inode block	iii. Contains boot program and partition table.
d. Data block	iv. Contains a table for every file in the file system. Attributes of files are stored here.

Codes:

	a	b	c	d
(1)	iii	i	ii	iv
(2)	iii	i	iv	ii
(3)	iv	iii	ii	i
(4)	iv	iii	i	ii

Paper III January 2017

Match the following for Unix rue system:

List-I	List-II
a. Boot block	i. Information about file system, free block list, free inode list etc.
b. Super block	ii. Contains operating system files as well as program and data files created by users.
c. Inode block	iii. Contains boot program and partition table.
d. Data block	iv. Contains a table for every file in the file system. Attributes of files are stored here.

Codes:

	a	b	c	d
(1)	iii	i	ii	iv
(2)	iii	i	iv	ii
(3)	iv	iii	ii	i
(4)	iv	iii	i	ii

Paper II November 2021

In a file allocation system, the following allocation schemes are used:

A. Contiguous
B. Indexed
C. Linked allocation

Which of the allocation scheme(s) given above will not suffer from external fragmentation? Choose the correct answer from the options given below:

a) A and B only
b) A only
c) B and C only
d) C only

c) B and C only

In contiguous allocation there is always possibility of external fragmentation. Both indexed and linked allocation are free from external fragmentation.

Paper II JUNE 2023

The multiuser operating system, 30 requests are made to use a particular resource per hour, on an average the probability that no request are made in 40 minutes is

(A) e^{-15}
(B) e^{-20}
(C) $1 - e^{-15}$
(D) $1 - e^{-20}$

(A) e^{-20}

A Poisson process is a stochastic process that counts the number of events and time points at which these events occur in a given time interval. An event can occur 0, 1, 2, … times in an interval. The average number of events in an interval is designated lambda. Lambda is the event rate, also called the rate parameter. The probability of observing k events in an interval is given by the equation

Formula for poisson distribution:

λ is the average number of events per interval

Explanation:

30 requests are sent in 1 hour.

i.e. in 60 min, 30 requests are sent

in 40 min, number of requests sent are 20.

λ = 20. k = 0

$P(0, 20) = 20^0 * e^{-20} / 0!$

Probability that no requests are made in 40 minutes = e^{-20}

Paper II JUNE 2023

Match List I with List II

LIST I		LIST II	
A.	RAID level 2	I.	Bit interleaved parity
B.	RAID level 4	II.	Block interleaved distributed parity
C.	RAID level 5	III.	Error correcting parity
D.	RAID level 3	IV.	Block interleaved parity

Choose the correct answer from the options given below:

a) A - IV, B - III, C - I, D - II
b) A - III, B - IV, C - II, D - I
c) A - III, B - I, C - II, D - IV
d) A - I, B - III, C - IV, D - I

B) A - III, B - IV, C - II, D - I

- **RAID level 2 (III - Error correcting parity): RAID 2 uses an error-correcting code known as Hamming code, which is a set of parity bits for error detection and correction. Each bit of data is written to a separate disk drive in the array, and corresponding parity bits are written to additional drives, which allows the error correction to occur.**
- **RAID level 4 (IV - Block interleaved parity): In RAID 4, data is split into blocks and written across multiple drives in an array (i.e., block-level striping). But unlike other RAID levels, RAID 4 has a dedicated disk for storing parity information. This configuration allows for high read data transaction rates because the data blocks and the parity are stored on different drives.**
- **RAID level 5 (II - Block interleaved distributed parity): The parity information, which is used for data redundancy, is not written to a single, dedicated drive as in RAID 4. Instead, it is interspersed across all the drives in the array. This leads to a system where read and write operations can occur simultaneously on multiple drives, improving overall performance compared to RAID 4.**
- **RAID level 3 (I - Bit interleaved parity): RAID 3 is similar to RAID 2 but it uses a simpler parity calculation. Data is split at the bit level and written across the drives in the array (i.e., bit-level striping) with parity being stored on a dedicated drive.**

Paper II December 2023

In Linux, where is the user password stored ?

(1) /etc/password

(2) /root/password

(3) /etc/passwd

(4) /root/passwd

In Linux, where is the user password stored ?

(1) /etc/password

(2) /root/password

(3) /etc/passwd

(4) /root/passwd

Paper II December 2023

Indexed/grouped allocation is useful as:

(A) It supports both sequential and direct access.

(B) Entire block is available for data.

(C) It does not require lots of space for keeping pointers.

(D) No external fragmentation.

Choose the correct answer from the options given below:

(1) (A) Only

(2) (B) and (C) Only

(3) (B) Only

(4) (A), (B) and (D) Only

(4) (A), (B) and (D) Only

(A) It supports both sequential and direct access. Indexed/grouped allocation allows both sequential (one block after another) and direct (jumping straight to a desired block) access to data.

(B) Entire block is available for data. In indexed allocation, the entire block is indeed available for storing data. The index block, a separate entity, contains the pointers to the data blocks.

(D) No external fragmentation. Indexed allocation helps prevent external fragmentation as all files have their own index block that keeps track of the blocks associated with it.