Year: 2020 (Fall)

Section A

Q.No.1

Ans. A system call is a request from computer software to an operating system's kernel where API (Application program Interface) connects the operating system's functions to user programs.

Some of the major services of operating system are:-

i, Process Management
ii, Memory Management
iii, File management
iv, Device Management
v, Security
vi, Network Management
vii, User Interface

Q.No.2,

Ans The difference between multiprogramming and multiprocessing are:-

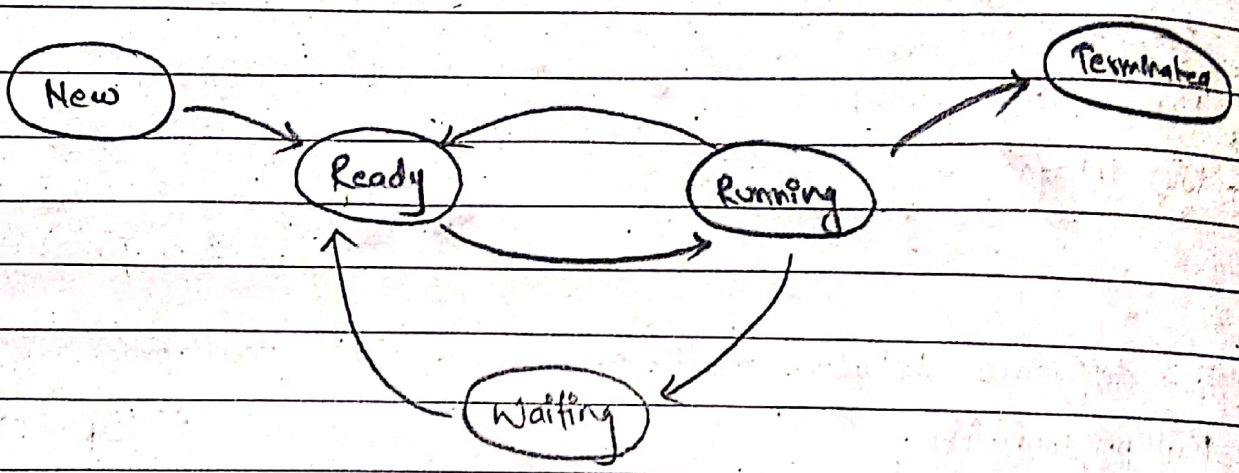
	Multiprogramming	Multiprocessing
i,	The concurrent execution of more than one program in main memory is known as multiprogramming.	The availability of more than one processor per system that can execute a set of instruction in parallel is known as multiprocessing.
ii,	Throughput is less.	Throughput is maximum
iii,	The no. of users can be only one.	The no. of users can be one or more than one
iv,	Only one process can be executed at a time.	More than one process can be executed at a time.

Q.No. 3

Ans The difference between Process and thread are:-

	Process	Thread.
i,	Process is a program in a execution	Thread is a segment of a program.
ii,	It takes more time for creation.	It takes less time for creation
iii,	It takes more ~~than~~ time for terminate.	It takes less time for termin
iv,	It doesnot share data to each other.	It share data to each other

Q.No.4 The process states are:-



Q.No.4

Ans CPU scheduling is the task performed by the CPU that decides the way and order in which processes should be executed.

The criteria for CPU scheduling are:-

i, CPU utilization
ii, Throughput
iii, Turnaround Time
iv, Waiting time
v, Response Time

vi, Completion Time
vii, Priority
viii, Predictability.

Q.No.5
Ans. Race condition or Race Hazard is an undesirable situation of software, electronics or other systems. When the output of the system or program depends on the sequence or timing of other uncontrolled events, this condition is called Race Condition.

Some solution to race condition are:-

i, Mutual exclusion
ii, Synchronize the process.

Q.No.6
Ans. Starvation is the problem that occurs when high priority processes keep executing and low priority processes get blocked for indefinite time. In starvation, resources are continously utilized by high priority processes

Q.No.7
Ans. Logical Address is generated by CPU while a program is running that can used by the user to access the physical address.

Physical Address identifies a physical location of required data in a memory that can indirectly access physical address by the user but not directly.

Q.No.8
Ans. Thrashing is when the page fault and swapping happens very frequently at a higher rate, and then the operating system has to spend

more time scrapping these pages.

Q.No.9

Ans A file descriptor is a number that uniquely identifies an open file in a computer's operating system that describes a data resource, and how that resource may be accessed.

Q.No.10

Ans Distributed system is a collection of autonomous computer systems that are physically separated but are connected by a centralized computer network that is equipped with distributed system software.

The advantages over centralized system are:-

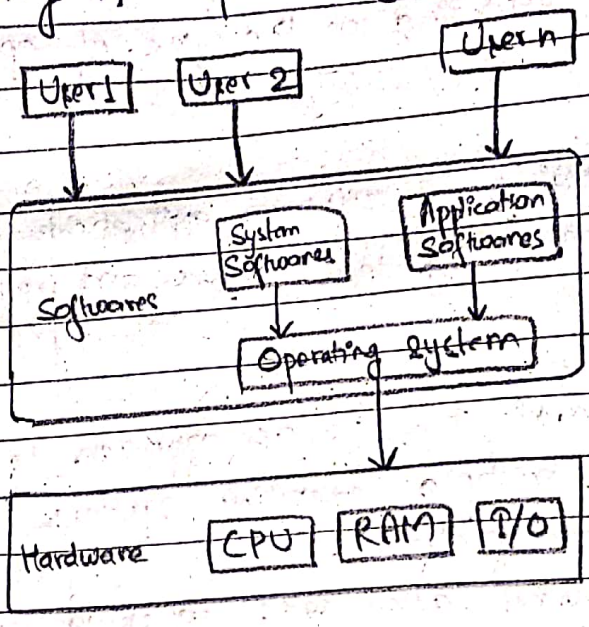
i. Scalability.
ii. Availability
iii. Fault tolerance
iv. Performance
v. Security

Section "B"

Q.No.11

Ans. Operating System is a software which performs all the basic tasks like file management, memory management, process management, handling input and output and controlling peripheral devices.

The architecture of an operating system refers to its organization and structure, including its components, layers and interfaces.



Most operating systems follow a similar architecture that can be divided into four main layers or components :-

i) Hardware Layer:-

The operating system interacts with the hardware layer through devices, ~~and~~ drivers, which provide an interface bet" the hardware and the software layers.

ii) Kernel Layer:-

This layer provides the core functionality of the operating system, including process management, memory, input/~~dpu~~ output, and file system management. The kernel is responsible for managing system resources,

allocating them to different processes, and enforcing security policies.

iii) Shell Layer:-

This layer provides a user interface to interact with the operating system, allowing users to run commands and launch applications. The shell can either be a command-line interface (CLI) or a graphical user interface (GUI), depending on the type of operating system.

iv) Application Layer:-

This layer consists of software applications that run on top of the operating system, using its services and resources to perform specific tasks.

Virtual machine architecture is an approach to operating system design that allows multiple operating systems to run concurrently on the same hardware platform.

There are several advantages to using virtual machine architecture for operating system designers:-

i) Isolation:-

Virtual machines provide a high degree of isolation that runs in its own isolated environments which helps prevent conflicts bet^n applications & ensures that each operating system runs reliably and securely.

ii) Resource allocation:-

This allows designers to optimize resource usage and ensure that each operating system has access to the resources it needs to run efficiently.

iii, **Testing:-**

Virtual machines provide a safe and controlled environment for testing operating systems and applications. Designers can create multiple virtual machines with different configurations and test their software in a variety of environments, without effecting the underlying hardware or other virtual machines.

iv, **Flexibility:-**

Virtual machines provides high degree of flexibility, allowing designers to add or remove virtual machines as needed, depending on the requirements of the system.

v, **Portability:-**

Virtual machine provide high degree of portability, which means that an operating system designed for a virtual machine environment can run on any hardware platform that supports virtualization.

Q.No-13

Ans. IPC (Inter-process communication) and synchronization are necessary in computing system to allow multiple processes or threads to communicate and coordinate with each other. In a multi-process or multi-threaded environment, processes/threads may need to access shared resources or data structures concurrently. However, if these processes/threads are not synchronized, they can interfere with each other's operations and lead to race conditions, deadlocks, and other concurrency-related issues.

To solve the producer-consumer problem using semaphores, we can use two semaphores: a mutex semaphore (for mutual exclusion) and a counting semaphore (for synchronization). The producer acquires the mutex semaphore before producing a data item and releases it afterward, while the consumer acquires the counting semaphore before consuming a data item and releases it afterward.

Here's how the solution works:-

i) The producer starts with an empty buffer and produces an item.

ii) The producer acquires the mutex semaphore to ensure exclusive access to the buffer.

iii) If the buffer is full, the producer releases the mutex semaphore and waits on the counting semaphore (which represents the no. of available slots in the buffer) until a slot is available.

iv) Once a slot is available, the producer adds the item to the buffer, releases the mutex semaphore, and signals the counting semaphore (incrementing it by 1) to indicate that an item is available in the buffer.

v) The consumer starts with a full buffer and consumes an item.

vi) The consumer acquires the mutex semaphore to ensure exclusive access to the buffer.

vii) If the buffer is empty, the consumer releases the mutex semaphore and waits on the counting semaphore (which represents the no. of items in the buffer) until an item is available.

viii) Once an item is available, the consumer removes it from the buffer, releases the mutex semaphore, and signals the counting semaphore (decrementing it by 1) to indicate that a slot is available in the buffer.

This solution doesnot have a busy wait condition because when a process is waiting for a semaphore, it goes to sleep until the semaphore is signaled by another process. This means that the process is not consuming CPU resources while waiting for the semaphore.

Q.No.4

a)

Ans. A file is a collection of logically related data that is recorded on the secondary storage in the form of sequence of operations.

There are several file operations that can be performed in an operating system, including:-

i) Creating a file:- This operation involves creating a new file and assigning it a name and location in the file system.

ii) Opening a file: This operation involves opening an existing file for reading, writing or both.

iii) Closing a file: This operation involves closing an open file after reading or writing operations are completed.

iv) Reading from a file: This operation involves retrieving data from a file.

v) Writing to a file:- This operation involves adding or modifying data in a file.

vi) Seeking:- This operation involves moving the file pointer to a specific location in the file.

vii) Deleting a file:- This operation involves removing a file from the file system.

Now, coming to file allocation methods, there are three common file allocation methods used in operating systems:-

i) Contiguous allocation
ii) Linked allocation
iii) Indexed allocation.

i) Contiguous allocation:-

In contiguous allocation, a file is allocated a contiguous block of disk space. This means that the file occupies a continuous sequence of blocks on the disk. This method is simple to implement, but it suffers from fragmentation as files are created and deleted.

ii) Linked allocation

In linked allocation, each file is made up of a linked list of disk blocks, where each block contains a pointer to the next block in the file. This method eliminates fragmentation, but it requires extra disk space to store the pointers.

iii, Indexed Allocation:-

In indexed allocation, each file has its own index block that contains a list of pointers to the actual data blocks of the file. This method allows for fast access to individual blocks of the file and eliminates fragmentation, but it requires extra disk space to store the index blocks.

Q.NO.14

b>

Ans. Page replacement is a technique used by operating systems to manage memory when the available memory is not sufficient to accomodate all the processes that are currently running. In such situations, some pages of a process may need to be removed from the memory to make space for new pages. This process of removing pages from memory and bringing them back when required is known as page replacement.

There are several page replacement algorithms that are used by operating systems ~~to decide which~~ are briefly described below with example.

i) OPR:

OPR stands for Optimal Page Replacement. OPR is an optimal algorithm that selects the page that will not be used for the longest period of time in the future. In other words, it selects the page that will have the longest time interval before it is referenced again. OPR requires the knowledge of future page references, which is not possible in real systems.

Example:-

Number of frame, fn = 4.

Reference string, pg[] = {7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2};

7	0	1	2	0	3	0	4	2	3	0	3	2
7	7	7	7		3		3					
	0	0	0		0		0					
		1	1		1		4					
			2		2		2					

No. of hits = 7

page fault = 6.

ii) LRU:-

LRU stands for Least Recently Used Algorithm. The LRU algorithm selects the page that has not been used for the longest period of time. It assumes that the pages that have not been used for a long time are less likely to be used in the near future.

Example:

Number of frame, fn = 4

Reference string, pg[] = {7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2};

7	0	1	2	0	3	0	4	2	3	0	3	2
7	7	7	7		3		3					
	0	0	0		0		0					
		1	1		1		4					
			2		2		2					

page fault = 6

page hit = 7.

Q.No.16

Ans There are several security issues related to operating systems, such as:

i) Malware:

Malware, such as viruses, worms, and Trojans, can infect an operating system and cause harm by stealing sensitive data, disrupting system performance, and controlling the system remotely.

ii) Unauthorized access:

Unauthorized users may attempt to access a system, either locally or remotely, to steal or manipulate data or to cause damage to the system.

iii) Denial of service (DoS): Attackers may use DoS attack to overwhelm a system, preventing legitimate users from accessing the system's resources.

iv) Vulnerabilities:

Operating systems are often complex, and vulnerabilities may be discovered that could be exploited by attackers.

To address these security issues, operating systems implement various protection mechanisms and access control mechanisms. Two important access control mechanisms are:-

i) Protection mechanism:

This mechanism uses hardware and software features to protect the operating system and its resources from unauthorized access or modification. This includes features such as memory protection, input/output protection, and system call interposition.

Q1, **Access Control List (ACL):**

ACL is a list of permissions attached to a file or folder that specifies who can access the resource and what actions they can perform. ACLs typically include entries for users, groups, and system processes, and they can be used to restrict access to sensitive resources.

An ACL typically includes the following permissions:-

- Read: Allows a user to read the contents of a file or folder.
- Write: Allows a user to modify the contents of a file or folder.
- Execute: Allows a user to execute a file or access the contents of a folder.
- Delete: Allows a user to delete a file or folder.

ACLs can be used to implement various access control policies, such as mandatory access control (MAC) or discretionary access control (DAC). MAC is a policy that specifies what actions a user can perform based on their security clearance, while DAC allows users to control access to resources they own.

Q.No.17

Ans. Distributed systems are computer systems that consist of multiple independent computers connected by a communication network. Some of the design issues related to distributed systems are:-

i) Communication: Communication between nodes is crucial in a distributed system. A communication network that connects the nodes must be reliable, and messages must be delivered in a timely manner.

ii) Consistency and Replication:-

In a distributed system, data is often replicated to improve availability and performance. However, maintaining consistency among replicas can be a challenge.

iii) Fault tolerance:

Distributed systems must be designed to tolerate failures of individual nodes, communication links and even entire sub-networks.

iv) Scalability:

A distributed system must be designed to handle a growing number of nodes and increasing traffic.

v) Security:

Distributed systems must be designed to prevent unauthorized access to data and prevent malicious attacks.

Remote Procedure Call (RPC) Mechanism:-

Remote Procedure Call (RPC) is a mechanism that allows a program to invoke a procedure on a remote computer as if it were a local procedure. RPC hides the complexities of the underlying network and provides a simple programming model for distributed computing.

RPC works as follows:-

i) A client program calls a procedure on a remote server as if it were a local procedure.

ii) The client stub (a local program) marshals the procedure arguments into a message and sends it over the network to the server.

iii) The server stub receives the message and unmarshals the arguments.

iv) The server program executes the procedure with the arguments.

v) The server marshals the results into a message and sends it back to the client.

vi) The client stub receives the message and unmarshals the results.

vii) The client program continues execution with the results.

RPC provides a transparent way of programs to call procedures on remote computers, making it easy to build distributed applications. However, RPC doesnot address many of the challenges of distributed computing, such as reliability, consistency and fault tolerance. These issues must be addressed by other mechanisms, such as distributed transactions and replication.

Q No. 12.

Ans. Given,

Process	Arrival Time	Burst Time	Priority
A	0	4	2(L)
B	2	1	4
C	5	3	3
D	7	6	5(H)

Turnaround Time (TAT) = Completion time − Arrival Time

Waiting Time (WT) = Turnaround Time − Burst Time.

a) FCFS

A	B	C	D

0 4 5 8 14.

Process	TAT	WT
A	4 − 0 = 4	4 − 4 = 0
B	5 − 2 = 3	3 − 1 = 2
C	8 − 5 = 3	3 − 3 = 0
D	14 − 7 = 7	7 − 6 = 1
	ATAT = 17/4	AWT = 3/4
	= 4.25	= 0.75

b) SJF (preemptive)

A	B	A	C	D
0	2	3	5	8	14

Process	TAT	WT
A	5 - 0 = 5	5 - 4 = 1
B	3 - 2 = 1	1 - 1 = 0
C	8 - 5 = 3	3 - 3 = 0
D	14 - 7 = 7	7 - 6 = 1
	ATAT = 16/4	AWT = 2/4
	= 4	= 0.5

c) SJF (non-preemptive)

A	B	C	D
0	4	5	8	14

Process	TAT	WT
A	4 - 0 = 4	4 - 4 = 0
B	5 - 2 = 3	3 - 1 = 2
C	8 - 5 = 3	3 - 3 = 0
D	14 - 7 = 7	7 - 6 = 1
	ATAT = 17/4	AWT = 3/4
	= 4.25	= 0.75

e, Round-robin (quantum = 3)

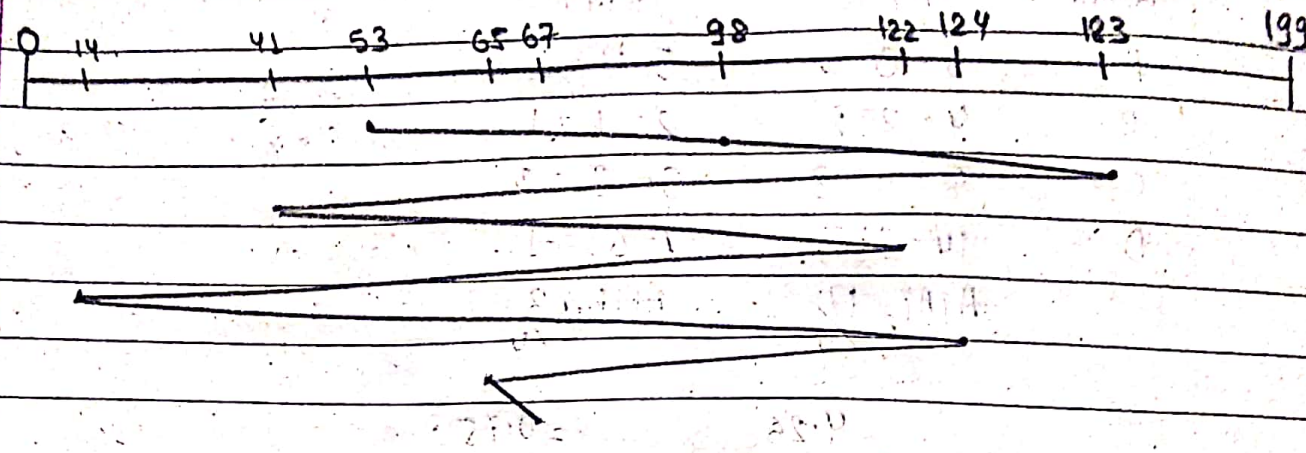
A	B	A	C	D	D
0	3	4	5	8	11	14

Process	TAT	WT
A	5 - 0 = 5	5 - 4 = 1
B	4 - 2 = 2	2 - 1 = 1
C	8 - 5 = 3	3 - 3 = 0
D	14 - 7 = 7	7 - 6 = 1.
	ATAT = 17/4	AWT = 3/4.
	= 4.25	= 0.75.

Q.No.15. Given,

Ans. 98, 183, 41, 122, 14, 124, 65, 67.

a) FCFS.

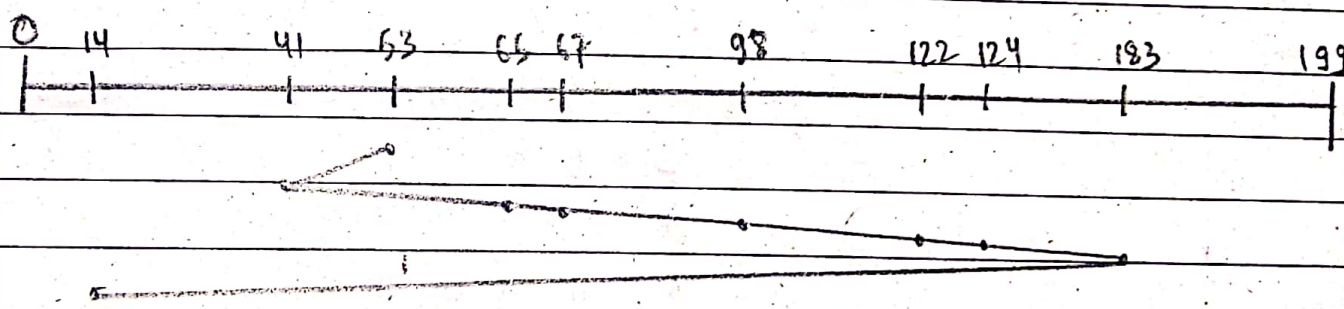


Total head movement:

= 130 + 142 + 81 + 108 + 110 + 59 + 2

= 632

b, SSTF.

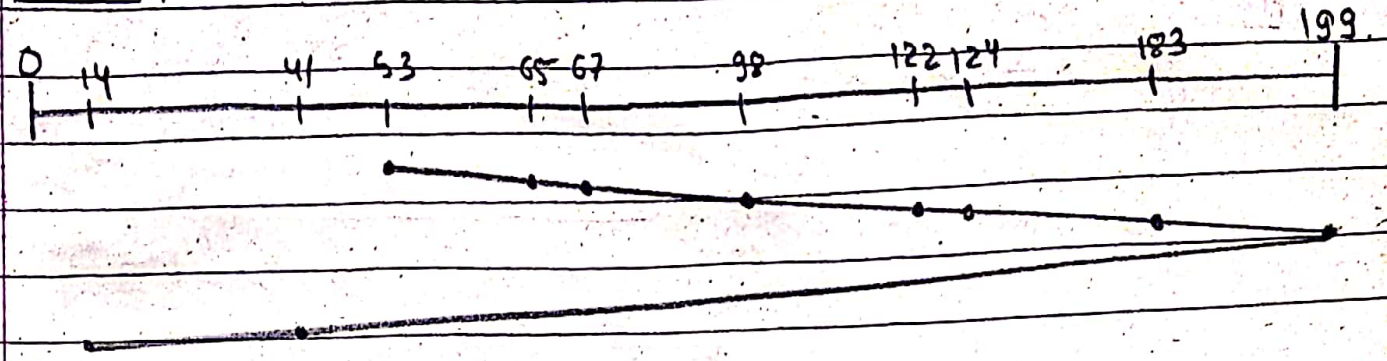


Total head movement.

= 142 + 12 + 169

= 323.

c, SCAN.

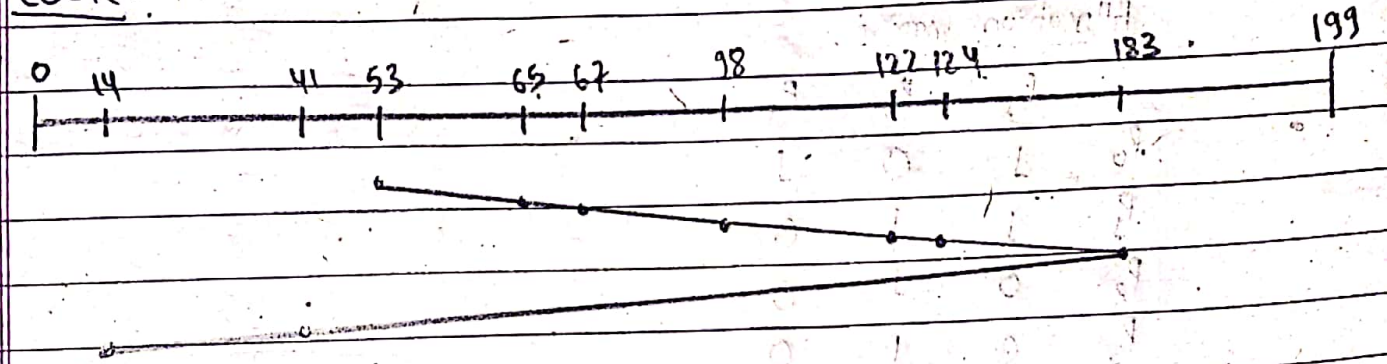


Total head movement.

= 146 + 185.

= 331.

d, LOOK.

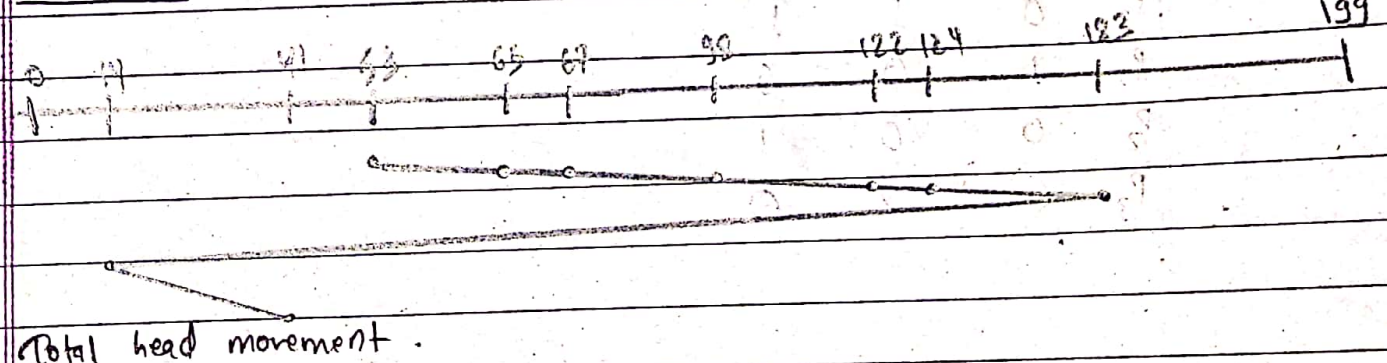


Total head movement.

= 130 + 169

= 299.

e, C-LOOK.



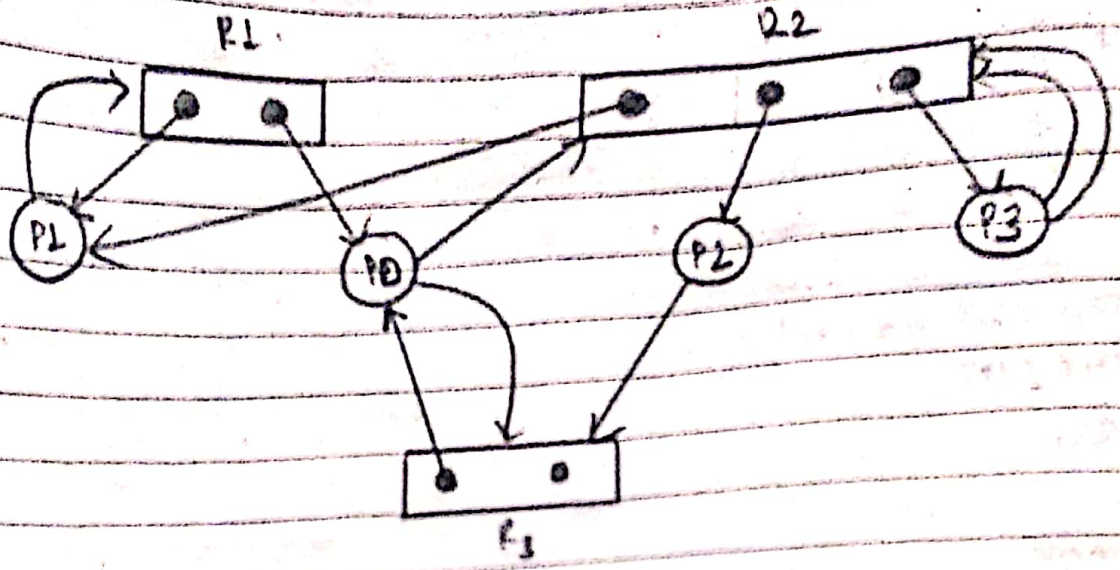
Total head movement.

= 130 + 169 + 27

= 326.

Q.No.18.

Ans.



Soln,

Allocation matrix.

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

Need Matrix

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

Existing Resource (E)

	R_1	R_2	R_3
Existing resource (E)	2	3	2
Held (H)	2	3	1
Available (A)	0	0	1

Step 1 : Allocation Matrix.

	R_1	R_2	R_3
P_0	1	0	1
P_1	1	1	0
P_2	0	1	1
P_3	0	1	0

Need Matrix.

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

E	2	3	2
H	2	2	1
A	0	1	1

Step 2 : Allocation Matrix

	R_1	R_2	R_3
P_0	1	1	2
P_2	1	1	0
P_2	0	0	0
P_3	0	1	0

Need Matrix

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

E	2	3	2
H	1	2	0
A	1	1	2

Step 3: Allocation matrix

	R_1	R_2	R_3
P_0	0	0	0
P_1	2	1	0
P_2	0	0	0
P_3	0	1	0
E	2	3	2
H	0	1	0
A	2	2	2

Need matrix:

	R_1	R_2	R_3
P_0	0	0	0
P_1	0	0	0
P_2	0	0	0
P_3	0	2	0

Step 4: Allocation matrix.

	R_1	R_2	R_3
P_0	0	0	0
P_1	0	0	0
P_2	0	0	0
P_3	0	2	0
E	2	3	2
H	0	0	0
A	2	3	2

Need matrix.

	R_1	R_2	R_3
P_0	0	0	0
P_1	0	0	0
P_2	0	0	0
P_3	0	0	0

a)

Ans. So, there is safe sequence, Hence there is no deadlock.

b,

Ans. Prevention method used in deadlocks in an operating system include:-

i) Resource Allocation Graph:-

A directed graph used to detect cycles that could lead to deadlocks.

ii) Banker's Algorithm:-

A resource allocation algorithm that requires processes to declare their maximum resource requirements before running and prevents deadlocks by denying resource allocation.

iii) Timeouts: The operating system sets a time limit for resource allocation, and if a process fails to acquire the needed resources within the specified time, it is aborted.

iv) Avoidance:-

Detecting the likelihood of a deadlock and adjusting the allocation of resources to prevent it.

v) Deadlock Recovery and Detection:-

The operating system detects deadlocks after they have occurred and takes steps to recover from them, such as aborting one or more processes or releasing resources.