OS Assignment

THA076BCT038 Sajjan Achaiya

2078 Chaitra, Regular.

1) In an operating system, the set of extended instructions that defines an interface between the operating system and the user program is called system call. It plays an important role in the OS as it allows a computer program to request a service from the kernel of the operating system.

The major differences between the following types of operating system are written as follows:

a) Batch system

i. Similar jobs are collected and same tasks are executed repeatedly in the batches without intervention.

ii. Initial commands are set up, but input of data is not possible.

iii. There is difficulty to provide the desired priority.

b) Interactive System:

i. The user can interact directly with OS to supply commands and data as the application program executes.

ii. The user receives output immediately.

iii. An user interface may be provided.

c) Real time system

i. The processing time to user inputs and the response time is very small and critical.

ii. Quick input is taken to give precisely immediate output to user.

iii. Inputs can be given based on priority.

d) Time sharing system

i. More than one users receive the output from the OS

ii. The resources are shared and can handle multiple inputs and different tasks.

iii. It provides advantage of quick response and reduces the CPU idle time.

Q No. 2

5 state process model is a common model used in the operating system design to describe the life cycle of a process. A demo of 5 state process model is given as follows:

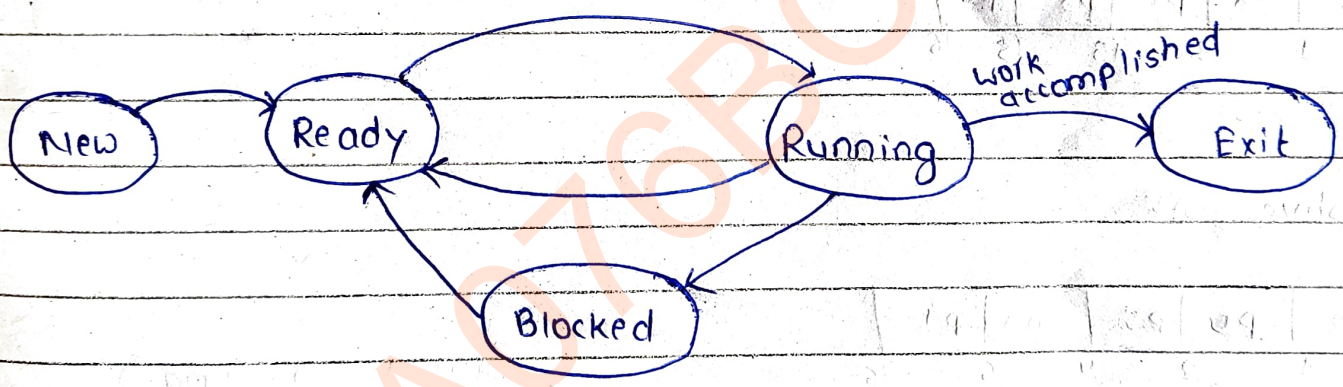


Fig: 5 state process model

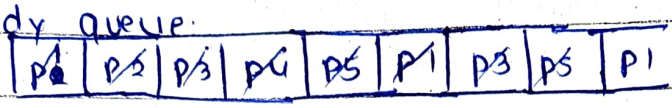
Given:

Process	Burst Time	Priority
p1	10	3
p2	1	1
p3	2	3
p4	1	4
p5	2	2

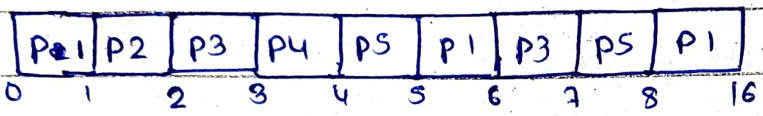
a) Gantt chart:

i. Round Robin (quantum = 1)

Ready queue:

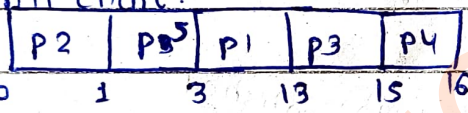


Gantt chart:



ii) Priority pre emptive:

Gantt chart:



iii) Preemptive SJF

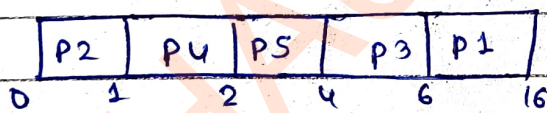


fig: Gantt chart for preemptive SJF

iv) FCFS

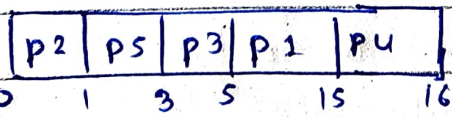


fig: Gantt chart for FCFS

TAT = CT - AT

WT = TAT - BT

Computation

i) Round Robin (Quantum = 1)

Process	AT	BT	CT	TAT	WT
p1	0	10	16	16	6
p2	0	1	2	2	1
p3	0	2	7	7	5
p4	0	1	4	4	3
p5	0	2	8	8	6
				= 37	= 21

Av. TAT = $\frac{37}{5}$ = 7.4

Av. WT = $\frac{21}{5}$ = 4.2

ii) Priority pre emptive

process	AT	BT	CT	TAT	WT
p1	0	10	13	13	3
p2	0	1	1	1	0
p3	0	2	15	15	13
p4	0	1	16	16	15
p5	0	2	3	3	1
				= 48	= 32

Av TAT = $\frac{48}{5}$ = 9.6

Av. WT = $\frac{32}{5}$ = 6.4

iii) Preemptive SJF

process	AT	BT	CT	TAT	WT
p1	0	10	16	16	6
p2	0	1	1	1	0
p3	0	2	6	6	4
p4	0	1	2	2	1
p5	0	2	4	4	2
				= 29	= 13

Av TAT = $\frac{29}{5}$ = 5.8

Av WT = $\frac{13}{5}$ = 2.6

iv) FCFS

process	AT	BT	CT	TAT	WT
P1	0	10	15	15	5
p2	0	1	1	1	0
p3	0	2	5	5	3
p4	0	1	16	16	15
p5	0	2	3	3	1
				= 40	= 24

Av.TAT = $\frac{40}{5}$ = 8

Av.WT = $\frac{24}{5}$ = 4.8

Q No. 3

Critical section is that part of the program where the shared memory is accessed in an atomic manner. Only one process can execute its critical section at a time and all other processes have to wait to execute in their critical section.

Executing the critical section must be mutually exclusive because it avoids other processes to get into access resources when another process is utilizing them. Thus, there will be no race condition with mutual exclusion.

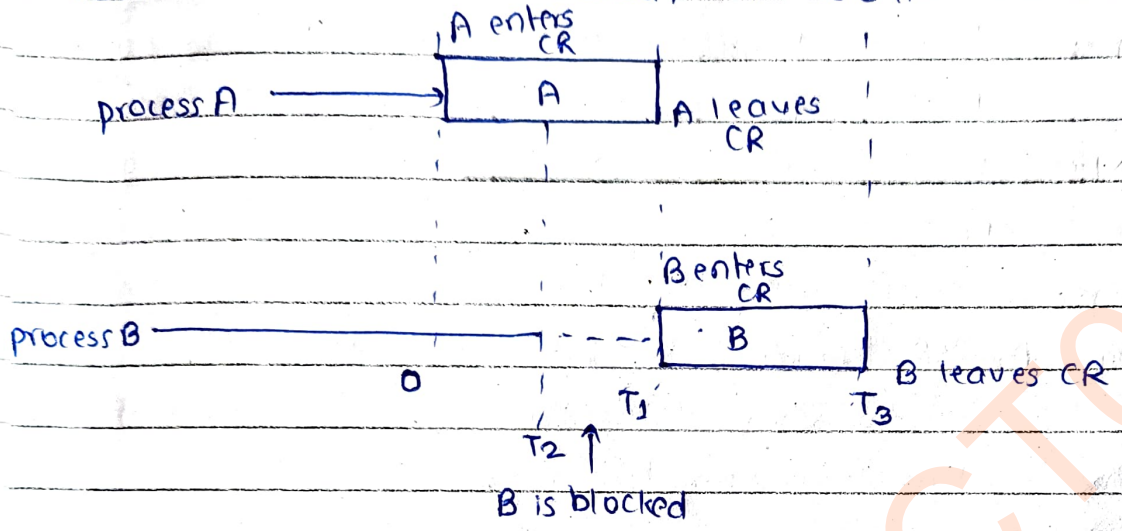


fig: Demonstration of mutual exclusion

Race condition can occur in producer-consumer problem when multiple producers and consumers access a shared buffer simultaneously without proper synchronization, leading to data loss or corruption. For example, multiple producers may attempt to insert data into the same slot, or multiple consumers may attempt to remove data from the same slot in the buffer.

To solve it with semaphores, 3 semaphores can be used.

a) full: for counting no. of slots that are full
b) empty: for counting number of slots that are empty
c) mutex: to make sure the producer and consumer do not access the buffer at the same time.

The following is the code to solve producer-consumer problem using semaphore:

```
#define N 100
typedef int semaphore;
semaphore mutex = 1;
semaphore empty = N;
semaphore full = 0;

void producer(void)
{
    int item;
    while (TRUE) {
        item = produce_item();
        down (&empty); /* decreases empty count */
        down (&mutex); /* enters critical region */
        insert_item (item);
        up (&mutex); /* leaves critical region */
        up (&full); /* increases full slot count */
    }
}
void consumer(void)
{
    int item;
    while (TRUE) {
        down (&full); /* decreases full slot count */
        down (&mutex); /* increases critical region */
        item = remove_item ();
```

```
        up(&mutex); /* leaves critical region */
        up(&empty); /* increases empty slot count */
        consume_item(item);
    }
}
```

Q No4.

Page fault is an error that occurs when a process tries to access a page that is not currently present in the physical memory, so the operating system needs to load it from secondary storage.

Demand paging is a virtual memory management technique where pages are loaded into memory only when needed, as opposed to loading the entire process into memory at once.

Given:

logical address space = 8 pages of 1024 words each
Physical memory = 32 frames

Page size = $1024 \times 2 = 2048$ bytes
Assuming, each word is 2 bytes (16-bit addressing)

No. of pages = 8.
ie No. of bits for 8 pages = $\log_2(8) = 3$ bits
No. of bits for 1024 words = $\log_2(1024) = 10$ bits

So, Total no. of bits in logical address = $3 + 10 = 13$ bits

No. of frames = 32

Frame size = 2048

No. of bits to address each frame = $\log_2(32) = 5$ bits

No. of bits to address each byte = $\log_2(2048) = 11$ bits

Total no of bits in physical address = 11+5 = 16 bits.

To perform paging, logical address is divided into a page number and an offset. The page number is used to index in page table that has corresponding frame number in physical memory. The offset is added to the frame number to obtain the physical address of the desired memory location.

Q No 5

File system plays a vital role in the operating system as it permits users to create data, collections called files with properties like long term existence, shareable between processes and having a hierarchial structure.

Files can be implemented by the following ways:

a) Contiguous allocation

The files are stored as a contiguous run of disk blocks. It requires each file to occupy a set of contiguous addresses on a disk. It supports both sequential and direct access methods.

Advantages:

i) It is simple to implement

ii) The read performance is excellent.

Disadvantages:

i. Fragmentation of disk

ii. File size may not be known in advance.

b) Linked list allocation

It stores files as a linked list of disc blocks. The first word of each block is used as a pointer to the next one and the rest of the block is for data.

Advantages:

i) Only address of first block appears in the directory entry.

ii) No disk fragmentation

iii) Every disk block can be used.

Disadvantages

i) Random access is extremely slow.

ii) Loss of one pointer can be problematic overall as the files are linked together with them.

c) I-node:

It associates each file with a data structure called an i-node (index node). I-node is a unique number given to a file in an operating system. Both name and their I node numbers are stored as entries in their directory that appears to ~~be~~ the user.

Advantages:

i) Size of array with i-nodes is less.

ii) It solves external fragmentation problem.

Disadvantages:

i) ~~There can~~ Multi-level index.

Q No 6

Principles of I/O software:

The main concept of I/O software is called as device independence. It stands for the program can be written or matched in such a way that it can be read on hard disk without modifying the program for each different device. I/O software should control the error by using control program (error handling).

Soln

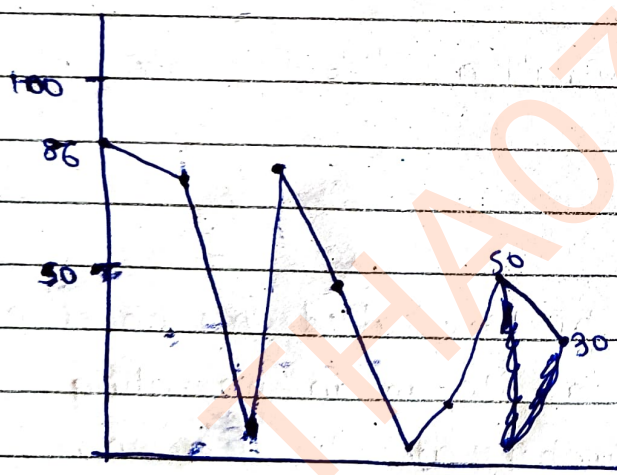
Total cylinders = 100

currently at 43, previously at 25 ie upward movement

The queues

86, 70, 13, 74, 48, 9, 22, 50, 30

a) FCFS

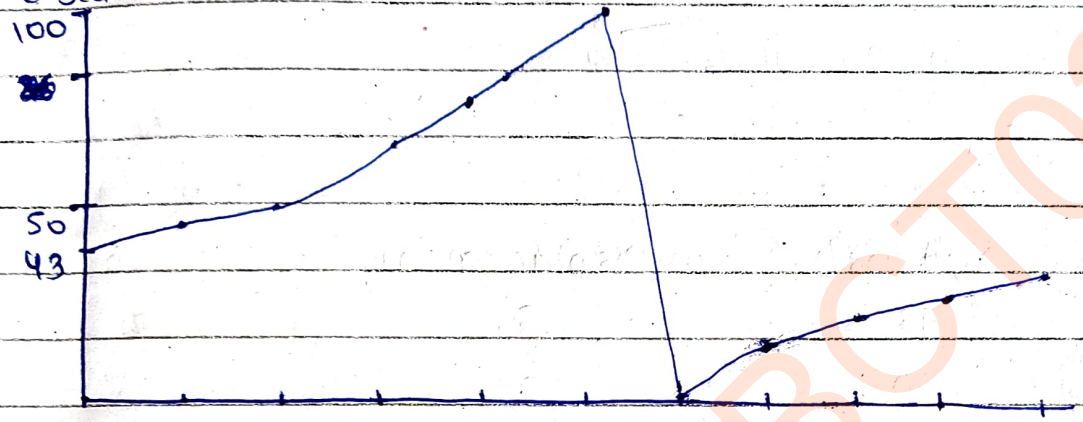


Graph according to FCFS

S.N	Current pos.	Next pos.	Displacement
1	86	70	16
2	70	13	57
3	13	74	61
4	74	48	26

5	48	9	39
6	9	22	13
7	22	50	28
8	50	30	20
	[illegible]		Σ = 260

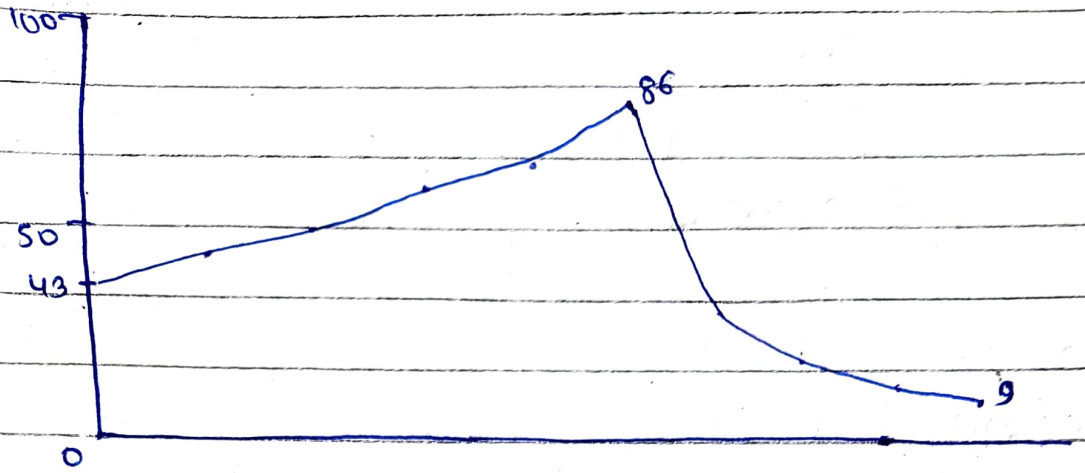
b) C-scan



Graph according to circular-scan

S.N	current pos.	next pos.	Displacement
1	43	48	5
2	48	50	2
3	50	70	20
4	70	74	4
5	74	86	12
6	86	100	14
7	100	0	100
8	0	9	9
9	9	13	4
10	13	22	9
11	22	30	8
			Σ = 187

c) SSTF (Shortest Seek Time First)



Graph according to SSTF

S.N.	current pos	next pos	Displacement
1	43	48	5
2	48	50	2
3	50	70	20
4	70	74	4
5	74	86	12
6	86	80	56
7	80	22	8
8	22	13	9
9	13	9	4
			$\Sigma = 120$

Q No 7

Four conditions of deadlock are given as follows:

a) Mutual exclusion

b) Hold and wait: A process must be holding at least one resou
and waiting for another resource held by another process

c) No preemption: Resources can't be preempted once assigned.

d) Circular wait : There must be a chain of processes such that each member of the chain is waiting for a resource held by next member of the chain.

Soln

Available matrix = [3, 3, 2]

Allocation matrix = $\begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 2 \\ 2 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}$

Maximum matrix = $\begin{bmatrix} 7 & 5 & 3 \\ 3 & 2 & 2 \\ 9 & 0 & 2 \\ 2 & 2 & 2 \\ 4 & 3 & 3 \end{bmatrix}$

Need matrix = Maximum matrix - Allocation matrix

$$= \begin{bmatrix} 7 & 5 & 3 \\ 3 & 2 & 2 \\ 9 & 0 & 2 \\ 2 & 2 & 2 \\ 4 & 3 & 3 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 2 \\ 2 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 4 & 3 \\ 1 & 2 & 2 \\ 6 & 0 & 0 \\ 0 & 1 & 1 \\ 4 & 3 & 1 \end{bmatrix}$$

Initially,

Work = available = [3, 3, 2]

Iter I:

For P0,

Is need <= work ?

ie [7, 4, 3] <= [3, 3, 2] ? No.

So, resource is not given to P0.

Iter II:

For P1, [1, 2, 2] <= [3, 2, 2] ? Yes.

P1 executes and work is updated.

New work = old work + allocation of P1

= [3, 3, 2] + [2, 0, 0]

= [5, 3, 2]

Iter III:

For P2, is [6, 0, 0] <= [5, 3, 2] ? No.

So, resource is not given to P2.

Iter IV:

For P3, is [0, 1, 1] <= [5, 3, 2] ? Yes.

P3 executes.

New work = old work + allocation of P3

= [5, 3, 2] + [2, 1, 1] = [7, 4, 3]

Iter V:

For P4, is [4, 3, 1] <= [7, 4, 3] ? yes.

P4 executes.

new work: old work + allocation of P4

= [7, 4, 3] + [0, 0, 2] = [7, 4, 5]

Iter VI:

For P0,

Is [7, 4, 3] <= [7, 4, 5] ? Yes.

∴ P0 executes.

New work = old work + allocation of P0

= [7, 4, 5] + [0, 1, 0] = [7, 5, 5]

Iter VII:

For P2,

Is [6, 0, 0] <= [7, 5, 5] ? Yes

So, P2 is executed.

Hence, the system is in a safe state because all the process executes.

Sequence: P1, P3, P4, P0, P2,

Q No. 8

Access control list (ACL) is a list of permissions associated with a system resource. It specifies which users or system processes are granted access to objects, as well as what operations are allowed on given objects. Each entry in an ACL specifies a subject and an operation. It can allow one user to access a part of the system and prevents another user from accessing the same part of the system.

It is used to define and enforce permissions on files, directories, and other system resource. The list can be managed using tools including command-line utilities and graphical user interfaces.

The roles of system administrators in change management are as follows:

a) They assess potential impact of any proposed changes on system availability.
b) They develop and implement change policies and procedures.
c) They plan and schedule changes to minimize disruption in system availability.
d) They monitor the system to ensure it is functioning correctly and the change has not caused any unforeseen problems.

Q No 9

a) Public key cryptography

It is a cryptographic method that uses two different keys one for encryption (public key) and one for decryption (private key). The public key is freely available and can be shared with anyone, while the private key is kept secret by the owner. Public key encrypts the message but can be decrypted only by owner of private key. This allows for secure communication and data transfer without the need for both parties to have access to the same key. It is used in secure communication protocols like SSL, TLS, SSH etc.

iii) Process Vs Thread

Process consists of a virtual address space, program code and system resource during execution of a program but thread is a lightweight unit of execution within a process, sharing the same virtual address space and system resources of the parent process. Processes use interprocess communication (IPC) mechanism such as pipes or sockets to communicate with each other but

threads can communicate directly with other threads within the same process. Thus, both process and thread represent units of execution within an OS but differ in their memory management, communication mechanism and resource usage.

2079 Bock
Ashwin

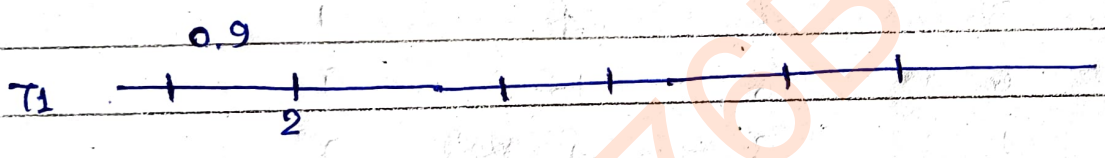
Only numericals

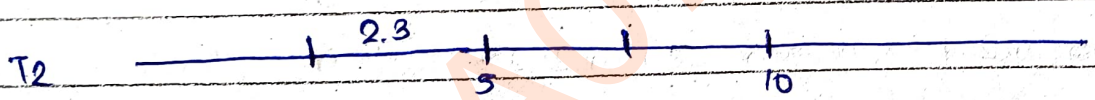
(2).

Task	Execution time	period
T1	0.9	2
T2	2.3	5

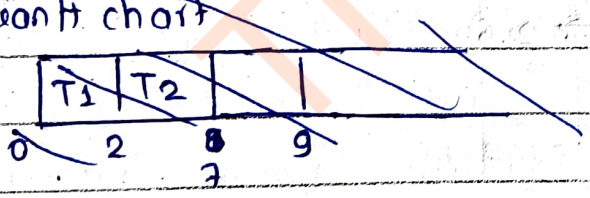
LCM = 10

EDF (Earliest Deadline First)

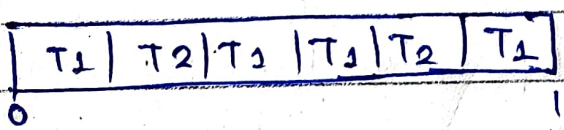




Gantt chart



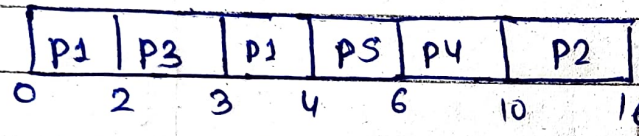
Gantt chart



③.

process	Execution time	Arrival Time
P1	3	0
P2	6	1
P3	1	2
P4	4	3
P5	2	4

SRTF (Shortest Remaining Time First) algorithm



Gantt chart for SRTF algorithm

process	AT	BT	CT	TAT	WT
P1	0	3	4	4	1
P2	1	6	16	15	9
P3	2	1	3	1	0
P4	3	4	10	7	3
P5	4	2	6	2	0
				Σ = 29	Σ = 13

Av TAT = $\frac{29}{5}$ = 5.8 Av WT = $\frac{13}{5}$ = 2.6

Q No: 4

page reference string:

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

page frame = 4

a) LRU

No of frame \ Reference	7	0	1	2	0	3	0	4	2	3	0	3	2
0	7*	7	7	7	7	3*	3	3	3	3	3	3	3
1		0*	0	0	0	0	0	0	0	0	0	0	0
2			1*	1	1	1	1	4*	4	4	4	4	4
3				2*	2	2	2	2	2	2	2	2	2
Res:	F	F	F	F	H	F	H	F	H	H	H	H	H

No. of page fault : 6.

b) FIFO

No of frame \ Ref. page	7	0	1	2	0	3	0	4	2	3	0	3	2
0	7*	7	7	7	7	3*	3	3	3	3	3	3	3
1		0*	0	0	0	0	0	4*	4	4	4	4	4
2			1*	1	1	1	1	1	1	1	0*	0	0
3				2*	2	2	2	2	2	2	2	2	2
	F	F	F	F	H	F	H	F	H	H	F	H	H

No. of page fault : 7

c) optimal Replacement Algorithm

No of frame \ Ref. page	7	0	1	2	0	3	0	4	2	3	0	3	2
0	7*	7*	7*	7*	7	3*	3	3	3	3	3	3	3
1		0*	0	0	0	0	0	0	0	0	0	0	0
2			1*	1	1	1	1	4*	4	4	4	4	4
3				2*	2	2	2	2	2	2	2	2	2
	F	F	F	F	H	H	H	F	H	H	H	H	H

No of page fault : 5

Q No 7

No of cylinders = 150.

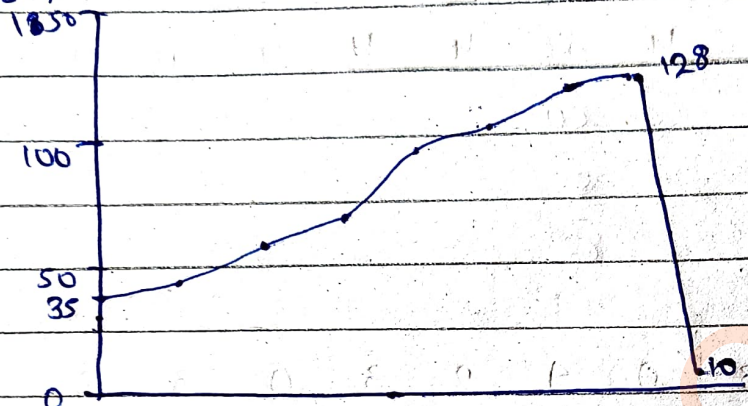
Currently = 35 previous = 120.

ie Direction = downward.

Queue of pending request:

98, 103, 38, 122, 10, 128, 65, 75

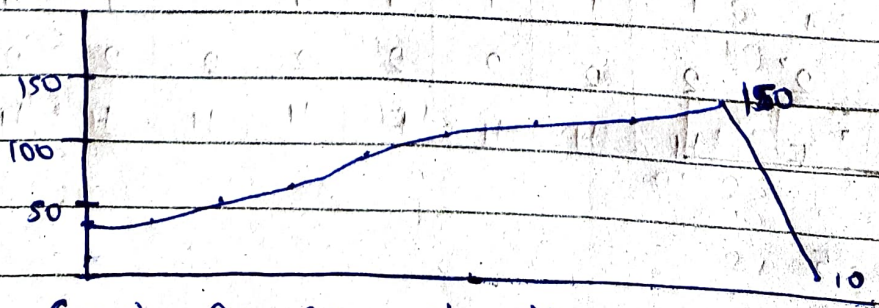
a) SSTF



Graph for SSTF Algorithm

SN	current pos	next pos	Displacement
1	35	38	3
2	38	65	27
3	65	75	10
4	75	98	23
5	98	103	5
6	103	122	19
7	122	128	6
8	128	10	118
			Σ = 211

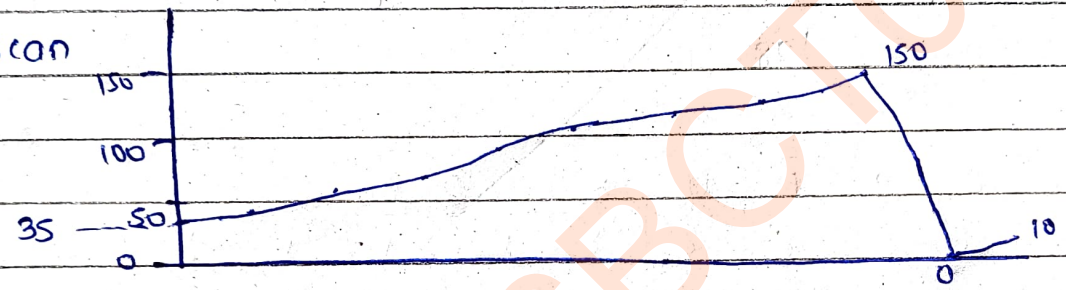
b) SCAN



Graph for scan algorithm

S.N	current pos.	next pos.	Displacement
1	35	38	3
2	38	65	27
3	65	75	10
4	75	98	23
5	98	103	5
6	103	122	19
7	122	128	6
8	128	150	22
9	150	10	140
			Σ = 255

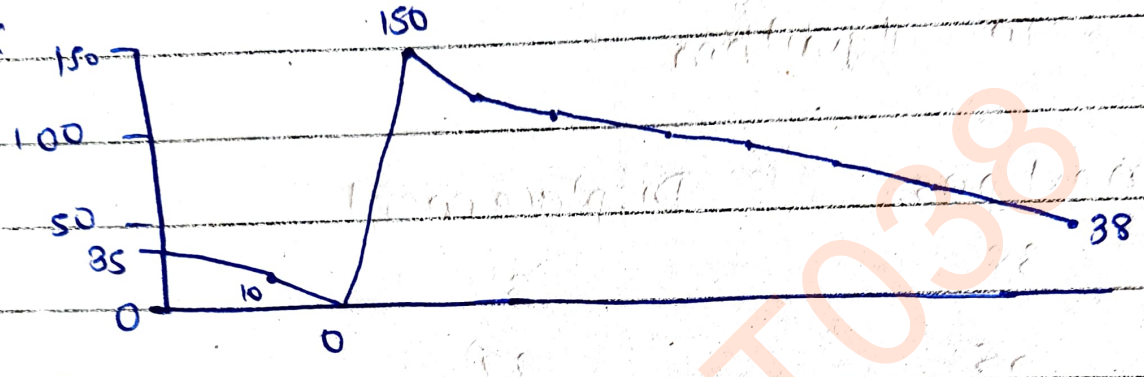
c) C-SCAN



Graph for C-scan algorithm

SN	current pos	next pos	Displacement
1	35	38	3
2	38	65	27
3	65	75	10
4	75	98	23
5	98	103	5
6	103	122	19
7	122	128	6
8	128	150	22
9	150	0	150
10	0	10	10
			Σ = 275

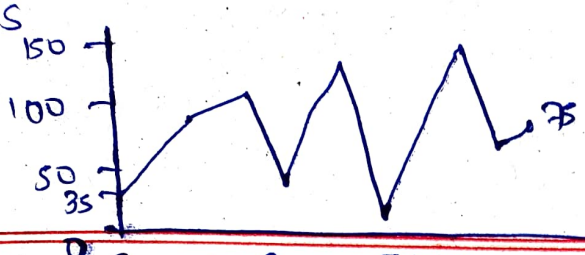
d) C-LOOK



Graph For C-LOOK algorithm

S.N	current pos	next pos.	Displacement
1	35	10	25
2	10	0	10
3	0	150	150
4	150	128	22
5	128	122	6
6	122	103	19
7	103	98	5
8	98	75	23
9	75	65	10
10	65	38	27
			Σ: 297

e) FCFS



Graph for FCFS

S.N	current pos.	next pos.	Displacement
1	35	98	63
2	98	103	5
3	103	38	65
4	38	122	84
5	122	10	112
6	10	128	118
7	128	65	63
8	65	75	10
			$\Sigma = 520$