

DATE  
11 FEB, 2019

Note: Work = Available

\* IF NEED <= AVAILABLE

THEN → WORK = WORK + ALLOCATION

Page No.:	
Date:	

Find safe state if  $P_1 = (1, 0, 2) = \text{extra}$   
Need Matrix.

	ALLOCATION			MAX			AVAILABLE		
	A	B	C	A	B	C	A	B	C
$P_0$	0	1	0	7	5	3	3	3	2
$P_1$	2+1	0	0+2	3	2	2	-1	0	2
$P_2$	3	0	2	9	0	2			
$P_3$	2	1	1	2	2	2			
$P_4$	0	0	2	4	3	2			

0 2 2

NEED MATRIX

$NEED = | \text{Max} - \text{Allocation} |$

	A	B	C
$P_0$	7	4	3
$P_1$	1	2	2
$P_2$	6	0	0
$P_3$	0	1	1
$P_4$	4	3	0

$P_0 = \text{Need} \leq \text{Avail}$   
 $= 743 \not\leq 332$   
 $= \text{False}$

$P_2 = \text{Need} \not\leq \text{Avail}$   
 $= 600 \not\leq 532$   
 $= \text{False}$

$P_4 = \text{Need} \leq \text{Avail}$   
 $= 430 \leq 743$   
 $= \text{True}$

$P_1 = \text{Need} \leq \text{Avail}$   
 $= 122 \leq 332$   
 $= \text{True}$   
 $= 332 + 200$   
 $= 532$

$P_3 = \text{Need} \leq \text{Avail}$   
 $= 011 \leq 532$   
 $= \text{True}$   
 $= 011 + 532$   
 $= 743$

$= 430 + 743$   
 $= 745$

$$P_0 = \text{Need} \leq \text{Avail}$$

$$743 \leq 745$$

True

$$\therefore 010 + 745$$

$$= 755$$

$$P_2 = \text{Need} \leq \text{Avail}$$

$$= 600 \leq 755$$

True

$$= 300 + 755$$

$$= 10 \ 5 \ 7$$

SAFE STATE :  $\langle P_1, P_3, P_4, P_0, P_2 \rangle$

$P_1(1, 0, 2)$  EXTRA.

ALLOCATION

MAX

AVAIL

	A	B	C	A	B	C	A	B	C
$P_0$	0	1	0	7	5	3	3	3	2
$P_1$	2+1	0+0	0+2	3	2	2	-1	0	2
	3	0	2				2	3	0
$P_2$	3	0	2	9	0	2			
$P_3$	2	1	1	2	2	2			
$P_4$	0	0	2	4	3	2			

NEED MATRIX

	A	B	C
$P_0$	07	4	3
$P_1$	0	2	0
$P_2$	6	0	0
$P_3$	0	1	1
$P_4$	4	3	0

FOR  $P_0$ , Need  $\leq$  Avail

$$743 \not\leq 230$$

False.

FOR  $P_1$ , Need  $\leq$  Avail

$$020 \leq 230$$

True.

$$\text{Work} = \text{Work} + \text{Allocation}$$

$$= 230 + 302$$

$$= 532$$

FOR  $P_2$ , Need  $\leq$  Avail

$$A \rightarrow 6 \not\leq 5$$

$$B \rightarrow 0 \not\leq 3$$

$$C \rightarrow 0 \not\leq 2$$

False.

FOR  $P_3$ , Need  $\leq$  Avail

$$A \rightarrow 0 \leq 5$$

$$B \rightarrow 1 \leq 3$$

$$C \rightarrow 1 \leq 2$$

True.

$$A \rightarrow 7$$

$$B \rightarrow 4$$

$$C \rightarrow 3$$

FOR  $P_4$ , Need  $\leq$  Avail

$$A \rightarrow 4 \leq 7$$

$$B \rightarrow 3 \leq 4$$

$$C \rightarrow 0 \leq 3$$

True.

$$\text{Work} = \text{Work} + \text{Allocation}$$

$$002 + 743 = 745$$

For  $P_0$ , Need  $\leq$  Avail

$$A \rightarrow 7 \leq 7$$

$$B \rightarrow 4 \leq 4$$

$$C \rightarrow 3 \leq 5$$

True

$$\text{Work} = \text{Work} + \text{Allocation}$$

$$= 010 + 745$$

$$= 755$$

For  $P_2$ , Need  $\leq$  Avail

$$A \rightarrow 6 \leq 7$$

$$B \rightarrow 0 \leq 5$$

$$C \rightarrow 0 \leq 5$$

True

$$\text{Work} = \text{Work} + \text{Allocation}$$

$$= 302 + 755$$

$$= 1057$$

SAFE STATE :  $\{P_1, P_3, P_4, P_0, P_2\}$

DATE 9<sup>TH</sup> FEB; 2019

	ALLOCATION				MAX				AVAILABLE			
	A	B	C	D	A	B	C	D	A	B	C	D
$P_0$	0	0	1	2	0	0	1	2	1	5	2	0
$P_1$	1	0	0	0	1	7	5	0				
$P_2$	1	3	5	4	2	3	5	6				
$P_3$	0	6	3	2	0	6	5	2				
$P_4$	0	0	1	4	0	6	5	6				

- What is content of Need Matrix?
- Safe State
- $P_1 = 0420 = \text{EXTRA}$





safe state =  $\langle P_0, P_2, P_3, P_4, P_1 \rangle$

$\therefore$  System is in safe state.

• EXTRA  $P_1 = 0 \ 4 \ 2 \ 0$

	ALLOCATION				MAX				AVAILABLE			
	A	B	C	D	A	B	C	D	A	B	C	D
$P_0$	0	0	1	2	0	0	1	2	1	1	0	0
$P_1$	1	4	2	0	1	7	5	0				
$P_2$	1	3	5	4	2	3	5	6				
$P_3$	0	6	3	2	0	6	5	2				
$P_4$	0	0	1	4	0	6	5	6				

Need Matrix = | Max - Allocation |

	A	B	C	D
$P_0$	0	0	0	0
$P_1$	0	3	3	0
$P_2$	1	0	0	2
$P_3$	0	0	2	0
$P_4$	0	6	4	2

Need  $\leq$  Available

i) FOR  $P_0$ ,  $0000 \leq 1100$   
True.

ii) FOR  $P_1$ ,  $0330 \leq 1112$   
False.

Available :

$$A \rightarrow 0 + 1 = 1$$

$$B \rightarrow 0 + 1 = 1$$

$$C \rightarrow 1 + 0 = 1$$

$$D \rightarrow 2 + 0 = 2$$

$\therefore 1112$

iii) For  $P_2$ ,  $1002 \leq 1112$

True.

Available:

$$A \rightarrow 1 + 1 = 2$$

$$B \rightarrow 3 + 1 = 4$$

$$C \rightarrow 5 + 1 = 6$$

$$D \rightarrow 4 + 2 = 6$$

$\therefore 2 \ 4 \ 6 \ 6$

ix) For  $P_3$ ,  $0020 \leq 2466$

True

Available:

$$A \rightarrow 0 + 2 = 2$$

$$B \rightarrow 6 + 4 = 10$$

$$C \rightarrow 3 + 6 = 9$$

$$D \rightarrow 2 + 6 = 8$$

$\therefore 2 \ 10 \ 9 \ 8$

v) For  $P_4$ ,  $0642 \leq 21098$

True.

Available:

$$A \rightarrow 0 + 2 = 2$$

$$B \rightarrow 0 + 10 = 10$$

$$C \rightarrow 1 + 9 = 10$$

$$D \rightarrow 4 + 8 = 12$$

$\therefore 2 \ 10 \ 10 \ 12$

vi) For  $P_1$ ,  $0330 \leq 2101012$

True.

Available:

$$A \rightarrow 1 + 2 = 3$$

$$B \rightarrow 4 + 10 = 14$$

$$C \rightarrow 2 + 10 = 12$$

$$D \rightarrow 0 + 12 = 12$$

$\therefore 3 \ 14 \ 12 \ 12$

CONCLUSION:  $P_1$  can be granted extra.

DATE

11<sup>th</sup> FEB, 2019 \* \*

Explain User level Thread (ULT) & Kernel level Thread (KLT) in detail.

KERNEL THREAD:

- It is the entity that is scheduled by kernel.
- It uses kernel text & global data, but has its own kernel stack as well as a Data Structure (DS) do hold scheduling & synchronization information.

MICRO KERNEL

MONOLITHIC KERNEL

i) Its design is complex.

i) Its implementation is easy.

ii) A bug in Microkernel system does not crash the entire system.

ii) A bug in Monolithic kernel system usually crashes the entire system.

DATE  
13<sup>th</sup> FEB, 2019

PROCESS	ARRIVE TIME	BURST TIME / PROCESSING TIME
A	0	3
B	1	6
C	4	4
D	6	3
		<u>16</u>

\* ALGORITHM: FCFS

Gantt CHART.



WAITING TIME OF PROCESSES

$A = 0$   
 $B = 3 - 1 = 2$   
 $C = 9 - 4 = 5$   
 $D = 13 - 6 = 7$

Average Waiting Time =  $14/4 = 3.5$  ms



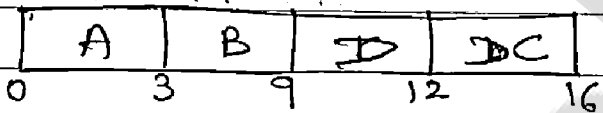
TURN AROUND TIME OF PROCESSES: = W.T. + B.T.

$$A = 0 + 3 = 3$$
$$B = 2 + 6 = 8$$
$$C = 5 + 4 = 9$$
$$D = 7 + 3 = 10$$

$$\text{Average Turn Around Time} = \frac{30}{4} = 7.5 \text{ ms}$$

\* ALGORITHM: SJF (Non-Preemptive)

Gantt CHART:



WAITING TIME OF PROCESSES:

$$A = 0$$
$$B = 3 - 1 = 2$$
$$C = 12 - 4 = 8$$
$$D = 9 - 6 = 3$$

$$\text{Average Waiting Time} = \frac{13}{4} = 3.25 \text{ ms}$$

TURN AROUND TIME OF PROCESSES: W.T. + B.T.

$$A = 0 + 3 = 3$$
$$B = 2 + 6 = 8$$
$$C = 8 + 4 = 12$$
$$D = 3 + 3 = 6$$

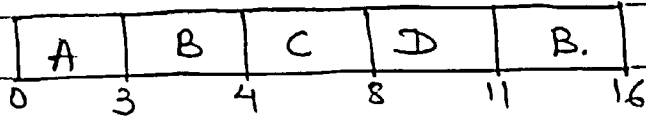
$$\text{Average Turn Around Time} = \frac{29}{4} = 7.25 \text{ ms}$$

4

A	B	C	D	B
3	1	4	3	5

\* ALGORITHM: SJF (Preemptive)

Gantt Chart



WAITING TIME OF PROCESSES:

A = 0

B = 10 - 1 = 9

C = 4 - 4 = 0

D = 8 - 6 = 2

Average Waiting Time =  $\frac{11}{4} = 2.75$  ms

TURN AROUND TIME OF PROCESSES:

A = 0 + 3 = 3

B = 9 + 6 = 15

C = 0 + 4 = 4

D = 2 + 3 = 5

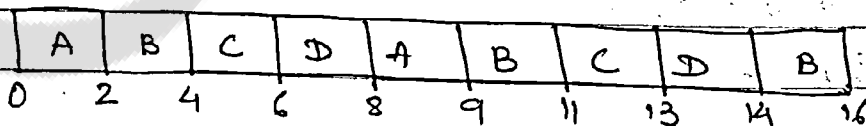
Average Turn Around Time =  $\frac{27}{4} = 6.75$  ms

\* ALGORITHM: ROUND ROBIN

(q = 2)

Pending Time	1	4	2	1	0	2	0	0
	A	B	C	D	A	B	C	D
	0	2	4	6	8	9	11	13
								14

Gantt Chart:



AVERAGE WAITING TIME OF PROCESSES

A = 8 - 2 - 0 = 6

B = 14 - 2 - 2 - 1 = 9

$C = 11 - 2 - 4 = 5$

$D = 13 - 2 - 6 = 5$

Average Waiting Time =  $\frac{25}{4} = 6.25 \text{ ms}$

TURN AROUND Time of PROCESSES:

$A = 6 + 3 = 9$

$B = 9 + 6 = 15$

$C = 5 + 4 = 9$

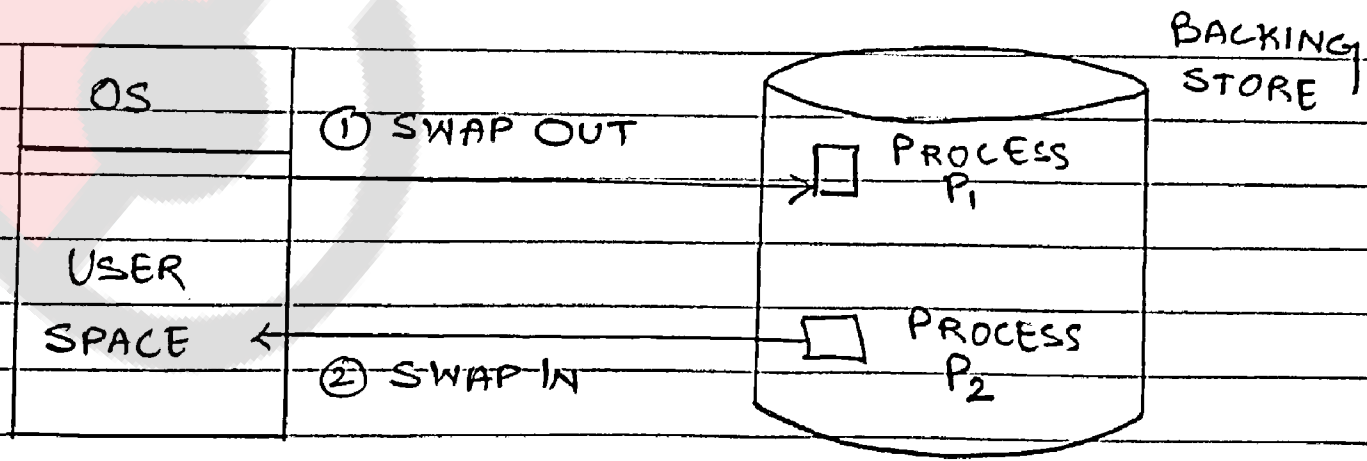
$D = 5 + 3 = 8$

Average Turn Around Time =  $\frac{41}{4} = 10.25 \text{ ms}$

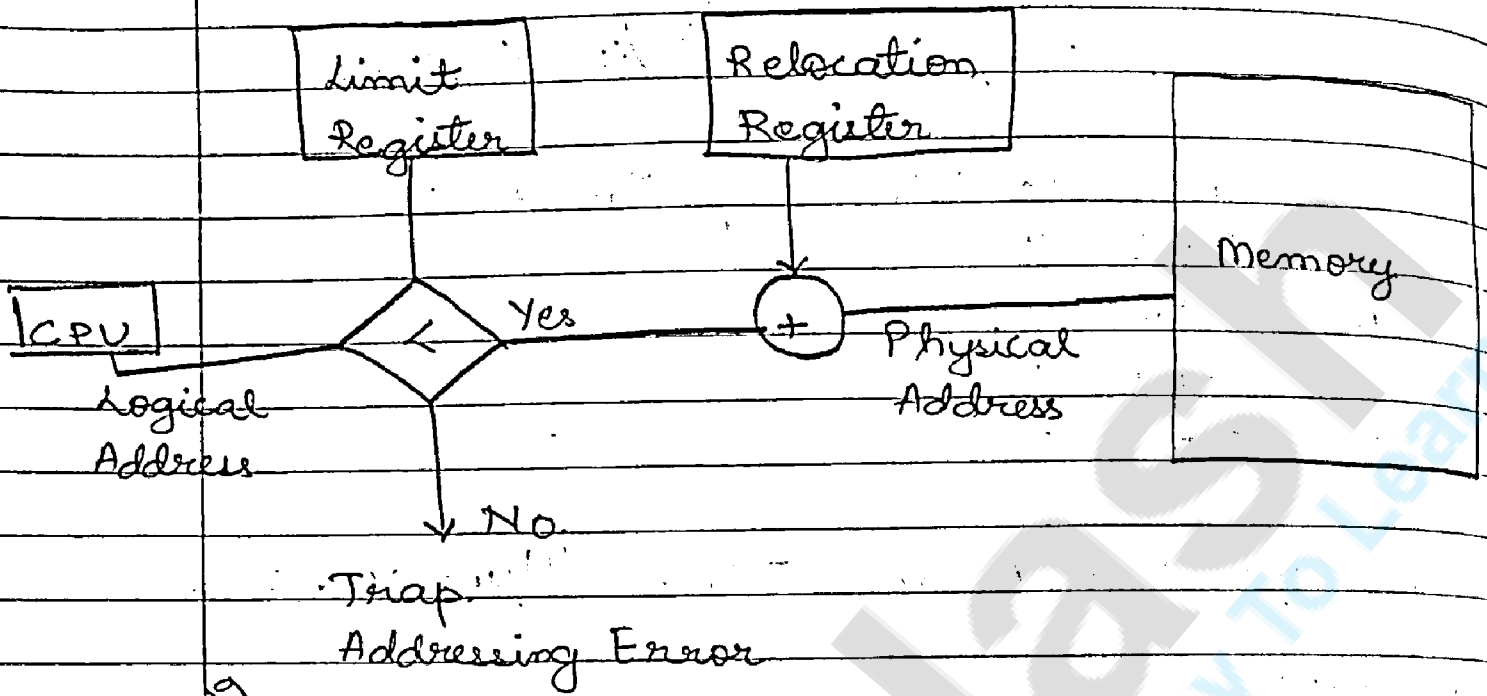
DATE  
23 FEB, 2019

SWAPPING

- A process needs to be in Memory to be executed.
- A process can be swapped out temporarily from the Main Memory to a backing store and then brought back into Main Memory for continued execution.



# Hardware Support for Relocation & Limit Registers



DATE / 25 FEB 2019

## • Multiple Partition Allocation

Most common strategies are:

- First-Fit
- Best-Fit
- Worst-Fit

## Check Virtual Memory

DATE  
27<sup>TH</sup> FEB, 2019

0, 9, 0, 1, 8, 1, 7, 8, 7, 1, 2, 8, 2, 7, 8, 2, 3, 8, 3

(Referencing)

FIFO Replacement.

\*: Page Fault

3 FRAMES.

- : No Page Fault

BEFORE	PAGE	FAULT	AFTER
	0	*	0
0	9	*	9, 0
9, 0	0	-	9, 0
9, 0	1	*	1, 9, 0
1, 9, 0	8	*	8, 1, 9
8, 1, 9	1	-	8, 1, 9
8, 1, 9	7	*	7, 8, 1
7, 8, 1	8	-	7, 8, 1
7, 8, 1	7	-	7, 8, 1
7, 8, 1	1	-	7, 8, 1
7, 8, 1	2	*	2, 7, 8
2, 7, 8	8	-	2, 7, 8
2, 7, 8	2	-	2, 7, 8
2, 7, 8	7	-	2, 7, 8
2, 7, 8	8	-	2, 7, 8
2, 7, 8	2	-	2, 7, 8
2, 7, 8	3	*	3, 2, 7
3, 2, 7	8	*	8, 3, 2
8, 3, 2	3	-	8, 3, 2

TOTAL NO. OF PAGE FAULTS : 8



DATE

6<sup>TH</sup> MARCH, 2019

Page No.	
Date	19/11

Eg. 2, 3, 3, 1, 5, 2, 4, 5, 3, 2, 5, 2, 3.

FIFO Replacement

3 FRAMES.

BEFORE	PAGE	FAULT	AFTER.
, , -	2	*	2, - , -
2, - , -	3	*	2, 2, -
3, 2, -	3	-	3, 2, -
3, 2, -	1	*	1, 3, 2
1, 3, 2	5	*	5, 1, 3
5, 1, 3	2	*	2, 5, 1
2, 5, 1	4	*	4, 2, 5
4, 2, 5	5	-	4, 2, 5
4, 2, 5	3	*	3, 4, 2
3, 4, 2	2	-	3, 4, 2
3, 4, 2	5	*	5, 3, 4
5, 3, 4	2	*	2, 5, 3
2, 5, 3	3	-	2, 5, 3

TOTAL NO. OF PAGE FAULTS: 9.

Eg. 2, 3, 3, 1, 5, 2, 4, 5, 3, 2, 5, 2, 3.

OPTIMAL REPLACEMENT

3 FRAMES.

BEFORE	PAGE	FAULT	AFTER.
, , -	2	*	2, - , -
2, - , -	3	*	3, 2, -
3, 2, -	3	-	3, 2, -
3, 2, -	1	*	1, 3, 2
1, 3, 2	5	*	5, 3, 2
5, 3, 2	2	-	5, 3, 2
5, 3, 2	4	*	5, 3, 4

4, 5, 3			4, 5, 3
5, 3, 4	5	-	5, 3, 4
4, 5, 3			4, 5, 3
5, 3, 4	3	-	5, 3, 4
4, 5, 3			5, 3, 4
5, 3, 4	2	*	5, 3, 4
2, 5, 3, 2	5	-	2, 5, 3, 2
2, 5, 3, 2	2	-	2, 5, 3, 2
2, 5, 3, 2	3	-	2, 5, 3, 2

TOTAL NO. OF PAGE FAULTS: 6

DATE  
5th MARCH, 2019

REFERENCE STRING:

0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7, 8, 2, 3, 8, 3

OPTIMAL REPLACEMENT.

3 FRAMES.

BEFORE	PAGE	FAULT	AFTER
	0	*	0, -, -
0, -, -	9	*	9, 0, -
9, 0, -	0	-	9, 0, -
9, 0, -	1	*	1, 9, 0
1, 9, 0	8	*	8, 1, 9
8, 1, 9	1	-	8, 1, 9
8, 1, 9	8	-	8, 1, 9
8, 1, 9	7	*	7, 8, 1
7, 8, 1	8	-	7, 8, 1
7, 8, 1	7	-	7, 8, 1
7, 8, 1	1	-	7, 8, 1
7, 8, 1	2	*	2, 7, 8
2, 7, 8	8	-	2, 7, 8
2, 7, 8	2	-	2, 7, 8
2, 7, 8	7	-	2, 7, 8
2, 7, 8	8	-	2, 7, 8

2, 7, 8	2	-	2, 7, 8
2, 7, 8	3	*	3, 2, 8
3, 2, 8	5	-	3, 2, 8
3, 2, 8	5	-	3, 2, 8

Total No. Of PAGE FAULTS: 7

Eg REFERENCE STRING:

0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7, 8, 2, 3, 8, 3

LEAST RECENTLY USED.

BEFORE	PAGE	FAULT	AFTER
	0	*	0, ,
0, ,	9	*	9, 0, ,
9, 0, ,	0	-	0, 9, ,
0, 9, ,	1	*	1, 0, 9
1, 0, 9	8	*	8, 1, 0
8, 1, 0	1	-	1, 8, 0
1, 8, 0	8	-	8, 1, 0
8, 1, 0	7	*	7, 8, 1
7, 8, 1	8	-	8, 7, 1
8, 7, 1	7	-	7, 8, 1
7, 8, 1	1	-	1, 7, 8
1, 7, 8	2	*	2, 1, 7
2, 1, 7	8	*	8, 2, 1
8, 2, 1	2	-	2, 8, 1
2, 8, 1	7	*	7, 2, 8
7, 2, 8	8	-	8, 7, 2
8, 7, 2	2	-	2, 8, 7
2, 8, 7	3	*	3, 2, 8
3, 2, 8	8	-	8, 3, 2
8, 3, 2	3	-	3, 8, 2

Total No. Of PAGE FAULT: 9

## DISK MANAGEMENT.

- **SEEK TIME:** It is a time for the disk to move the heads to the cylinder containing the desired sector.
- **Rotational latency** is the additional time waiting for the disk to rotate the desired sector to the disk head.

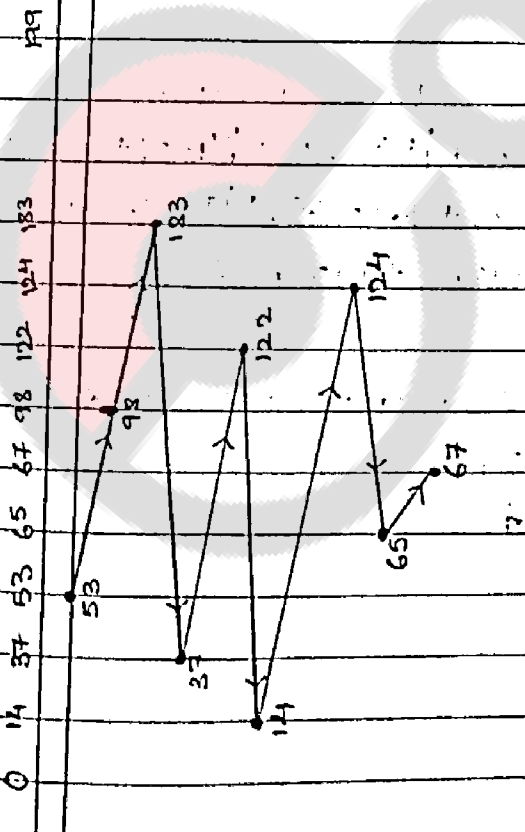
Eg. We illustrate them with a request queue (0-199).  
Head Pointer 53 (98, 183, 37, 122, 14, 124, 65, 67).  
FCFS.

TOTAL HEAD MOVEMENT.

$$= |53 - 98| + |98 - 183| + |183 - 37| + |37 - 122| + |122 - 14| + |14 - 124| + |124 - 65| + |65 - 67|$$

$$= 45 + 85 + 146 + 85 + 108 + 110 + 59 + 2$$

$$= 640$$

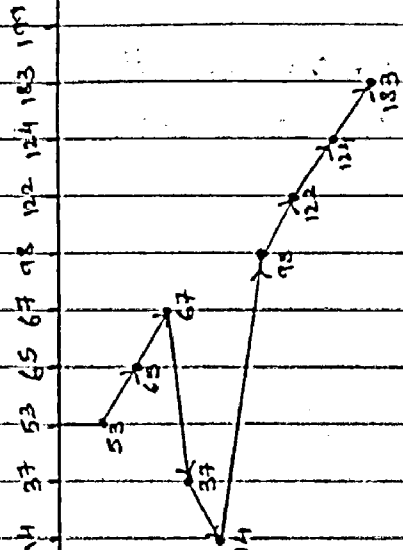


DATE  
11 MARCH, 2019

Squid. 2019

Page No	
Date	11/3/19

SHORTEST SEEK TIME FIRST (SSTF) FOR Avg Head Movement  
 (0-199) 53 - Head Starting  
 (98, 183, 37, 122, 14, 124, 65, 67) 9 (14-183) count



TOTAL HEAD MOVEMENT

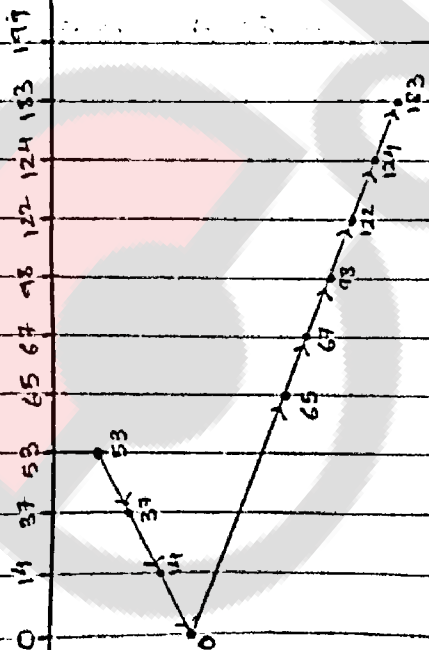
$$= |53-65| + |65-67| + |67-37| + |37-14| + |14-98| + |98-122| + |122-124| + |124-183|$$

$$= 12 + 2 + 30 + 23 + 84 + 24 + 2 + 59$$

$$= 236$$

SCAN

(0-199) - 53 - Head Starting  
 (98, 183, 37, 122, 14, 124, 65, 67)



TOTAL HEAD MOVEMENT

$$= |53-37| + |37-14| + |14-0| + |0-65| + |65-67| + |67-98| + |98-122| + |122-124| + |124-183|$$

$$= 16 + 23 + 14 + 65 + 2 + 31 + 24 + 2 + 59$$

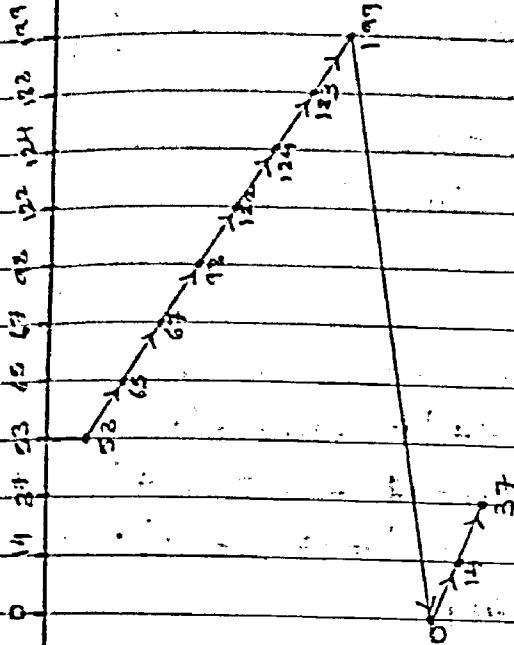
$$= 236$$



DATE  
12<sup>TH</sup> MARCH, 2019

C SCAN.

(0-199) 53-Head Starting  
(98, 183, 37, 122, 14, 124, 65, 67)



TOTAL HEAD MOVEMENT.

$$= |53-65| + |65-67| + |67-98| + |98-122| + |122-124| + |124-183| + |183-199| + |0-14| + |14-37| + |199-0|$$

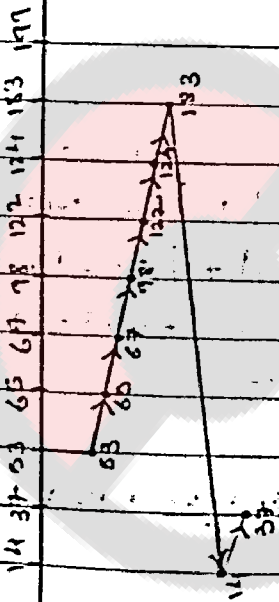
$$= 12 + 2 + 31 + 24 + 2 + 59 + 16 + 14 + 23 + 199$$

$$= 183 + 199$$

$$= 382$$

C LOOK.

(0-199) 53-Head Starting  
(98, 183, 37, 122, 14, 124, 65, 67)



TOTAL HEAD MOVEMENT

$$= |53-65| + |65-67| + |67-98| + |98-122| + |122-124| + |124-183| + |183-0| + |0-14| + |14-37|$$

$$= 12 + 2 + 31 + 24 + 2 + 59 + 169 + 23$$

$$= 322$$

AVERAGE HEAD MOVEMENT

$$= \frac{322}{9}$$

9

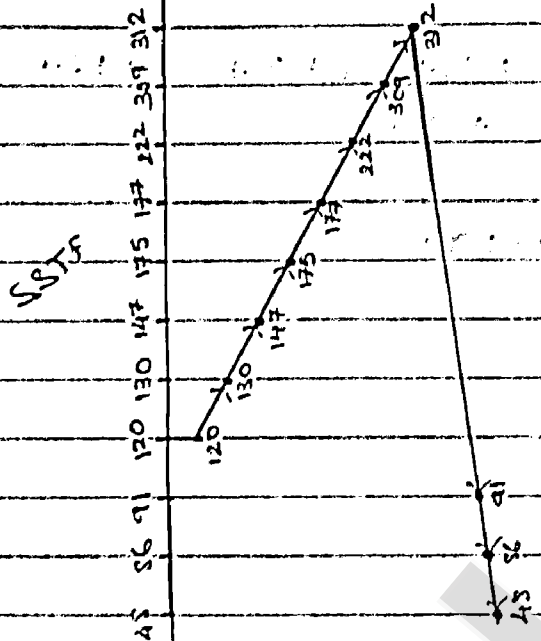
$$= 35.77$$

120  
18, 86, 91, 120, 147, 175, 177, 222, 309, 312

Eg Suppose a disk drive has 400 cylinders, numbered 0 to 399. The drive is currently serving a request at cylinder 120 and previous request was at cylinder 140. The queue of pending request in FIFO order is:

86, 147, 312, 91, 177, 418, 309, 222, 175, 130

SOLVE: i) SSTF ii) SCAN iii) C-SCAN.

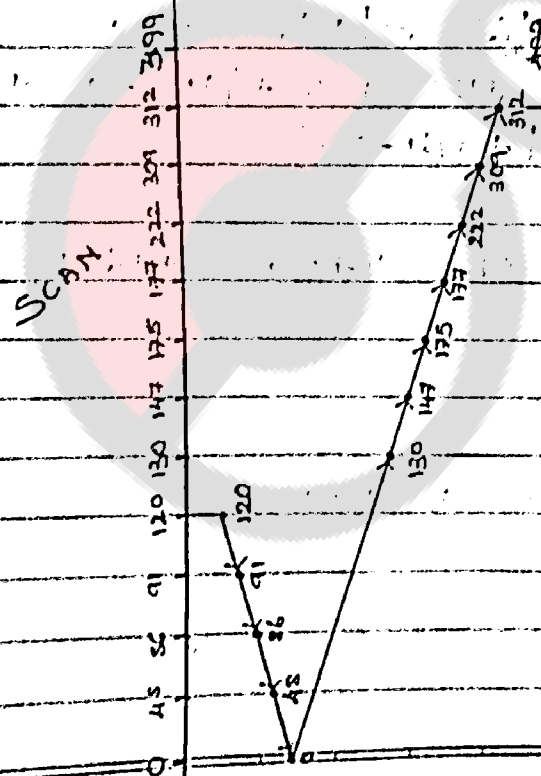


TOTAL HEAD MOVEMENT

$$= |120-130| + |130-147| + |147-175| + |175-177| + |177-222| + |222-309| + |309-312| + |312-91| + |91-86| + |86-48|$$

$$= 10 + 17 + 28 + 2 + 45 + 87 + 3 + 221 + 5 + 38$$

$$= 456$$

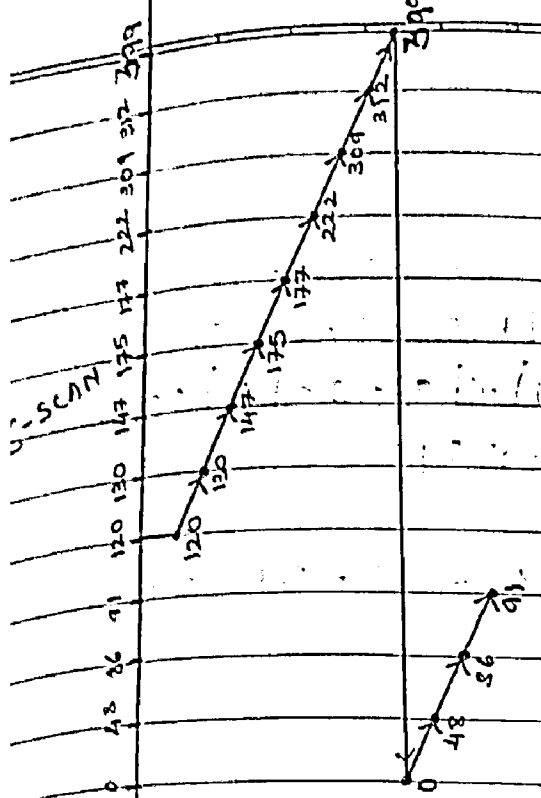


TOTAL HEAD MOVEMENT

$$= |120-91| + |91-86| + |86-130| + |130-147| + |147-175| + |175-177| + |177-222| + |222-309| + |309-312|$$

$$= 29 + 5 + 38 + 48 + 130 + 17 + 22 + 2 + 45 + 87 + 3$$

$$= 432$$



TOTAL HEAD MOVEMENT

$$= |120 - 130| + |130 - 147| + |147 - 175|$$

$$+ |175 - 177| + |177 - 222| + |222 - 309| +$$

$$|309 - 312| + |312 - 399| + |0 - 48| +$$

$$|48 - 86| + |86 - 91| + |399 - 0|$$

$$= 10 + 17 + 28 + 2 + 45 + 87 + 3$$

$$+ 87 + 48 + 38 + 5 + 399$$

$$= 370 + 399$$

$$= 769$$

DATE  
10 MARCH, 2019

Eg

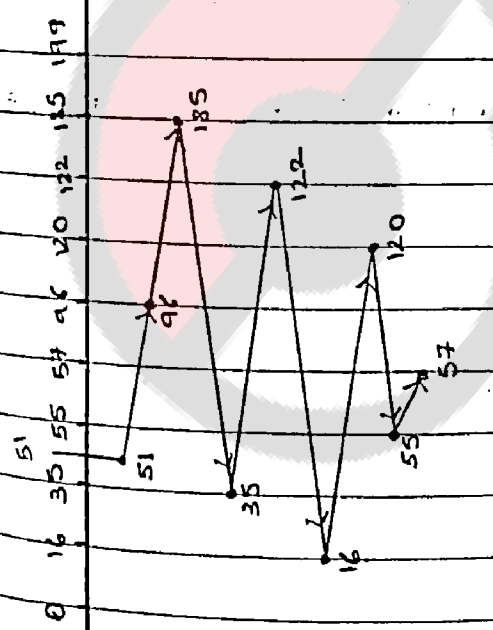
96, 185, 35, 122, 16, 120, 55, 57

Queue Head Start = 51

200 cycle (0-199)

i) Algorithm: FCFS

16, 35, 55, 57, 96, 120, 122, 185



TOTAL HEAD MOVEMENT

$$= |51 - 96| + |96 - 185| + |185 - 35| +$$

$$|35 - 122| + |122 - 16| + |16 - 120| +$$

$$|120 - 55| + |55 - 57|$$

$$= 45 + 89 + 150 + 87 + 106 +$$

$$104 + 65 + 2$$

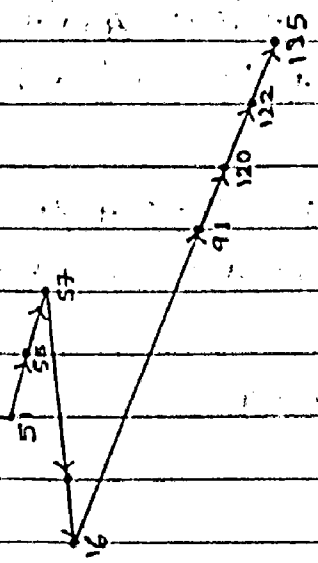
$$= 648$$

ii) ALGORITHM: SSTF

16, 35, 55, 57, 96, 120, 122, 185

Head start: 51

0 16 35 51 55 57 96 120 122 185 199



TOTAL HEAD MOVEMENT  
 $= |51 - 55| + |55 - 57| + |57 - 16| + |16 - 96| + |96 - 120| + |120 - 122| + |122 - 185|$

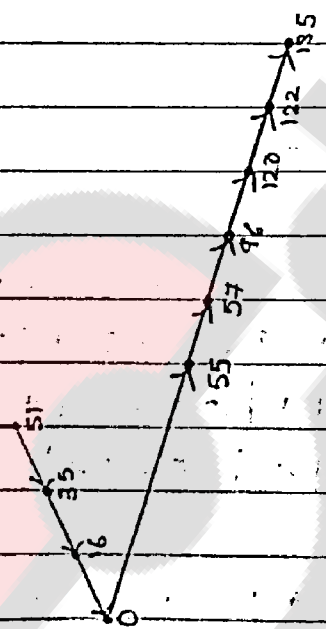
$= 4 + 2 + 41 + 75 + 29 + 2 + 63$   
 $= 216$

iii) SCAN

16, 35, 51, 55, 57, 96, 120, 122, 185

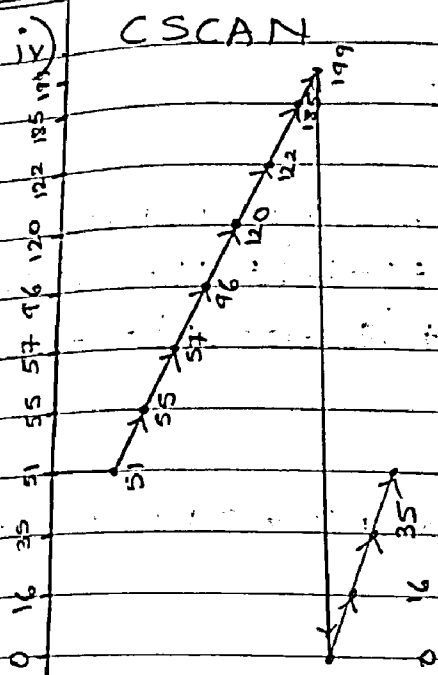
Head start: 51

0 16 35 51 55 57 96 120 122 185 199



TOTAL HEAD MOVEMENT  
 $= |51 - 35| + |35 - 16| + |16 - 0| + |0 - 55| + |55 - 57| + |57 - 96| + |96 - 120| + |120 - 122| + |122 - 185|$

$= 16 + 19 + 16 + 55 + 2 + 39 + 24 + 2 + 63$   
 $= 236$

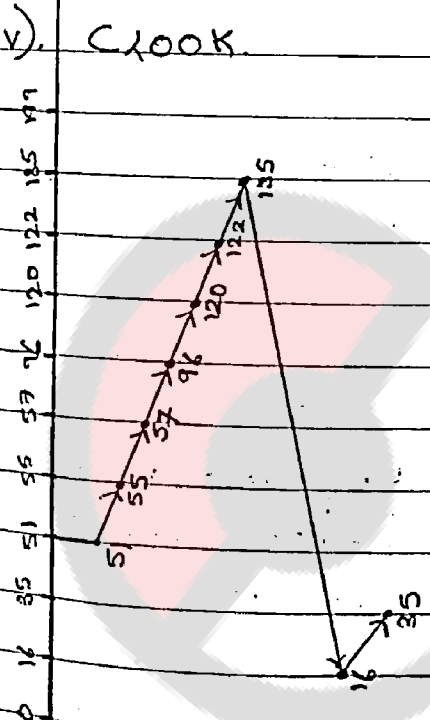


**TOTAL HEAD MOVEMENT.**

$$= |51-55| + |55-57| + |57-96| + |96-120| + |120-122| + |122-185| + |185-199| + |199-35| + |35-16| + |16-0|$$

$$= 4 + 2 + 39 + 24 + 2 + 63 + 14 + 16 + 19 + 16 + 199$$

$$= 392$$



**TOTAL HEAD MOVEMENT.**

$$= |51-55| + |57-96| + |55-57| + |96-120| + |120-122| + |122-185| + |185-16| + |16-35|$$

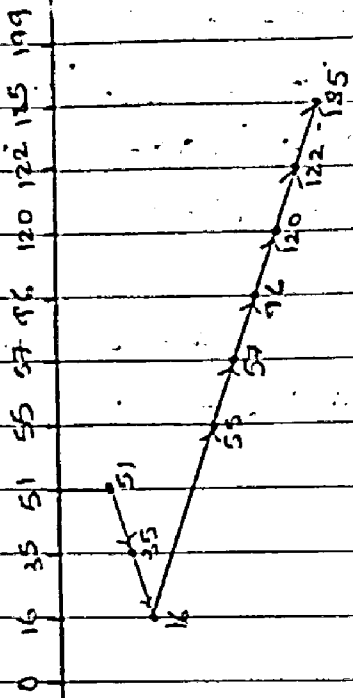
$$= 4 + 39 + 2 + 24 + 2 + 63 + 169 + 19$$

$$= 322$$



vi) Look.

16, 35, 51, 55, 57, 96, 120, 122, 135



TOTAL HEAD MOVEMENT.

$$= |51 - 35| + |35 - 16| + |16 - 55| + |55 - 57| \\ + |57 - 96| + |96 - 120| + |120 - 122| + \\ |122 - 135|.$$

$$= 16 + 19 + 39 + 2 + 39 + 24 + 2 + 13$$

$$= 204$$

DATE

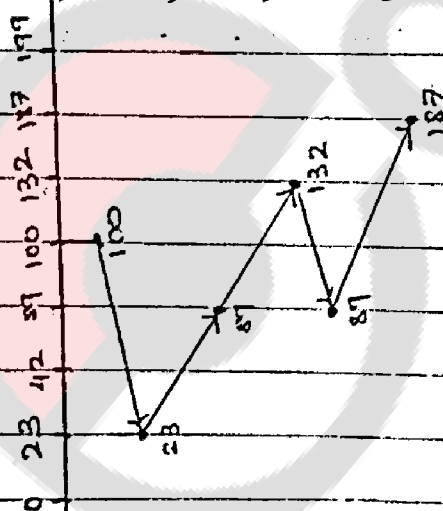
23 MARCH, 2019

Eg. WORK: QUEUE: 23, 89, 132, 42, 187

200 Cylinders (0-199)

100 Disk Head Start.

23, 42, 89, 100, 132, 187



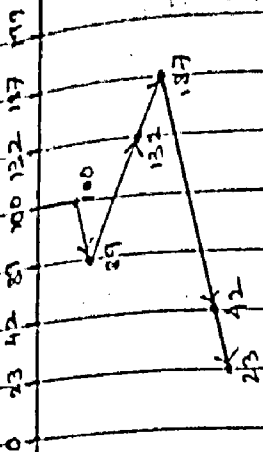
TOTAL HEAD MOVEMENT.

$$= |100 - 23| + |23 - 89| + |89 - 132| \\ + |132 - 42| + |42 - 187|$$

$$= 77 + 66 + 43 + 90 + 145$$

$$= 421$$

## ALGORITHM: SSTF



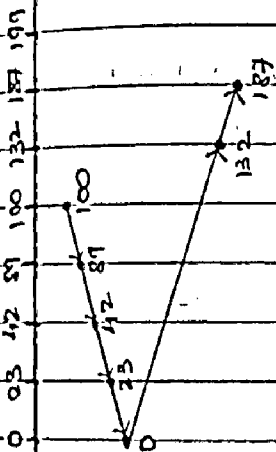
TOTAL HEAD MOVEMENT

$$= |100 - 89| + |89 - 42| + |42 - 23| + |23 - 132| + |132 - 187|$$

$$= 11 + 47 + 19 + 109 + 55$$

$$= 273$$

## ALGORITHM: SCAN



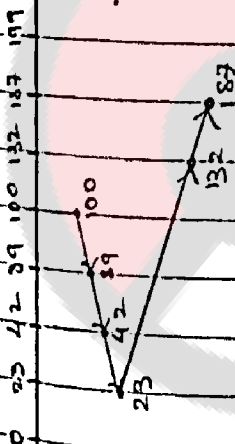
TOTAL HEAD MOVEMENT

$$= |100 - 89| + |89 - 42| + |42 - 23| + |23 - 0| + |0 - 132| + |132 - 187|$$

$$= 11 + 47 + 19 + 23 + 132 + 55$$

$$= 287$$

## ALGORITHM: LOOK



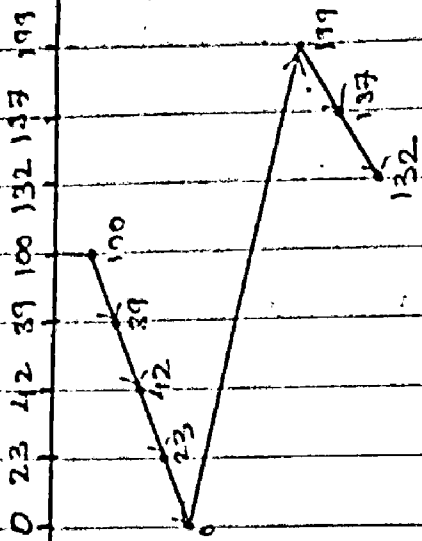
TOTAL HEAD MOVEMENT

$$= |100 - 89| + |89 - 42| + |42 - 23| + |23 - 132| + |132 - 187|$$

$$= 11 + 47 + 19 + 109 + 55$$

$$= 241$$

ALGORITHM: CSAN.



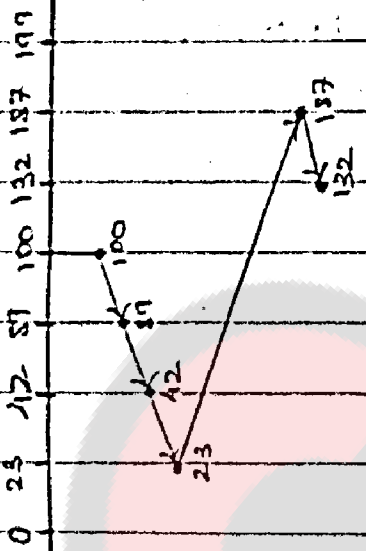
TOTAL HEAD MOVEMENT

$$= |100 - 89| + |89 - 42| + |42 - 23| + |23 - 0| + |0 - 199| + |199 - 127| + |127 - 132|$$

$$= 11 + 47 + 19 + 23 + 199 + 12 + 55$$

$$= 366$$

ALGORITHM: CLOOK.



TOTAL HEAD MOVEMENT

$$= |100 - 89| + |89 - 42| + |42 - 23| + |23 - 187| + |132 - 187|$$

$$= 11 + 47 + 19 + 164 + 55$$

$$= 296$$

DATE  
25<sup>TH</sup> MARCH, 2019

Page No.:	YOUVA
Date:	

## SECURITY :

Security violation of the system can be categorized as accidental intentional (malicious)

- i). Unauthorized reading of data.
- ii). Unauthorized modification of data.
- iii). Unauthorized destruction of data.
- iv). Denial of Service.

## UNIT TEST

- CPU scheduling Problem 

FCFS + SJF (N.P.)
SJF (P.) <sup>OR</sup> + R.R.
- 5 state Model
- Monolithic Kernel / Microkernel
- Linker, loader, Spoofing, Sniffing

## UNIT

- Banker's Algorithm
- Page Faults
- Disk Scheduling

- S.N. • Semaphore, monitor, Internal & External fragmentation
- Thrashing, Access list & Matrix
  - Clock & s/w
  - \* • Android OS, TLB, Seek Time, Demand Paging



## NOTES

SN. IN FINAL Exam.

- Spooling Vs Buffering
- \* Semaphore & Monitoring
- TLB
- Demand Paging

NOTES

VIVA.

1). Deadlock 2). Semaphore 3). PCB 4). 5 state model