CLASS TEST

Sl. : 01_SK_A+B_CS_09042023

MADE EASY
India's Best Institute for IES, GATE & PSUs

Delhi | Bhopal | Hyderabad | Jaipur | Pune | Bhubaneswar | Kolkata

Web: www.madeeasy.in | **E-mail:** info@madeeasy.in | **Ph:** 011-45124612

OPERATING SYSTEM

COMPUTER SCIENCE & IT

Date of Test : 09/04/2023

ANSWER KEY

1.	(d)	7.	(c)	13.	(d)	19.	(b)	25.	(b)
2.	(a)	8.	(b)	14.	(b)	20.	(a)	26.	(d)
3.	(c)	9.	(a)	15.	(d)	21.	(c)	27.	(b)
4.	(c)	10.	(c)	16.	(a)	22.	(b)	28.	(c)
5.	(b)	11.	(d)	17.	(c)	23.	(b)	29.	(c)
6.	(b)	12.	(a)	18.	(b)	24.	(b)	30.	(b)

DETAILED EXPLANATIONS

1. **(d)**

- Present bit → Page is present in RAM
- Dirty bit → Differences in contents of page in RAM and the disk
- Swap out → Moving page from memory to disk
- Swap in → Moving page from disk to memory

2. **(a)**

Let try to serve every request order for every strategy one by one.

I. Best Fit

- 100 K and 25 K request can be served in 125 K memory span.
- 125 K fits in 175K chunk and remaining 50 K used for 50 K request.

All request served in Best Fit.

II. First Fit

- 100 K fits in 125 K chunks and remaining 25 K chunks for next request i.e. 25 K.
- 125 K fits in 175 K chunks and remaining 50 K used for next request i.e. 50 K.

All request served in First Fit.

III. Worst Fit

- 100 K fits in 175 K. Now 75 K chunks is remaining.
- 25 K fits in 125 K. Now look chunks is remaining.
- 125 K can not be served as there is no face chunks but 50 K request can be served

All request are not being served in Worst Fit.

So, I and II successfully served all the requests.

3. **(c)**

	MAX Allocation			Allocated			Current Need		
	A	B	C	A	B	C	A	B	C
P_0	7	5	3	0	1	0	7	4	3
P_1	3	2	2	2	0	0	1	2	2
P_2	9	0	2	3	0	2	6	0	0
P_3	2	2	2	2	1	1	0	1	1
P_4	4	3	3	0	0	2	4	3	1

- After P_1 available resources are (5, 3, 2).
- Now P_3, and P_4 both can be served. After serving both available resources (7, 4, 5).
- Now, similarly P_0 and P_2 an be served.

So possible safe sequence is option (c) which is P_1, P_3, P_4, P_0, P_2.

4. **(c)**

Both statements S_1 and S_2 are correct.

5. **(b)**

As we know, at first level we always get a page directory and each entry in the page directory points to a page table.

So, given 10 bits for page directory.

Then number of page table = 2^{10} = 1024.

Total number of page table = 1 (for first level page table) + 1024 (for second level page table) = 1025.

6. **(b)**

Only tot_sum need to be access exclusively as it shares the same variable.

7. **(c)**

All the options are correct except option (c).

8. **(b)**

Upper bound is RUNNING, READY and BLOCKED states respectively are n, K, K and similarly in lower bound are 0, 0, 0.

9. **(a)**

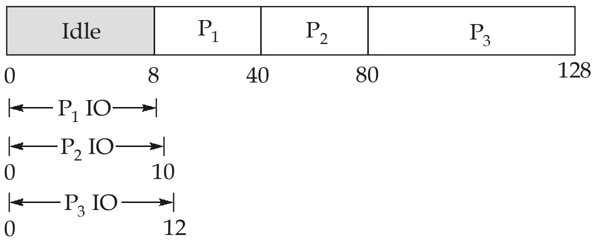
Process P:	Process Q:
W: P(S)	Y: P(T)
Print '5';	Print '6';
Print '5';	Print '6';
X : V(T)	Z : V(S)

If initially S = 0 and T = 1, then we will get the designed result as output. Because initially process Q would be executed then process P.

Hence option (a) is correct.

10. **(c)**

Process	Burst time	I/O time
P_1	32	8
P_2	40	10
P_3	48	12



11. **(d)**

The above codes provides mutual exclusion, progress but not bounded waiting. Let's see how.

- No two process can be in the critical section at any point of time. Hence it provides mutual exclusion.
- The same process can again enter in the critical section after just exiting from critical section. Hence it provides progress.
- Assume process P_1 is in critical section and P_2 is waiting to enter in the critical section. Once P_1 is finished in critical section and suddenly a new process P_3 executed "Swap (Lock, Key)"; first than P_2 and gets into critical section. Hence it violates bounded waiting property. So, code does not provide bounded waiting.

13. **(d)**

P_1 executed first then P_2

$$a = 0 + 15 = 15$$
$$a = 20 + 20 = 40$$

P_2 executed first then P_1

$$a = 20 + 20 = 40$$
$$a = 40 + 15 = 55$$

Total 2 different values that a can have i.e. 40 and 55.

14. **(b)**

Necessary conditions to occur the deadlock.

1. Mutual exclusion
2. Hold and wait
3. No preemption
4. Circular wait

∴ No "Hold and wait" is not a necessary condition to occur the deadlock.

15. (d)

- S_1 is incorrect as OS automatically loads pages from disk when it is needed.
- S_2 is incorrect as dynamic loading follows efficient memory utilization.
- S_3 is incorrect, i.e. when pages are shared between 2 or more process then it can be swapped out from memory to disk using demand paging to swap in new pages when memory is full.

16. (a)

FIFO replacement

3	2	1	5	6	3	2	7	1	3	6	7	3	2	6
			5	5	5	5	7	7		7		7	2	
		1	1	1	1	2	2	2		2		3	3	
	2	2	2	2	3	3	3	3		6		6	6	
3	3	3	3	6	6	6	6	1		1		1	1	

$x = 12$

LRU replacement

3	2	1	5	6	3	2	7	1	3	6	7	3	2	6
			5	5	5	5	7	7		7			7	
		1	1	1	1	2	2	2		6			6	
	2	2	2	2	3	3	3	3		3			3	
3	3	3	3	6	6	6	6	1		1			2	

$y = 11$

Optimal Replacement

3	2	1	5	6	3	2	7	1	3	6	7	3	2	6
			5	6			6						6	
		1	1	1			1						2	
	2	2	2	2			7						7	
3	3	3	3	3			3						3	

$z = 7$

So, $z < . y < x$

17. (c)

Both the statements S_1 and S_2 are correct.

18. (b)

$$\text{TLB hit ratio} = \frac{130}{400} = 0.325$$

$$\text{E.M.A.T.} = P \times (t + M) + (1 - P)(t + 3M) = 0.325 \times (20 + 80) + 0.675\,(20 + 3 + 80)$$

$$= 32.5 \text{ ns} + 0.67 \times 260 \text{ ns} = 32.5 \text{ ns} + 175.5 \text{ ns} = 208 \text{ ns}$$

19. (b)

Let page fault rate be P.

$$\text{EMAT} \geq P * [0.72 * 15 + 0.28 * 8] + (1 - P) * 5 \text{ ms}$$

$$7 \geq P * 13.04 + (1 - P) * 5$$

$$7 \geq 8.04 P + 5$$

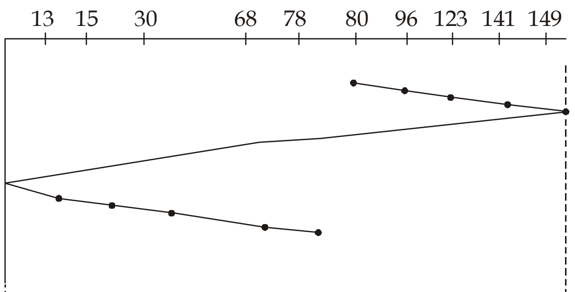
$$8.04 P \leq 2$$

$$P \leq 0.2487 * 100$$

$$P \leq 24.87\%$$

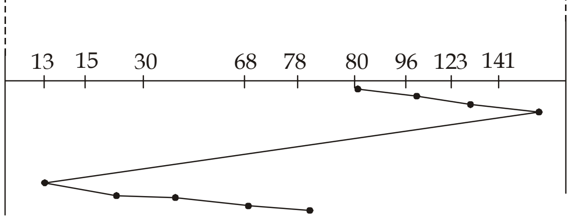
20. (a)

CSCAN:



Total distance traversed by R/W head = (149 – 80) + (149 – 0) + (78 – 0)
= 69 + 149 + 78 = 296

CLOOK:



Total distance traversed by R/w head

$$= (149 - 80) + (149 - 0) + (78 - 0) = 69 + 149 + 78 = 296$$

Figure

Total distance traversed by R/w head

$$= (141 - 80) + (141 - 13) + (78 - 13) = 61 + 128 + 65 = 254$$

Absolute difference $= 296 - 254 = 42$

21. (c)

OS Printed $\rightarrow 2^{n-1} - 1$

Gate Printed $\rightarrow 2^{n-1}$

22. (b)

1. A( )

P(mutex) $\Rightarrow$ mutex = 0
P(Q) $\Rightarrow$ Q = 0

2. B( )

P(Q) $\Rightarrow$ Process B( ) goes in sleep mode

3. A( )

P(R) $\Rightarrow$ R = 0
V(Q) $\Rightarrow$ Q = 1 $\Rightarrow$ Process B( ) awake

4. B( )

P(Q) $\Rightarrow$ Q = 0
P(R) $\Rightarrow$ Process B( ) goes in sleep mode

5. A( )

V(mutex) $\Rightarrow$ mutex = 1
P(Q) $\Rightarrow$ Process A( ) goes in sleep mode

Now both process A( ) and B( ) are in sleep mode and waiting for each another to execute. Hence deadlock occurs. Since there is deadlock surely there will be starvation.
So, option (b) is correct.

24. **(b)**

Statement I and II is correct.

26. **(d)**

(a) Round robin works on time quantum, after certain period of time every process gets the CPU unit for its completion, hence it's most suitable.

(b) Since OS is multiuser and multiprocessing, hence security is the primary concern so that user processes and Kernel processes can be isolated.
Hence two modes are required.

(c) When CPU temperature is too high, the BIOS initiate an interrupt. OS given top priority to this interrupt.

(d) Address translation table need to be changed when switching context from process A to process B.

27. **(b)**

To enter the critical section, process P_i first sets flag [i] to be true set $S_1 = S_2$, thereby asserting that if the other process wishes to enter the critical section it can do so. If both processes try to enter at the same time. The S_1 will be set S_2 or S_2 +1 at roughly the same time. Only, one of these assignment will last; the other will occur, but will be overwritten immediately.

28. **(c)**

1. CPU senses interrupt request line after every instruction.
2. Nearest cylinder next disk scheduling strategy gives the best throughput but the only problem is it can lead to starvation.
3. Using large file block size in a fixed block size file system leads to better disk throughput but poor disk space utilization.

29. **(c)**

1. As files are allocated and deleted, the free disk space is broken into little pieces, hence can lead to external fragmentation.
2. Linked-allocation can be used effectively only for sequential access file. To find the i^{th} block of a file. We must start at the beginning of that file and follow the pointers until we get the i^{th} block.
3. Statement is correct.

30. **(b)**

Calculating the need matrix

	A	B	C	D	E
P_0	0	1	0	0	2
P_1	0	2	1	0	0
P_2	1	0	3	0	0
P_3	0	0	1	1	1

Since, available = 00123, hence only P_3 can be satisfied.
Remaining = (00123) – (00111) = (00012) + (11221) = (11233)
Now P_0 can be executed,
Remaining = (11233) – (01002) = (10231) + (11213) = (21444)
Now P_2 can be executed,
Remaining = (21444) – (10300) = (11144) + (21310) = (32454)
Now P_1 can be executed.

■■■■