MADE EASY
Leading Institute for IES, GATE & PSUs
Delhi | Bhopal | Hyderabad | Jaipur | Pune
Web: www.madeeasy.in | E-mail: info@madeeasy.in | Ph: 011-45124612

Operating System

COMPUTER SCIENCE & IT

Date of Test : 04/04/2026

ANSWER KEY ➤

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

DETAILED EXPLANATIONS

1. (a)

Long term scheduler create a new process and loads into memory by making a transition from new to ready.

Long term schedule controls the degree of multiprogramming.

So option (a) is correct.

2. (c)

Access time: total time needed to access the data

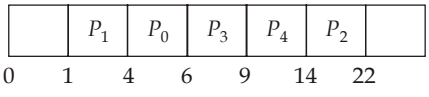
Access time = seek time + rotational latency + data transfer time

Seek time: Time taken to move the head to the correct cylinder that contains desired sector.

Rotational latency: Time taken to move the head to the desired sector within the cylinder.

Data transfer time : Time taken to transfer the actual data.

3. (b)



Gantt chart

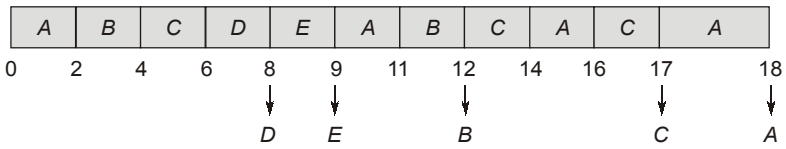
Processes	Waiting Time
P_0	1
P_1	0
P_2	9
P_3	4
P_4	1

$$\text{Average waiting time} = \frac{\sum_{i=0}^{n} \text{Waiting time of } P_i}{\text{Total process}}$$

$$= \frac{1+0+9+4+1}{5} = 3$$

4. (c)

The gantt chart will be,



Calculating TAT will be,

$A \Rightarrow 18 - 0 = 18$

$B \Rightarrow 12 - 0 = 12$

$C \Rightarrow 17 - 0 = 17$

$D \Rightarrow 8 - 0 = 8$

$$E \Rightarrow 9 - 0 = 9$$

$$\text{Average TAT} = \frac{18+12+17+8+9}{5} = 12.8$$

5. **(c)**

S_1: Aging is used to solve starvation problem.

S_2: A thread running in user mode in critical section may get context switched.

Both S_1 and S_2 is true.

6. **(a)**

Race condition is occur if more than one process entering into critical section at the same time. If mutual exclusion is satisfies race condition will not occur, if only one process is trying to update the shared variable race condition is not occur, it if possible if more than one process trying to update at same time.

7. **(c)**

The main reason to have TLB hardware is to speedup the address translation by checking 1st in TLB if there is a miss in TLB then only there is a check in main m/m or cache m/m.

In deadlock avoidance scheme i.e. Banker's algorithm knowledge of resources is required a priori.

8. **(b)**

Page table entry contain bit for representing frame and other information.

$$\text{Number of frames} = \frac{2^{36}}{2^{14}} = 2^{22}$$

22 + other information bit = 38

= 16

9. **(d)**

I. There are four condition required for deadlock, mutual exclusion circular wait, no preemption, hold and wait. If only circular wait then it will not always describe deadlock.

II. Using large file block size in a fixed block file system leads to better disk throughput but poor disk space utilization.

10. **(a)**

- S_1 is correct, if the file is very-very large indexed allocation will fail to accommodate then we will use unix inode.
- S_2 is false because metadata does not include the actual data or contents of files, it only includes the file system structure.

11. **(d)**

There are these fork calls. So, $2^n - 1$ child process will be created i.e. 7.

And we have to include parent process also, so total 8 processes are there.

12. **(d)**

S_1 : Context switching time in uses level threads will be less compared to Kernel level threads.

S_2 : Threads can share the code segment.

S_3 : Kernel level threads are best suitable for I/O bound processes.

13. (a)

Disk size = 48 GB, Block size = 4 KB

$$\text{Number of blocks} = \frac{48\text{ GB}}{4\text{ KB}} = \frac{48\times 2^{30}}{4\times 2^{10}} = 12 \times 2^{20}$$

Now, to keep record of a block of 4 KB is free as not 1 bit is needed.
So, for 12×2^{20} i.e. 12 M, 12 Mbits are needed.

So, in bytes $\frac{12\times 2^{20}}{8} \Rightarrow 3 \times 2^{19}$ bytes

$\Rightarrow 3 \times 2^9 \times 2^{10}$ bytes

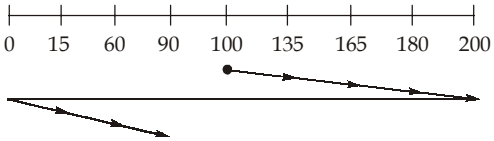
$\Rightarrow 1536$ KB

14. (c)

$$\text{Number of entries in inverted page table} = \frac{\text{Physical address}}{\text{Page size}} = \frac{2^{30}}{2^{13}} = 2^{17}$$

Number of bits = 17

15. (a)



Total seek time = (135 – 100) + (165 – 135) + (180 – 165) + (200 – 180) + 10 + (15 – 0) + (60 – 15) + (90 – 60)
= 35 + 30 + 15 + 20 + 10 + 15 + 45 + 30 = 200 msec

16. (c)

- The process of loading the page into memory on demand (whenever page fault occurs) is known as demand paging.
- Lazy swapper concept is implemented in demand paging in which a page is not swapped into the memory unless it is required. So option (b) is correct.
- Option (c) is false because a process larger than the main memory can be executed because of demand paging. The OS itself loads pages of a process in main memory as required.
- More processes may be maintained in the main memory: Because we are going to load only some of the pages of any particular process, there is room for more processes. This leads to more efficient utilization of the processor because it is more likely that at least one of the more numerous processes will be in the ready state at any particular time.

17. (d)

	X	*Y*	*Z*	*W*
P_0	2	2	2	2
P_1	3	2	0	0
P_2	0	3	2	4
P_3	2	5	0	2
P_4	2	0	0	1

Since available is a 0 0 b, let's suppose a takes value 2 and b takes the value 1.
Available = 2 0 0 1
$P_4 \rightarrow$ Complete $\rightarrow$ Avail = (0000 + 6214) = 6214
$P_1 \rightarrow$ Complete $\rightarrow$ Avail = (6214) – (3200) = (3014) + (3512) = (6526)

$P_0 \rightarrow$ Complete $\rightarrow$ Avail = (6526) – (2222) = (4304) + (3242) = (7546)
$P_2 \rightarrow$ Complete $\rightarrow$ Avail = (7546) – (0324) = (7222) + (2775) = (9, 9, 9, 7)
$P_3 \rightarrow$ Complete $\rightarrow$ Avail = (9997) – (2502) = 7495
Hence, the system is in a safe state will value of a as 2 and value of b as 1.

18. (b)

Optimal page replacement policy is used 1, 2, 0, 3, 4, 1, 2, 3, 0, 1, 2, 3, 5, 4, 2

1
2
0

Initial 3 page fault.

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

1 + 1 + 1 + 1 +1 + 1 + 1

Total = 3 + 7 = 10 page fault

When LRU is used.

1	1	1	1	1	1	5
2	2	2	2	2	2	4
0	0	0	0	0	0	2

3 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1

Total 15 page fault.

So 15 – 10 = 5 page fault is reduced.

19. (c)

Virtual address space = 36 bit

$$\text{Number of pages} = \frac{2^{36}}{2^{13}} = 2^{23}$$

$$\text{Number of frames} = \frac{2^{29}}{2^{13}} = 2^{16}$$

Page table entry contain 2 valid bit, 2 modified bit and 2 dirty bit.

Page table entry size = 16 + 2 + 2 + 2

= 22 $\cong$ 3B

Page table size = $2^{23} \times 3$ B

= 8 × 3 MB = 24 MB

20. **(d)**

S_1: Translation look aside buffer need not to be saved.

S_2: Program counter, stack counter and general purpose register must be saved when context switch.

S_3: System calls are usually invoked by using a software interrupt.

21. **(b)**

Given, DBA = 64 bits = 8 bytes

DB size = 12 KB

$$\text{Maximum size of a file system} = \left[\text{Number of direct DBA} + \frac{\text{DB size}}{\text{DBA}} + \left(\frac{\text{DB size}}{\text{DBA}}\right)^2\right] \times \text{DB size}$$

$$\Rightarrow \quad \left[12 + \frac{12\times 2^{10}\text{B}}{8\text{ B}} + \left(\frac{12\times 2^{10}\text{B}}{8\text{ B}}\right)^2\right] \times 12 \times 2^{10}\text{B}$$

$\Rightarrow$ 144×2^{10} B + 26×2^{19} B + $9 \times 2^{18} \times 12 \times 2^{10}$ B

$\Rightarrow$ $144 \times 2^{10} + 9 \times 2^{21}$ B + 27×2^{30} B $\cong$ 27 GB

22. **(c)**

Operation	3P	12P	14V	12P	13V	12P	2V
Suspended process	2	14	0	12	0	11	9
S	0	0	0	0	1	0	0

So, total 9 blocked processes are present at end.

23. **(d)**

User-level threads only managed by user space and Kernel is not aware about this. So, a single process can get only a single processor. And one processor can execute one thread only.

24. **(b)**

In a bit map one bit is used for each block.

Given, Total memory = 1.8 GB

Block size = 256 bytes

Hence number of blocks for it will be

$$= \frac{1.8\times 2^{30}}{2^8}\text{ bit} = \frac{1.8\times 2^{30}}{2^8\times 8}\text{ bytes} = 921.6\text{ KB}$$

25. **(c)**

Working set (t, k) is the set of pages used by k most recent memory references in the last t times unit.

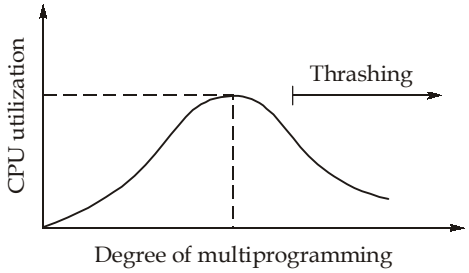
26. (c)

(a) The variable size partitioning initially process are contiguous but as processes leave their space, these partitions may not suit the new processes for allocation. Those unused partition is called external fragmentation.

(b) Degree of multiprogramming or number of partitions.

(c) False, as segmentation is memory management scheme that supports user view of memory.

(d) True, high paging activity is called thrashing.



27. (d)

28. (d)

$$\text{Number of processes} = 6$$

$$\text{Time required by each process} = 8 \text{ sec}$$

$$\text{CPU utilization} = \frac{\text{Useful time}}{\text{Total time}} = \frac{48}{60} \times 100 = 80\%$$

So, answer is 80.

29. (b)

I node hold 10 addresses $\rightarrow$ 10 KB

Single indirect block holds 256 addresses $\Rightarrow$ 256 KB

Double indirect block leads to $(2^8 \times 2^8) = 2^{16}$ KB

Triple indirect block leads to $(2^8 \times 2^8 \times 2^8)$ KB $= 2^{24}$ KB

$$\text{Maximum size} = 2^{24} \text{ KB}$$
$$= [2^{24}] \times 1 \text{ KB} = 2^{34} \text{ B}$$
$$= 2^4 \times 2^{30} \text{ B} = 16 \text{ GB}$$

30. (a)

- Mutual exclusion is satisfied by the given code.
- Progress does not satisfy because both P_0 and P_1 may enter the deadlock by making their flag true and spin lock by busy waiting in while loop.

■■■■