S.No. : 03 BS1_CS_B_290619

Operating Systems

MADE EASY

India's Best Institute for IES, GATE & PSUs

Delhi | Noida | Bhopal | Hyderabad | Jaipur | Lucknow | Indore | Pune | Bhubaneswar | Kolkata | Patna

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

CLASS TEST 2019-2020

COMPUTER SCIENCE & IT

Date of Test : 29/06/2019

ANSWER KEY > Operating Systems

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

Detailed Explanations

1. (b)

	P_1		P_2		P_3		P_4	P_5	
0	3	5	7	8	10	13	15	19	29

Scheduling time = Max completion time – Minimum arrival time = 29 – 3 = 26

2. (d)

Only triple indirect blocks are responsible for maximum file size.

$$\text{Max size} = \left(\frac{\text{DB size}}{\text{DBA}}\right)^3 \times \text{DB size} = 2\times\left[\frac{2\text{ KB}}{4\text{ B}}\right]^3 \times 2\text{ KB} = 2\times\left[\frac{2^{11}}{4^2}\right]^3 \times 2^{11}\text{ B}$$

$$= 2\times[2^9]^3\times 2^{11}\text{ B} = 2\times 2^{27}\times 2^{11}\text{ B} = 2^{39}\text{ B} = 512\text{ GB}$$

3. (c)

- An advantage of aging is to solve a starvation problem.
- An advantage of system call is to provide an interface between running program and the operating system.

4. (a)

Process	Process-1	Process-1	Process-1	Process-2	Process-3	Process-2
Before printing	$A = 3$, $B = 0$	$A = 2$, $B = 1$	$A = 1$, $B = 2$	$A = 0$, $B = 3$	$A = 0$, $B = 3$	$A = 0$, $B = 2$
Output	0	0	0	12	3	12
After printing	$A = 2$, $B = 1$	$A = 1$, $B = 2$	$A = 0$, $B = 3$	$A = 0$, $B = 3$	$A = 0$, $B = 2$	$A = 0$, $B = 2$

Correct output prefix is: 00012312....

5. (b)

Long-term scheduler can create a new process and makes a transition from either new to ready or new to suspended ready.

Medium-term scheduler makes a transition from either suspended ready to ready or suspended block to blocked. Short-term scheduler makes a transition from ready/blocked to running.

6. (c)

User thread scheduling done by thread library, because thread library is available at user level and not known to kernel. Process scheduling (process table) and kernel thread scheduling (thread table) are part of the kernel and OS is responsible for the scheduling.

7. (d)

Deadlock prevention, deadlock avoidance (Banker's algorithm) and deadlock detection and recovery are methods used for deadlock handling.

8. (a)

$$\text{Total instances} > (k_1 - 1) * N_1 \text{ process } (k_2 - 1) * N_2 \text{ process}$$

$$30 > 6 \times (4 - 1) + (4 - 1) * N \text{ Process}$$

$$30 > 18 + 3N$$

$$12 > 3N$$

$$4 > N$$

$$N = 4$$

When N = 4
the deadlock can occur, So, maximum value of $N = 4 - 1 = 3$. So, that system is in safe state.

9. (b)
Prevention : Ensuring that atleast one of the four conditions for deadlock never holds.
Avoidance : Ensuring that process never enter the unsafe regions of resource allocation.
Detection : Allowing deadlock to occur, and then recovering from the deadlock.

10. (c)
First fit allocates the first free hole.
Best fit allocates the smallest free hole that is big enough.
Next fit allocates the first free hole from the last allocation.
Worst fit allocates the largest free hole.

11. (a)
Mutual exclusion satisfies for multiple readers when they are reading the shared document.

12. (d)
The output is 'TGE'. So, to print 'T', we must give a value of 1 to semaphore b and should block rest three processes.
Now, process 3, after printing T, will give signal to semaphore a, which will wake up process 1 and will print 'G' and given signal to semaphore 'b' and 'c'. On giving signal to semaphore 'c', process '2' will get awake. But 'a' should not be printed in the output hence 'c' should be given value '–1'.
Process 4 will also awake after process 3 on signal 'a', but it will again be blocked by wait (b).

13. (b)
If, we remove the lock while acquiring the fork. It may lead to deadlock, if all process execute (i) statement before any philosopher has execute (ii) statement.
Removal of (iii) and (iv) will not affect the code, since no conflict can occur doing the V operation on forks.

14. (c)
- 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 flags true and spin lock by busy waiting in while loop.
- Bounded wait satisfies.

15. (d)
Page table can be stored in main memory, Hard disk and level-2 cache.

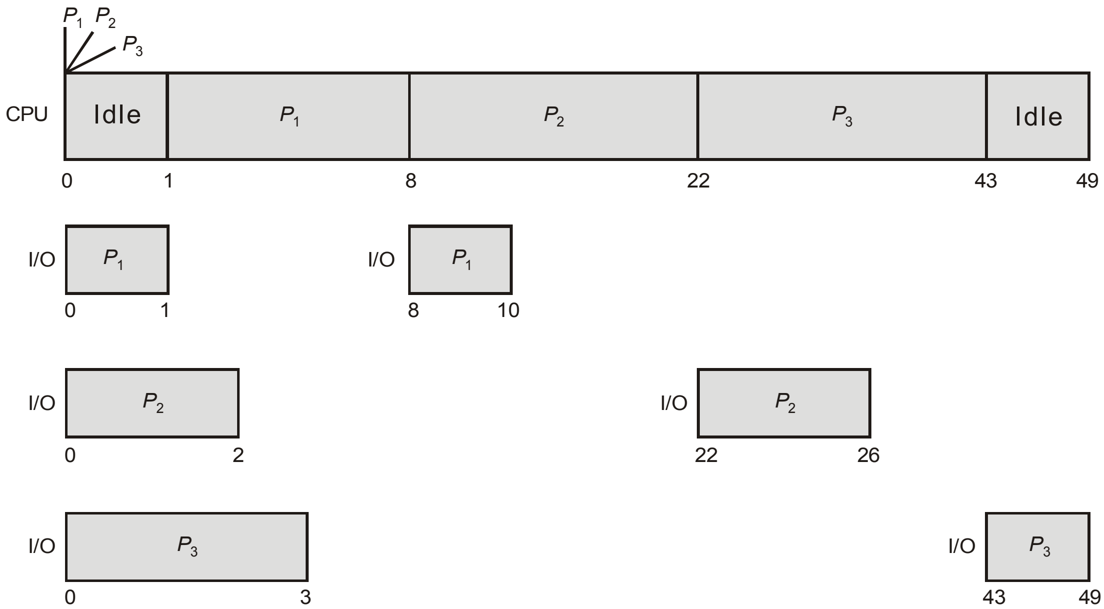
16. (d)
Deadlock can be recovered from any of the following.
(a) Resource preemption
(b) Rollback of process
(c) Killing process

17. (a)
The value of Registers, Program counter and Stack pointers will be changed. Memory management information does not change.

18. (c)

Process	Burst Time (B.T.)
P_1	10
P_2	20
P_3	30

Process	I/O	CPU	I/O
P_1	1	7	7
P_2	2	14	14
P_3	3	21	21



CPU idle time = (1 – 0) + (49 – 43) = 1 + 6 = 7 units

Total time taken to complete all processes = 49 units

CPU idle % time = $\frac{7}{49} \times 100\% = 14.2857\%$

19. (d)

Since 4 distinct page numbers are only to be accessed. Hence the best condition i.e., the condition with minimum number of page faults will be accessing all those elements repeatedly that are in the frame already, which will give maximum 4 page faults.

If, considered the worst case, it will be on every iteration, we are accessing the same element that has been removed from the frame, which will give 52 page faults.

20. (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.

21. (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.

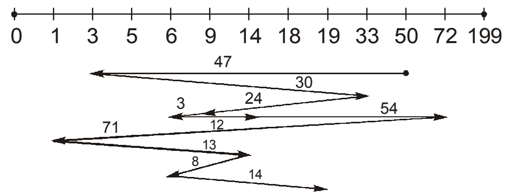
22. (c)

- P_1 and P_2 can enter the critical section simultaneously and possible to access the data at x. So, no mutual exclusion called race condtion.
- Both can be in deadlock by execution second wait by P_1 and P_2 alternatively.

∴ Race condition and deadlock can occur by the threads.

23. (c)

- FCFS



Total time = 47+30+27+12+54+71+13+8+14 = 276 msec

24. (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 → Complete → Avail = (0000 + 6214) = 6214

P_1 → Complete → Avail = (6214) – (3200) = (3014) + (3512) = (6526)

P_0 → Complete → Avail = (6526) – (2222) = (4304) + (3242) = (7546)

P_2 → Complete → Avail = (7546) – (0324) = (7222) + (2775) = (9, 9, 9, 7)

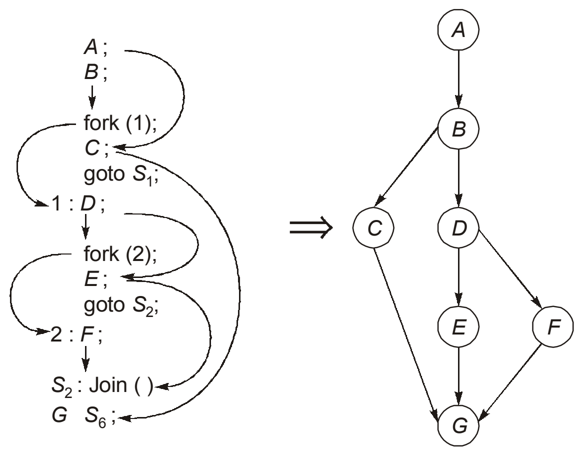
P_3 → Complete → Avail = (9997) – (2502) = 7495

Hence, the system is in a safe state will value of a as 2 and value of b as 1.

25. (d)

- Switching between two user level threads only require procedure calls not context switching.
- All Kernal threads operations are implemented in Kernal, and OS schedules all threads in the system.
- Since user level threads are transport to Kernal, hence are not scheduled independently and hence are not given independent time slice.
- Threads do share the code segment.

26. (c)



27. (c)

(i) For segment 0, length = Logical address 99 < 124

∴ Physical address = 330 + 99 = 429

(ii) For segment 2, length = Logical address 78 < 99

∴ Physical address = 111 + 78 = 189

(iii) For segment 1, length = Logical address 265 < 211

∴ Trap

(iv) For segment 3, length = Logical address 222 < 302

∴ Physical address = 498 + 222 = 720

(v) For segment 0, length = Logical address 111 < 124

∴ Physical address = 330 + 111 = 441

28. (c)

Page size = 8 KB ⇒ 13 bit offset

Number of frame bits = 32 – 13 = 19 bits

Page table entry = Valid + Translation (frame bits)

= 1 + Frame bits

= 1 + 19 = 20 bits

Page table size = 20 Mbytes

Number of pages = Number of page table entries

$$= \frac{20 \text{ Mbytes}}{20 \text{ Bits}} = 8 \text{ M} = 2^{23} \text{ pages}$$

∴ 23 bits needed for page and 13 bits offset

Length of virtual address = 23 + 13 = 36 bits.

29. (d)

Space occupied by holes = {Total Memory Space} – {Space occupied by all processes}

$$\frac{n}{2} \times kS = M - n * S$$

$$\frac{n}{2} kS + nS = M$$

$$M = nS\left(\frac{k}{2} + 1\right)$$

30. (b)

I node hold 10 addresses → 10 KB

Single indirect block holds 256 addresses → 256 KB

Double indirect block leads to → $[2^8 * 2^8]$ KB = 2^{16} KB

Triple indirect block leads to → $[2^8 * 2^8 * 2^8]$ KB = 2^{24} KB

Maximum size = $[2^{24}] \times 1$ KB = 16 GB

■■■■