

## Duration : 1:00 hr.

Maximum Marks: 50

## Read the following instructions carefully

1. This question paper contains 30 objective questions. Q.1-10 carry one mark each and Q.11-30 carry two marks each.
2. Answer all the questions.
3. Questions must be answered on Objective Response Sheet (ORS) by darkening the appropriate bubble (marked A, B, C, D) using HB pencil against the question number. Each question has only one correct answer. In case you wish to change an answer, erase the old answer completely using a good soft eraser.
4. There will be NEGATIVE marking. For each wrong answer $1 / 3$ rd of the full marks of the question will be deducted. More than one answer marked against a question will be deemed as an incorrect response and will be negatively marked.
5. Write your name \& Roll No. at the specified locations on the right half of the ORS.
6. No charts or tables will be provided in the examination hall.
7. Choose the Closest numerical answer among the choices given.
8. If a candidate gives more than one answer, it will be treated as a wrong answer even if one of the given answers happens to be correct and there will be same penalty as above to that questions.
9. If a question is left blank, i.e., no answer is given by the candidate, there will be no penalty for that question.

## Q. No. 1 to Q. No. 10 carry 1 mark each

Q. 1 Match the following:
A. Present bit 1. Difference in contents of page in RAM and the disk
B. Dirty bit
2. Moving page from disk to memory
C. Swap out
3. Moving page from memory to disk
D. Swap in
4. Page is present in RAM

|  | $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{C}$ | $\mathbf{D}$ |
| :--- | :--- | :--- | :--- | :--- |
| (a) 4 | 3 | 1 | 2 |  |
| (b) 2 | 1 | 4 | 3 |  |
| (c) 3 | 4 | 2 | 1 |  |
| (d) 4 | 1 | 3 | 2 |  |

Q. 2 Consider the following memory map using multiprogram with partition model. Shaded region shows memory in use and unshaded region represent free memory.

```
65K
```

Request for memory follows the following order:
$100 \mathrm{~K}, 25 \mathrm{~K}, 125 \mathrm{~K}, 50 \mathrm{~K}$. Which of the following allocation satisfies the above request?
I. Best Fit
II. First Fit
III. Worst Fit
(a) Only 1 and II
(b) Only II and III
(c) All I, II and III
(d) None of these
Q. 3 The system has 5 process and 3 resources (A, B, C). The maximum count of resources are $(10,5,7)$. Consider the following table of resource allocation.

|  | $\mathbf{M A X}$ |  |  |  | Allocated |  |  |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $\mathbf{( A}$ | B | C) | (A | B | C $)$ |  |
| $\mathrm{P}_{0}$ | 7 | 5 | 3 | 0 | 1 | 0 |  |
| $\mathrm{P}_{1}$ | 3 | 2 | 2 | 2 | 0 | 0 |  |
| $\mathrm{P}_{2}$ | 9 | 0 | 2 | 3 | 0 | 2 |  |
| $\mathrm{P}_{3}$ | 2 | 2 | 2 | 2 | 1 | 1 |  |
| $\mathrm{P}_{4}$ | 4 | 3 | 3 | 0 | 0 | 2 |  |

What will be the safe sequence?
(a) $P_{1}, P_{2}, P_{3}, P_{4}, P_{0}$
(b) $P_{0^{\prime}} P_{1^{\prime}}, P_{3^{\prime}}, P_{4^{\prime}} P_{2}$
(c) $P_{1}, P_{3}, P_{4}, P_{0}, P_{2}$
(d) Unsafe sequence
Q. 4 Consider the following statements:
$S_{1}$ : In linked allocation of disk spaces if a program $P_{1}$ reads $20^{\text {th }}$ disk block and now $P_{1}$ wants to read $13^{\text {th }}$ disk block then it will again start accessing from the first disk block.
$S_{2}$ : Linked allocation can have internal fragmentation.

Which of the above statements is/are correct?
(a) Only $S_{1}$
(b) Only $S_{2}$
(c) Both $S_{1}$ and $S_{2}$
(d) None of these
Q. 5 Consider a system using 2-level paging and the virtual address is 38 bits. The most significant 10 bits are used to index the page directory and next 16 bits index the page table. Each entry in both levels is 4 bytes. What is the maximum number of page tables that a process can have?
(a) 500
(b) 1025
(c) 2050
(d) 1090
Q. 6 Consider a program, which spawn 20 threads to find out the sum of elements in a shared array of 200 . Each thread $i$, takes elements $\mathrm{A}[i]$ to $\mathrm{A}[(20$ * $i)-1]$ and compute local sum loc-sum[ $i$ ] and eventually adds loc_sum $[i]$ to a shared variable tot_sum for total sum which of the following needs to be placed inside a critical section?
(a) Reading array valves
(b) Add to tot_sum
(c) Both (a) and (b)
(d) None of these
Q. 7 Which of the below statement is false:
(a) Kernel stack can be used to store the context of a process.
(b) User stack is used to store the function details during function calls.
(c) Kernel stack is used to store the user program functional arguments.
(d) Context of the process is useful to restart the process after some time.
Q. 8 Consider a system having n CPU's, $K$ processes and $K>n$. Calculate the upper bound and lower bound for the number of processes in the RUNNING, READY and BLOCKED states respectively in lower and upper bound is
(a) Upper bound is $\mathrm{K}, \mathrm{K}, \mathrm{K}$ and lower bound is $1,1,1$
(b) Upper bound is $n, \mathrm{~K}, \mathrm{~K}$ and lower bound is $0,0,0$.
(c) Upper bound is $K, K, n$ and lower bound is $0,0,0$.
(d) Upper bound is $\mathrm{K}, n, \mathrm{~K}$ and lower bound is $1,1,0$.
Q. 9 Suppose we want to synchronize two concurrent process $P$ and $Q$ using binary semaphores $S$ and $T$.

```
Process P:
While(1) {
    W:
        Print '5';
        Print '5';
    X:
}
```

Synchronization statements can be inserted only at point $W, X, Y, Z$ respectively. Which of the following can lead to an output starting with '66556655'?
(a) $\mathrm{P}(\mathrm{S}), \mathrm{V}(\mathrm{T}), \mathrm{P}(\mathrm{T}), \mathrm{V}(\mathrm{S})$ and initially $\mathrm{S}=0$ and $\mathrm{T}=1$
(b) $\mathrm{P}(\mathrm{S}), \mathrm{V}(\mathrm{S}), \mathrm{P}(\mathrm{T}), \mathrm{V}(\mathrm{T})$ and initially $\mathrm{S}=0$ $=\mathrm{T}=1$
(c) $\mathrm{P}(\mathrm{S}), \mathrm{V}(\mathrm{S}), \mathrm{P}(\mathrm{T}), \mathrm{V}(\mathrm{T})$ and initially $\mathrm{S}=1$ and $\mathrm{T}=0$
(d) $\mathrm{P}(\mathrm{S}), \mathrm{V}(\mathrm{T}), \mathrm{P}(\mathrm{T}), \mathrm{V}(\mathrm{S})$ and initially $\mathrm{S}=1$ and $\mathrm{T}=0$
Q. 10 Consider 3 processes that start simultaneously. Each process from start to finish takes 40,50 and 60 ms respectively of this $20 \%$ of the time is spent waiting for i o events and then next $80 \%$ time for computation.
The scheduler uses FCFs scheduling. Assuming that all I/O operations are overlapped. The minimum time in milliseconds for which the CPU idle is $\qquad$ ـ.
(a) 6
(b) 7
(c) 8
(d) 9

## Q. No. 11 to Q. No. 30 carry 2 marks each

Q. 11 Consider the following code comprising of a swap function which is atomic.
void swap (bool * a, bool * b)
\{
bool temp;
temp $=$ *a;

* $\mathrm{a}=$ *b;
*b = temp;
\}
bool Lock = false;
void process (int i)
\{ bool key; while(1) \{
// Non-critical section
Key = True;
While (Key = = true)
Swap (\&Lock, \&Key);
// Critical-section
Lock = false;
$\}$
\}
Which of the following is correct regarding above code?
(a) The code provides mutual exclusion but not progress.
(b) The code provides progress but not mutual exclusion.
(c) The code provides bounded waiting but not mutual exclusion.
(d) The code provides mutual exclusion but not bounded waiting.
Q. 12 Which of the following is correct about monitors?
(a) Prevents multiple process from executing monitor code of the same time.
(b) Mutual exclusion not guaranteed.
(c) Process can access the monitor's data from procedures declared outside the monitor.
(d) None of these
Q. 13 Consider the following code segment:
$P_{1}$ ( ) \{

$$
\mathrm{P}(\mathrm{~S})
$$

$a=a+15$;
V(S)
\}

Assume initial value of $a=0$ and $b=20$ and semaphore $\mathrm{S}=1 . \mathrm{P}()$ and V() are usual semaphore up and down operation $a$ and $b$ are shared variable.
The numbers of distinct values that $a$ can possibly take after concurrently executing the above code segment is
(a) 5
(b) 7
(c) 4
(d) 2
Q. 14 Which of the following is not a necessary condition to occur the deadlock?
(a) Mutual exclusions
(b) No "Hold and wait"
(c) No "preemption"
(d) Circular wait
Q. 15 Consider the following statements:
$S_{1}$ : Demand paging requires the programmer to instruct the operating system to load a particular virtual memory page.
$S_{2}$ : Dynamic loading follows inefficient memory utilization.
$S_{3}$ : Pages that are shared between 2 or more processes can never be swapped out of the memory.
Which of the above is correct?
(a) Only $S_{1}$
(b) Only $S_{2}$ and $S_{3}$
(c) All of the above
(d) None of these
Q. 16 Consider the following page reference string:

$$
3,2,1,5,6,3,2,7,1,3,6,7,3,2,6
$$

Assume there are 4 frames are available. $x$ represents number of page faults using FIFO, y represents page faults using LRU and $z$ represents page fault using optimal technique. Which of the following is correct?
(a) $z<y<x$
(b) $z<x<y$
(c) $z=x<y$
(d) $z=y=x$
Q. 17 Consider the following statements:
$S_{1}$ : To prevent deadlock, if a OS implements a policy that requires a process to release all resources before making a request, is free from deadlock but starvation may occur.
$S_{2}$ : Deadlock avoidance requires resource requirements in advance.

Which of the above is correct?
(a) Only $S_{1}$
(b) Only $S_{2}$
(c) Both $S_{1}$ and $S_{2}$
(d) None of these
Q. 18 Consider a system where 2-level paging is used with TLB support. The access time for TLB is 20 ns while main memory access time is 80 ns .130 pages references found in TLB and total 400 reference is generated. What is the effective memory access time?
(a) 180 ns
(b) 208 ns
(c) 176 ns
(d) 195 ns
Q. 19 In a demand paging scheme, page fault service time is 8 ms if replaced page is not modified or there is empty frame to accommodate the new page and it takes 15 ms if the page to be replaced is modified page. Main memory access time is 5 ms and $72 \%$ of the page to be replaced is modified In percentage what can be the maximum page fault rate to get the effective memory access time not more than 7 ms ?
(a) 20.52
(b) 24.87
(c) 15.57
(d) 29.87
Q. 20 Consider a disk has 150 cylinders, numbered from 0 to 149 . Currently the disk arm is at 80 and moving towards higher cylinders. There is disk access requests for cylinders are $30,78,96,123,141,15,13,68$. What will be the absolute difference traversed by R/ w head when CSCAN and CLOOK is used?
(a) 42
(b) 32
(c) 52
(d) 48
Q. 21 Consider the following program:
int main( )
\{

$$
\text { for }(i=1 ; i<n ; i++)
$$

\{ if (fork( ) = = 0)
printf("OS");
\}
printf("GATE");
\}
Number of times OS and GATE printed respectively are:
(a) $2^{n}$ and $2^{n}$
(b) $2^{n-1}$ and $2^{n-1}$
(c) $2^{n-1}-1$ and $2^{n-1}$
(d) $2^{n}$ and $2^{n-1}$
Q. 22 Consider 2 process $A()$ and $B()$ that are concurrent and uses 3 semaphore i.e. mutex, $Q$ and $R$ and all are initialized to 1 and these semaphores are shared between the processes. Q is a semaphore on file 1 and $R$ on file 2.
A()
B( )
\{
P (mutex);
$P(Q)$;
*write to file 1 *
$\mathrm{P}(\mathrm{R})$;
*write to file $2^{*}$
V(Q);
V(mutex);
$\mathrm{P}(\mathrm{Q})$;
*read from file 1*
*read from file 2*
V(R);
V(Q);
\}

Which of the following is true for above process?
(a) No deadlock but starvation occurs.
(b) Both deadlock and starvation occurs.
(c) Deadlock but no starvation occurs.
(d) Neither deadlock nor starvation occurs.
Q. 23 Consider the following statements:
$S_{1}$ : A lock that uses busy waiting will not waste CPU time.
$S_{2}$ : Inter process communication message passing requires Kernel supports.
$S_{3}$ : User level threads are not suitable for non blocking tasks.
Which of the above statements are correct?
(a) Only $S_{1}$ and $S_{2}$
(b) Only $S_{2}$
(c) Only $S_{1}$
(d) All of these
Q. 24 Consider the following statements:
I. Multiprogramming is used to increase CPU utilization, which time sharing is used to increase CPU responsiveness in interacting with user.
II. With a multi-level feedback queue scheduler, high priority jobs will be placed in the top level queue and low priority jobs will be placed in the bottom level queue.
III. If all the jobs have identical run lengths, a RR scheduler provides better average turnaround time than FCFS.
The number of correct statements is/are:
(a) 1
(b) 2
(c) 3
(d) 0
Q. 25 Consider the following concurrent program:

Int $x=0$;
Int $y=0$;
Parbegin \{
Begin \{

$$
x=2 ;
$$

$$
y=y+x ;
$$

\}
Begin \{
$y=5 ;$
$x=x+5 ;$
\}
\}
What will be the final value of $x$ and $y$ after completion of the above concurrent program?
(a) $x=2, y=12$
(b) $x=2, y=7$
(c) $x=7, y=15$
(d) $x=6, y=7$
Q. 26 Consider the following statements:
(a) Round-Robin scheduling policy is most suitable for a time shared operating system.
(b) A multi-user, multiprocessing operating system can not be implemented on hardware that does not support atleast 2 modes of CPU execution.
(c) Interrupt from CPU temperature sensor will be handled at a higher priority than interrrupt from hard disk by computer.
(d) While switching context from process A to process B, operating system does not change the address translation table.
Which of the above statement is false?
Q. 27 Consider the methods used by process $P_{1}$ and $P_{2}$ for accessing their critical sections whenever needed, as given below. The initial values of shared Boolean variables, $S_{1}$ and $S_{2}$ are random assigned.

| Method used by $\mathrm{P}_{1}$ | Method used by $\mathrm{P}_{2}$ |
| :---: | :---: |
| ```do { flag[i] = True; S}=\mp@subsup{S}{2}{ while (P); critical section flag[3] = False; Remainder section } while(1);``` | ```do { flag[j] = True; S}=\mp@subsup{S}{2}{}+1 while (Q); critical section flag[j]= False; Remainder section } while(1);``` |

For the program to guarantee mutual exclusion, the predicate P and Q in the while loop should be
(a) P: Flag $[j]=$ true and $S_{1}!=S_{2}$
$\mathrm{Q}:$ Flag $[i]=$ true and $\mathrm{S}_{1}=\mathrm{S}_{2}$
(b) P : Flag $[j]=$ true and $S_{1}=S_{2}$ $\mathrm{Q}:$ Flag $[i]=$ true and $\mathrm{S}_{1}!=\mathrm{S}_{2}$
(c) P: Flag $[i]=$ true and $S_{1}!=S_{2}$ Q: Flag $[j]=$ true and $S_{1}==S_{2}$
(d) P: Flag $[i]=$ true and $\mathrm{S}_{1}=\mathrm{S}_{2}$ $\mathrm{Q}:$ Flag $[j]=$ true and $\mathrm{S}_{1}!=\mathrm{S}_{2}$
Q. 28 Consider the following statements:

1. Loop instructions cannot be interrupted till they complete.
2. Nearest cylinder next disk scheduling strategy gives the best throughput in comparison to first come first serve scheduling strategy.
3. Using large file block size in a fixed block size file system leads to poor disk throughput.
Which of the above statements are false?
(a) Only 1
(b) Only 1, 2 and 3
(c) Only 1 and 3
(d) Only 2
Q. 29 Consider the following statements:
4. Contiguous allocation of disk space can lead to external fragmentation.
5. Under linked allocation of disk space, if a program has just read the tenth disk block, now if it want to read the sixth disk block, it will again start accessing from the first disk block.
6. Indexed allocation can not lead to external fragmentation.
Which of the above statements is/are correct?
(a) Only 1 and 2
(b) Only 2 and 3
(c) All of these
(d) None of these
Q. 30 A single processor system has five resource types A, B, C, D and E which are shared by four processes. The current allocation and maximum needs are as follows:

| Process | Allocated |  |  |  |  | Maximum |  |  |  |  |  | Available |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $\boldsymbol{A}$ | $\boldsymbol{B}$ | $\boldsymbol{C}$ | $\boldsymbol{D}$ | $\boldsymbol{E}$ | $\boldsymbol{A}$ | $\boldsymbol{B}$ | $\boldsymbol{C}$ | $\boldsymbol{D}$ | $\boldsymbol{E}$ | $\boldsymbol{A}$ | $\boldsymbol{B}$ | $\boldsymbol{C}$ | $\boldsymbol{D}$ | $\boldsymbol{E}$ |  |
| $P_{0}$ | 1 | 0 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 3 | 0 | 0 | 1 | 2 | 3 |  |
| $P_{1}$ | 2 | 0 | 1 | 1 | 0 | 2 | 2 | 2 | 1 | 0 |  |  |  |  |  |  |
| $P_{2}$ | 1 | 1 | 0 | 1 | 0 | 2 | 1 | 3 | 1 | 0 |  |  |  |  |  |  |
| $P_{3}$ | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 2 | 2 | 1 |  |  |  |  |  |  |

Which of these processes will finish LAST?
(a) $\mathrm{P}_{0}$
(b) $P_{1}$
(c) $\mathrm{P}_{2}$
(d) None because system is in deadlock

## CLASS TEST

mRDE ERSU
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 : 05/07/2023

## ANSWER KEY

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

## DETAILED EXPLANATIONS

1. (d)

- Present bit $\rightarrow$ Page is present in RAM
- Dirty bit $\rightarrow$ Differences in contents of page in RAM and the disk
- Swap out $\rightarrow$ Moving page from memory to disk
- Swap in $\rightarrow$ 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 175 K 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 |
| $\mathrm{P}_{0}$ | 7 | 5 | 3 | 0 | 1 | 0 | 7 | 4 | 3 |
| $\mathrm{P}_{1}$ | 3 | 2 | 2 | 2 | 0 | 0 | 1 | 2 | 2 |
| $\mathrm{P}_{2}$ | 9 | 0 | 2 | 3 | 0 | 2 | 6 | 0 | 0 |
| $\mathrm{P}_{3}$ | 2 | 2 | 2 | 2 | 1 | 1 | 0 | 1 | 1 |
| $\mathrm{P}_{4}$ | 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 $n, \mathrm{~K}, \mathrm{~K}$ and similarly in lower bound are $0,0,0$.
9. (a)

Process P:
W: P(S)
Print '5'; Print '5';

X: V(T)

Process Q:
$\mathrm{Y}: \mathrm{P}(\mathrm{T})$ Print ' 6 '; Print ' 6 ';

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 $P_{2}$

$$
\begin{aligned}
& a=0+15=15 \\
& a=20+20=40
\end{aligned}
$$

$P_{2}$ executed first then $P_{1}$

$$
\begin{aligned}
& a=20+20=40 \\
& a=40+15=55
\end{aligned}
$$

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
$\therefore$ No "Hold and wait" is not a necessary condition to occur the deadlock.
5. (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

|  | (2) | (1) (5) |  |  | (2) |  |  | 3 (6) 7 |  | (3) (2) |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  | 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 |  |

LRU replacement
(3) (2) (1) (5) (6) (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

| $3)$ |  |  |  |  | 3 |  | 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 |  |

So,

$$
z<. y<x
$$

17. (c)

Both the statements $S_{1}$ and $S_{2}$ are correct.
18. (b)

$$
\begin{aligned}
\text { TLB hit ratio } & =\frac{130}{400}=0.325 \\
\text { E.M.A.T. } & =\mathrm{P} \times(\mathrm{t}+\mathrm{M})+(1-\mathrm{P})(\mathrm{t}+3 \mathrm{M})=0.325 \times(20+80)+0.675(20+3+80) \\
& =32.5 \mathrm{~ns}+0.67 \times 260 \mathrm{~ns}=32.5 \mathrm{~ns}+175.5 \mathrm{~ns}=208 \mathrm{~ns}
\end{aligned}
$$

19. (b)

Let page fault rate be $P$.

$$
\begin{aligned}
\text { EMAT } & \geq \mathrm{P} *[0.72 * 15+0.28 * 8]+(1-\mathrm{P}) * 5 \mathrm{~ms} \\
7 & \geq \mathrm{P} * 13.04+(1-\mathrm{P}) * 5 \\
7 & \geq 8.04 \mathrm{P}+5 \\
8.04 \mathrm{P} & \leq 2 \\
\mathrm{P} & \leq 0.2487 * 100 \\
\mathrm{P} & \leq 24.87 \%
\end{aligned}
$$

20. (a)


Total distance traversed by R/w head

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

Figure
Total distance traversed by $\mathrm{R} / \mathrm{w}$ head

$$
\begin{aligned}
& =(141-80)+(141-13)+(78-13)=61+128+65=254 \\
\text { Absolute difference } & =296-254=42
\end{aligned}
$$

21. (c)

OS Printed $\rightarrow 2^{n-1}-1$
Gate Printed $\rightarrow 2^{n-1}$
22. (b)

1. A()

$$
\begin{aligned}
\mathrm{P}(\text { mutex }) & \Rightarrow \text { mutex }=0 \\
\mathrm{P}(\mathrm{Q}) & \Rightarrow \mathrm{Q}=0
\end{aligned}
$$

2. $B()$

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

3. A()

$$
\begin{aligned}
\mathrm{P}(\mathrm{R}) & \Rightarrow \mathrm{R}=0 \\
\mathrm{~V}(\mathrm{Q}) & \Rightarrow \mathrm{Q}=1 \Rightarrow \text { Process } \mathrm{B}() \text { awake }
\end{aligned}
$$

4. B( )

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

$$
P(R) \Rightarrow \text { Process } B() \text { goes in sleep mode }
$$

5. A()

$$
\begin{aligned}
\mathrm{V}(\text { mutex }) & \Rightarrow \text { mutex }=1 \\
\mathrm{P}(\mathrm{Q}) & \Rightarrow \text { Process } \mathrm{A}() \text { goes in sleep mode }
\end{aligned}
$$

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.
4. (c)
5. As files are allocated and deleted, the free disk space is broken into little pieces, hence can lead to external fragmentation.
6. Linked-allocation can be used effectively only for sequential access file. To find the $i^{\text {th }}$ block of a file. We must start at the beginning of that file and follow the pointers until we get the $i^{\text {th }}$ block.
7. Statement is correct.
8. (b)

Calculating the need matrix

|  | $\boldsymbol{A}$ | $\boldsymbol{B}$ | $\boldsymbol{C}$ | $\boldsymbol{D}$ | $\boldsymbol{E}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $P_{0}$ | 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 $\mathrm{P}_{3}$ can be satisfied.
Remaining $=(00123)-(00111)=(00012)+(11221)=(11233)$
Now $\mathrm{P}_{0}$ can be executed,
Remaining $=(11233)-(01002)=(10231)+(11213)=(21444)$
Now $\mathrm{P}_{2}$ can be executed,
Remaining $=(21444)-(10300)=(11144)+(21310)=(32454)$
Now $\mathrm{P}_{1}$ can be executed.

