GATE 2020
PSUs 2020

MADE EASY Publications

Revised & Updated Edition

POSTAL STUDY PACKAGE

COMPUTER SCIENCE & IT

Operating System

Objective Practice Sets

POSTAL Study Package 2020

Computer Science & IT

Objective Practice Sets

Operating System

Contents

MADE EASY Publications

1 CHAPTER
Basic Concepts of Operating System

Q.1 Which of the following should be allowed only in Kernel mode?
1. Changing mapping from virtual to physical address.
2. Mask and unmask interrupts.
3. Disabling all interrups.
4. Reading status of processor.
5. Reading time of day.

(a) 1, 2 and 3 (b) 1, 2, 4 and 5
(c) 2, 3 and 5 (d) all of these

Q.2 An interrupt handler is a
(a) location in memory that keeps track of recently generated interrupts
(b) peripheral device
(c) utility program
(d) special numeric code that indicates the priority of a request

Q.3 Executing more than one program concurrently by one user on one computer is known as
(a) multiprogramming (b) time-sharing
(c) multitasking (d) multiprocessing

Q.4 The simultaneous processing of two or more programs by multiple processors is
(a) multitasking
(b) multiprogramming
(c) time-sharing
(d) multiprocessing

Q.5 Which of the following does not interrupt a running process?
(a) timer interrupts (b) device
(c) power failure (d) scheduling process

Q.6 System call is used to access
(a) I/O functionality
(b) operating system functionality
(c) application functionality
(d) None of the above

Q.7 Swapping is performed by
(a) long term scheduler
(b) mid term scheduler
(c) short term scheduler
(d) dispatcher

Q.8 Choose the false statement
(a) static linking requires no support of OS
(b) dynamic linking requires no support of OS
(c) dynamic loading requires no support of OS
(d) none of the above

Q.9 Assume that the kernel mode is non-preemptive. What happens when an I/O interrupt comes while a process 'P_1' is running in the kernel mode on the CPU?
(a) CPU is given to the process for which the I/O has completed
(b) CPU is given to some other process based on the scheduling policy
(c) P_1 continues to execute on the CPU
(d) None of the above

Q.10 Overlay is
(a) a part of an operating system
(b) a specific memory location
(c) a single contiguous memory that was used in the olden days for running large programs by swapping
(d) overloading the system with many user files

Q.11 When an interrupt occurs, an operating system
(a) ignores the interrupt
(b) always changes the stage of the interrupted process after processing the interrupt

(c) always resumes execution of the interrupted process after processing the interrupt
(d) may change the state of the interrupted process to "blocked" and schedule another process

Q.12 Consider the following statements:
S_1: The OS is designed to maximize the resource utilization.
S_2: The control program manages the system programs.
Which of the above statements is/are true?
(a) S_1 is true S_2 is false
(b) S_2 is true and S_1 is false
(c) Both S_1 and S_2 are true
(d) Both S_1 and S_2 are false

Q.13 Bootstrap loader is always stored in
(a) cache (b) ROM
(c) RAM (d) disk

Q.14 Which of the following is true?
(a) Overlays are used to increase the size of physical memory.
(b) Overlays are used to increase the logical address space.
(c) When overlays are used, the size of a process is not limited to the size of physical memory.
(d) Overlays are used whenever the physical address space is smaller than the logical address space.

■■■■

Answers Basic Concepts of Operating System

1.	(a)	2.	(c)	3.	(c)	4.	(d)	5.	(b)	6.	(b)	7.	(b)	8.	(b)	9.	(c)		
10.	(c)	11.	(d)	12.	(a)	13.	(b)	14.	(c)										

Explanations Basic Concepts of Operating System

1. (a)
Only critical services must reside in the kernel. All services mentioned except reading status of processors and reading time of the day are critical.
Hence option (a) is correct.

9. (c)
When the kernel is non-preemptive and any process is running in a kernel mode, then process continues to run until either it completes or it waits for some input/output.

11. (d)
When a interrupt occurs operating system decides the request on the fact that the interrupt has higher priority or less priority. If less, the interrupted process is resumed and only after the execution of process, the interrupt is handled. However if interrupt has higher priority the process is blocked and interrupt is entertained. Hence an operating system may or may not change the state of the interrupted process to "blocked" and schedule another process.

14. (c)
By using the overlays we can execute much greater processes simultaneously which cannot be execute and reside in the memory at the same time. In this the process to be executed process brought to memory only when it is needed at the time of execution.

■■■■

2 CHAPTER

Process and Threads

Q.1 Which of the following statements comparing the context of a thread with that of a process is true?

(a) two processes will not share any context; two threads of a same process will only share the data and the code (text) areas of the context

(b) two processes will not share any context; two threads of a same process will share the data, code (text) and the stack areas of the context

(c) two processes will share the data and the code (text) areas of the user context; two threads of a same process will only share the register context

(d) the overhead involved in context switching for threads is much higher than that for processes

Q.2 Which of the following information is not part of process control block

1. Process state
2. List of open files
3. Process page table
4. Stack pointer

(a) only 3 (b) 3 and 4
(c) 2 and 4 (d) None of these

Q.3 Convoy effect is a result of

(a) one long CPU bound process and many other CPU bound processes are waiting

(b) many CPU bound processes and less I/O bound processes

(c) many CPU and I/O bound processes

(d) proper mix of CPU and I/O bound processes

Q.4 In a time-sharing operating system, when the time slot given to a process is completed, the process goes from the RUNNING state to the

(a) BLOCKED state

(b) READY state

(c) SUSPENDED state

(d) TERMINATED state

Q.5 In a multiprogramming environment

(a) the processor executes more than one process at a time

(b) the programs are developed by more than one person

(c) more than one process resides in the memory

(d) a single user can execute many programs at the same time

Q.6 If a system contains n processors and n processes then what will be maximum and minimum processes in running state respectively.

(a) n, n (b) $n, 0$
(c) $n^2, 0$ (d) n^2, n^2

Q.7 Match List-I with List-II select the correct answer using the codes given below the lists:

List-I

A. run → ready
B. run → blocked
C. blocked → run
D. run → terminated

List-II

1. not possible
2. when a process terminates itself
3. when a process time quantum expires
4. when a process issues an input / output request

Codes:

	A	B	C	D
(a)	1	4	3	2
(b)	2	1	3	4
(c)	3	4	1	2
(d)	1	4	2	3

Q.8 While designing a kernel, an operating system designer must decide whether to support kernel-level or user-level threading. Which of the following statements is/are true?

1. Kernel-level threading may be preferable to user-level threading because storing information about user-level threads in the process control block would create a security risk.
2. User-level threading may be preferable to kernel-level threading because in user-level threading, if one thread blocks on I/O, the process can continue.

(a) 1 only (b) 2 only
(c) 1 and 2 only (d) None of these

Q.9 Consider the following statements with respect to user-level threads and kernel-supported threads

(*i*) Context switching is faster with kernel-supported threads
(*ii*) For user-level threads, a system call can block the entire process
(*iii*) Kernel-supported threads can be scheduled independently
(*iv*) User-level threads are transparent to the kernel

Which of the above statements are true?
(a) (*ii*), (*iii*) and (*iv*) only
(b) (*ii*) and (*iii*) only
(c) (*i*) and (*iii*) only
(d) (*i*) and (*ii*) only

Q.10 Assume process A has 3 user level threads and process B has 4 Kernel-level threads. Consider while process A is running in CPU, process B is waiting in ready queue. If one of the thread in A is blocked then find status of A threads and B threads?

(a) All A threads are blocked and all B threads are blocked
(b) All A threads are blocked and B threads are not blocked
(c) All B threads are blocked and A threads are not blocked
(d) None of these

Q.11 Assume T_1 and T_2 are two threads of the same process. Consider the following information:

1. Data section
2. Stack section
3. Code section
4. I/O files

Find which of the above information can be shared by T_1 and T_2.

(a) 1, 2, 3 (b) 1, 2, 4
(c) 1, 3, 4 (d) 2, 3, 4

■■■■

Answers Process and Threads

1. (a) 2. (d) 3. (a) 4. (b) 5. (c) 6. (b) 7. (c) 8. (a) 9. (b)
10. (b) 11. (c)

Explanations Process and Threads

3. (a)
CPU bound processes requires lot of processor time, resulting in long wait for I/O bound processes for the processor. This effect is called convey effect. It results in lower CPU and I/O devices utilization.

6. (b)
When system contains '*n*' processor and '*n*' processes, then maximum number of processes in running state can be '*n*' with each processor containing maximum of one process in the running state. The minimum number is zero with no processor having a process in running state.
Hence correct option (b).

7. (c)
When a process issues an input/output request then it goes from running state to blocked state.
When a process terminates itself it goes from running state to terminate state.
A process cannot go to running state after completing its I/O, it most go to ready state.
Hence option (c) is correct.

9. (b)
Kernel level threads can be scheduled independently. For user level threads a system call can block the entire process and are not transparent to Kernel.

10. (b)
Process A has user-level threads. Whole process has single control block instead of maintaining control block for each thread. So blocking one thread cause all processes to block. Here process *A* and process *B* are independent, hence no relation between *A* and *B*.
∴ Option (b) is correct.

11. (c)
Each thread needs a program counter and stack section to keep the local variables of procedures. So stack section can not be shared by threads.
∴ option (c) is correct.

■■■■

3 CHAPTER
CPU Scheduling

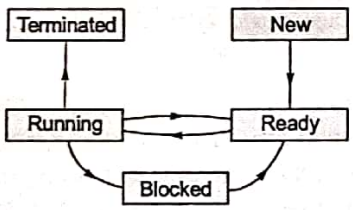
Q.1 In round-robin scheduling there are 'n' no. of processes in ready queue and time slice is 'q' units in worst case, the interrupted process will get the CPU again after

(a) $(n-1)q$ units (b) nq units
(c) $(q-1)n$ units (d) $(q+1)n$ units

Q.2 Consider 'n' processes sharing the CPU in a round-robin fashion. Assume that the context switch takes 'S' seconds. What must be the quantum 'q' such that the overhead of context switching is minimized and at the same time each process is guaranteed to execute on the CPU atleast once in every t seconds?

(a) $q \le (t - ns)/(n-1)$
(b) $q \le (t - ns)/(n-1)$
(c) $q \le (t - ns)/(n+1)$
(d) $q \le (t - ns)/(n+1)$

Q.3 The process state transition diagram in the figure is representative of



(a) a batch operating system
(b) an operating system with a preemptive scheduler
(c) an operating system with a non-preemptive scheduler
(d) a uni-programmed operating system

Q.4 Consider the following set of processes that arrive at time 0, with the length of the CPU-burst time given in milliseconds:

Process	Burst time
P_1	4
P_2	7
P_3	2

What is the average waiting time in milliseconds when we use the FCFS, and SJF scheduling algorithms?

(a) 9.33, 4.33 (b) 5.0, 4.33
(c) 5.0, 2.6 (d) 2.6, 2.6

Q.5 A uniprocessor computer system only has two processes, both of which alternate 10ms CPU bursts with 90ms I/O bursts. Both the processes were created at nearly the same time. The I/O of both processes can proceed in parallel. Which of the following scheduling strategies will result in the least CPU utilization (over along period of time) for this system?

(a) first come first serve scheduling
(b) shortest remaining time first scheduling
(c) static priority scheduling with different priorities for the two processes
(d) round robin scheduling with a time quantum of 5 ms

Q.6 Consider the following set of processes that need to be scheduled on a single CPU. All the times are given in msec.

Process Name	Arrival Time	Burst Time
A	0	3
B	3	4
C	7	3
D	9	5
E	12	7

Which of the following will give minimum average waiting time for all five processes?

(a) FCFS only (b) RR ($T.Q.$ = 3) only
(c) SRTF only (d) Both (a) and (c)

Q.7 Starvation can be avoided by which of the following statements.

1. By using shortest job first resource allocation policy.
2. By using first-come, first serve resource allocation policy.

(a) 1 only (b) 2 only
(c) 1 and 2 only (d) None of these

Q.8 Suppose a system contains n processes and system uses the round robin algorithm for CPU scheduling then which data structure is best suited ready queue of the processes.

(a) stack (b) queue
(c) circular queue (d) tree

Q.9 If a system contains CPU bound processes then which of the following scheduling algorithm produces maximum efficiency of the CPU.

(a) FIFO (b) round robin
(c) SJF (d) priority

Q.10 The jobs are assumed to have arrived at time 0 and in the order *p*, *q*, *r*, *s*, *t*. Calculate the departure time for job *p* if scheduling is round-robin with time slice of 1.

Job ID	CPU Burst Time
p	4
q	1
r	8
s	1
t	2

(a) 16 (b) 10
(c) 11 (d) 12

Q.11 Below process arrive in the order P_1, P_2, P_3 and are served in FCFS order. The average turn around time is _____ and average waiting time is _____

Process	CPU Burst time	Arrive time
P_1	5	0
P_2	3	1
P_3	4	2
P_4	1	3

(a) 4.5 millisec, 6.25 millisec
(b) 3.5 millisec, 4.75 millisec
(c) 8 millisec, 4.75 millisec
(d) 4.75 millisec, 8 millisec

Q.12 Consider the following performance table for FCFS scheduling. For this batch of processes, the throughout will be _____.

Position in batch	Job arrival time (*Ai*)	Job completion time (*Ci*)
1	3	8
2	8	10
3	10	16
4	12	18
5	15	29

(a) 0.184 (b) 0.192
(c) 0.269 (d) 0.238

Q.13 Consider we have four processes P_1, P_2, P_3 and P_4 and consider the following table. Consider the $a < b < c < d$ where *a*, *b*, *c* and *d* is the execution time of P_1, P_2, P_3 and P_4 respectively.

Processes	Execution Time
P_1	*a*
P_2	*b*
P_3	*c*
P_4	*d*

What is the average turn around time when shortest job first CPU scheduling algorithm is used?

(a) $\frac{(3a+4b+c+d)}{4}$ (b) $\frac{(4a+3b+2c+d)}{4}$

(c) $\frac{(a+2b+3c+4d)}{4}$ (d) None of these

Q.14 Pre-emptive scheduling, is the strategy of temporarily suspending a running process

(a) before the CPU time slice expires
(b) to allow starving processes to run
(c) when it requests I/O
(d) None of these

Q.15 In which of the following scheduling policies does context switching never take place?

1. Round-robin
2. Shortest job first (non pre-emptive)
3. Pre-emptive
4. First-cum-first-serve

(a) 4 only (b) 1 and 3
(c) 2 and 3 (d) 2 and 4

Q.16 'Aging' is
(a) keeping track of cache contents
(b) keeping track of what pages are currently residing in the memory
(c) keeping track of how many times a given page is referenced
(d) increasing the priority of jobs to ensure termination in a finite time

Q.17 Multilevel feedback queue scheduling
(a) allows to select a process and load it to memory for execution
(b) allows to select a process, that are ready to execute and allows CPU to execute one of them
(c) does not allow a process to move between queues
(d) allows processes which are permanently assigned to a queue on the entry to the system

Q.18 Consider following set of process with the CPU burst time given in milliseconds, arrival time and priority.

Process	Arrival	Burst time	Priority
A	0	8	4
B	1	4	3
C	2	6	1
D	3	1	2

Average turn around time using preemptive is _____ and using non-preemptive is _____.
(a) 12.5 ms, 10.5 ms
(b) 11.5 ms, 13.5 ms
(c) 10.5 ms, 12.5 ms
(d) 13.5 ms, 11.5 ms

Q.19 The sequence _____ is a non-preemptive scheduling sequence for the following jobs which leaves the CPU idle for _____ unit(s) of time

Job	Arrival	Burst Time
1	0.0	9
2	0.6	5
3	1.0	1

(a) {3, 2, 1), 1 (b) {2, 1, 3}, 0
(c) {3, 2, 1), 0 (d) {1, 2, 3), 5

Q.20 Suppose there are five processes in the ready queue as shown below:

i	$T(Pi)$	Priority
0	350	5
1	125	2
2	475	3
3	250	1
4	75	4

If FCFS, SJF and priority scheduling algorithms are used, for which algorithm is the average time minimum? Assume lower integer indicates higher priority. Assume all arrives at time zero.
(a) FCFS (b) SJF
(c) Priority scheduling (d) None of these

Q.21 Consider three CPU-intensive processes, which require 10, 20 and 30 time units and arrive at times 0, 2, and 6, respectively. How many context switches are needed if the operating system implements a shortest remaining time first scheduling algorithm? Do not count the context switches at time zero and at the end
(a) 1 (b) 2
(c) 3 (d) 4

Q.22 Which of the following is the solution of priority inversion problem?
(a) Kill the higher-priority process
(b) Kill the lower-priority process
(c) Priority-Inheritance protocol
(d) Semaphore

Q.23 Suppose a system uses shortest job first scheduling and exponential average of the measured length of previous CPU burst is 0.25. If the initial value of the predicted CPU burst time is 4 unit. The predicted time for 4th CPU burst for a process with burst time of 4 unit, 12 unit and 8 unit respectively (units) is _____.

Q.24 Consider four process all are arriving at time zero, with total execution time of 20, 10, 10 and 20 unit respectively. Each process spends the first 20% of execution time doing CPU, the next 60% of doing I/O computation and the last 20% of time doing CPU computation again. The operating

system uses longest time first scheduling algorithm and schedules a new process either when running process get blocked I/O or when the running process finishes its CPU burst.

Assume that I/O operations can be overlapped as much as possible. The average TAT of the system given by _____ unit.

[*Note:* When same burst occurs for multiple process high priority given to lowest process id] (upto one decimal place)

Q.25 Consider 4 processes sharing the CPU in a round robin fashion. Assuming that context switching takes 5 seconds. What must be the maximum quantum size P, such that the overhead resulting from process switching is minimized but at same time each process is guaranteed to gets its turn at CPU at-least every 40 seconds _____ (upto 2 decimal place).

Q.26 Consider all the processes are arriving at large time intervals. Let t be the time interval between two processes P_i and P_{i+1} for any i and service time of P_i is s_i. If $t > s_i$ for every i then find the strategy to schedule the processes.

(a) FCFS (b) SJN
(c) RR (d) SRTF

Q.27 Consider the following pre-emptive priority scheduling algorithm based on dynamically changing priorities. Larger priority numbers imply higher priority. When a process is waiting for the CPU (in the ready queue but not running), its priority changes at a rate 'X'. When it is running, it's priority changes at a rate Y. All processes are given a priority of '0'. When they enter the ready queue, the parameters can be set to give many different scheduling algorithms.

What is the algorithm that results from $Y > X > 0$?

(a) Last in First out (b) First come first serve
(c) Round robin (d) None of these

Q.28 Consider the below table where processes and respective times

P. No.	A.T.	CPU Time	I/O Time	CPU Time
1	0	5	5	2
2	3	2	22	2
3	7	8	0	0
4	25	9	2	1

By using SRTF algorithm find the completion times of P_1, P_2, P_3 and P_4 respectively.

Note:

1. Process first performs CPU operation followed by I/O operation and followed by CPU operation again.
2. Multiple processes will perform I/O operations simultaneously.

(a) 12, 29, 15, 39 (b) 12, 31, 17, 38
(c) 12, 31, 17, 39 (d) None of these

Q.29 Consider 3 processes P_0, P_1 and P_2 to be scheduled as per the SRTF algorithm. The process 'P_0' is known to be scheduled first and when 'P_0 is running '5' units of time, the process 'P_2' has arrived. The process 'P_2' has own '2' unit of time, the process 'P_1' has arrived and completed running in '4' units of time. Then the minimum burst time of 'P_0' is _____ (in units).

Q.30 Consider a preemptive SRTF scheduling technique followed by three processes P1, P2, P3. All of these 3 processes arrive at time $t = 0$ and their total execution time is 10, 20, 30 units respectively. Each process spends first 10% of execution doing I/O, 70% CPU and rest doing I/O operation.

What will be the CPU idle % time to execute all processes? (Assume a uniprocessor and all I/O operations can be overlapped)

(a) 9.160 (b) 10.630
(c) 14.286 (d) CPU will not be idle

■■■■

Answers CPU Scheduling

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

Explanations CPU Scheduling

2. (b)

In Round-robin scheduling, every one get quantum time for executing

Number of processes = n

Each process takes S seconds, so total time taken by processes = nS

For process switching is minimized

t = total time

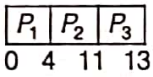
For quantum size

$$q >= \frac{(t - \text{total time taken by processes})}{\text{Number of process} - 1}$$

the switching is minimum.

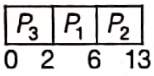
4. (c)

Gantt chart of FCFS



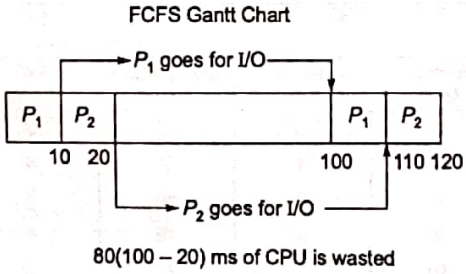
$$\text{Avg. waiting time} = \frac{0+4+11}{3} = 5$$

Gantt chart of SJF



$$\text{Avg. waiting time} = \frac{0+2+6}{3} = 2.6$$

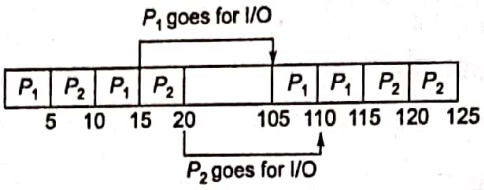
5. (d)



Same gantt chart will be for SRTF.

Priority scheduling results in same amount of time waste.

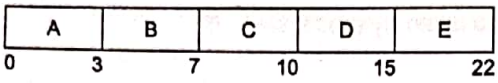
But in round robin



In RR 85 (105 – 20) ms of CPU is wasted. Also note that at time 110 P_1 continue to avoid process switch overhead since P_2 is just arrived.

6. (a)

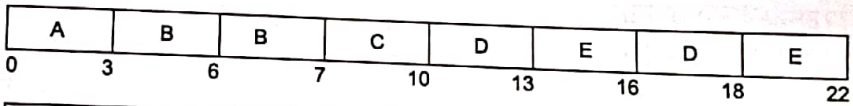
FCFS:



Process Name	Arrival Time	Burst Time	Completion Time	TAT Time	Waiting Time
A	0	3	3	3	0
B	3	4	7	4	0
C	7	3	10	3	0
D	9	5	15	6	1
E	12	7	22	10	3

4/5 = 0.8

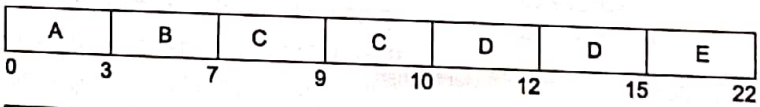
RR (T.Q. = 3):



Process Name	Arrival Time	Burst Time	Completion Time	TAT Time	Waiting Time
A	0	3	3	3	0
B	3	4	7	4	0
C	7	3	10	3	0
D	9	5	18	9	4
E	10	7	22	12	5

9/5 = 1.8

SRTF:



Process Name	Arrival Time	Burst Time	Completion Time	TAT Time	Waiting Time
A	0	3	3	3	0
B	3	4	7	4	0
C	7	3	10	3	0
D	9	5	15	6	1
E	10	7	22	12	5

6/5 = 1.2

Since FCFS time < SRTF time < RR time.

7. (b)

Starvation can be avoided either by using round-robin policy or FCFS policy, By using SJF starvation is not avoided.

Hence option (b) is correct.

8. (c)

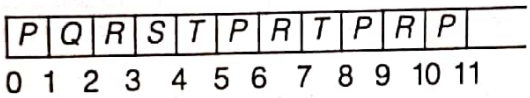
In round robin policy each process has allotted fix time quantam, after its time quantam is over it goes to tail of the ready queue if not completed.

hence it act as a circular queue implementation.

10. (c)

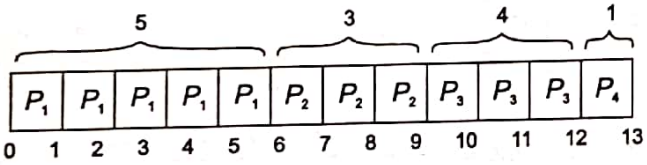
Round Robin Policy

Gantt Chart



Hence option (c), 11 is correct.

11. (c)



$$\text{Average turnaround time} = \frac{(5-0)+(8-1)+(12-2)+(13-3)}{4} = 8 \text{ milli seconds}$$

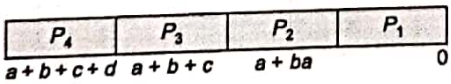
$$\text{Average waiting time} = \frac{0+(5-1)+(8-2)+(12-3)}{4} = 4.75 \text{ m.sec}$$

12. (b)

$$\text{Through put} = \frac{\text{No. of processes}}{\text{Time duration}} = \frac{5}{29-3} = 0.1923$$

13. (b)

Given execution time ($a < b < c < d$) arrive at zero time and then by SJF scheduling



$$\text{turn around time} = \frac{(a-0)+(a+b-0)+(a+b+c-0)+(a+b+c+d-0)}{4}$$

$$= \frac{(4a+3b+2c+d)}{4}$$

14. (d)

All the 3 options (a), (b) and (c) are pre-emptive scheduling.

15. (d)

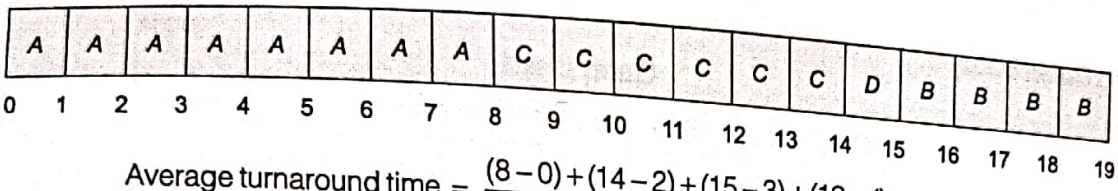
Context switching takes place when a process is preempted (forcefully) and another process goes into the running state. In FIFO and SJF (non-preemptive) techniques, the processes finishes their execution then only their context is switched to other processes.

16. (d)

Aging is logically the process of assigning a high priority to a low priority process to ensure that it does not lead to endless starvation.

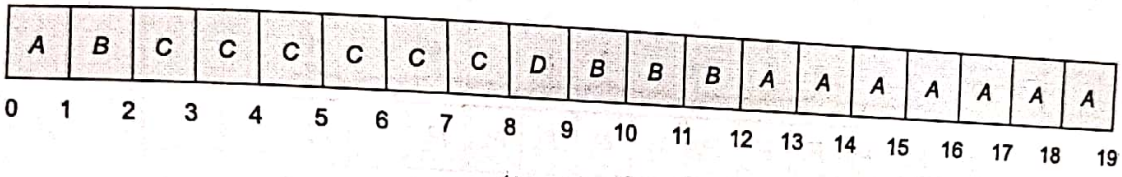
18. (c)

For non preemptive



$$\text{Average turnaround time} = \frac{(8-0)+(14-2)+(15-3)+(19-1)}{4} = 12.5 \text{ ms.}$$

For preemptive



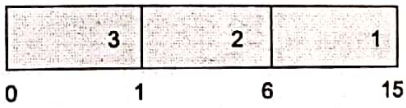
$$\text{Average turnaround time} = \frac{(19-0)+(12-1)+(8-2)+(9-3)}{4} = 10.5 \text{ ms}$$

Hence (c) is the correct option.

19. (a)

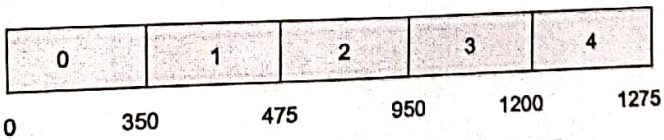
We can observe clearly that if sequence (3, 2, 1) is executed in the non-preemptive manner then it leaves the CPU idle for 1 unit of time.

Hence option (a) is correct



20. (b)

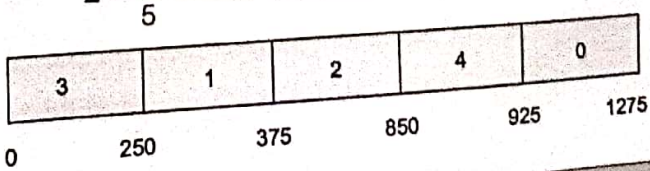
FCFS



$$\text{Average turnaround time} = \frac{350+475+950+1200+1275}{5}$$

$$= \frac{4250}{5} = 850 \text{ time units}$$

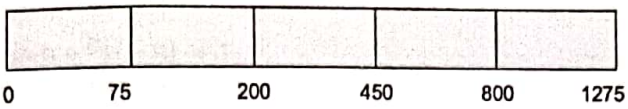
Priority Scheduling



$$\text{Average turnaround time} = \frac{250+375+850+925+1275}{5}$$

$$= \frac{3675}{5} = 735 \text{ time units}$$

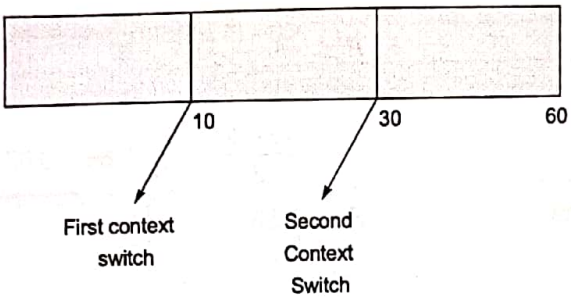
SJF Scheduling



$$\text{Average turnaround time} = \frac{75+200+450+800+1275}{5} = 560 \text{ time units}$$

Hence (b) is the correct option.

21. (b)



Hence only two context switch occurs.

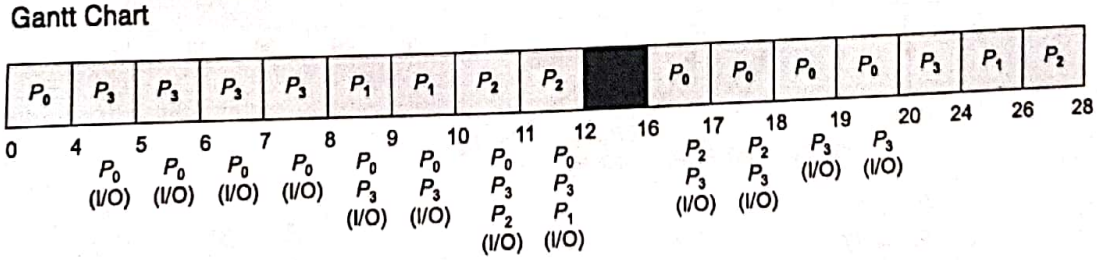
23. (6.5)

$$\tau_{n+1} = \alpha \times T_n + (1-\alpha)\tau_n$$
$$\tau_2 = 0.25 \times 4 + 0.75 \times 4 = 4 \text{ Unit}$$
$$\tau_3 = 0.25 \times 12 + 0.75 \times 4 = 6 \text{ Unit}$$
$$\tau_4 = 0.25 \times 8 + 0.75 \times 6 = 6.5 \text{ Unit}$$

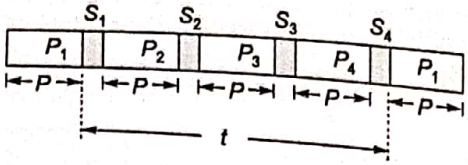
24. (24.5)

Process	Burst Time	CPU	I/O	CPU	CT	TAT
P_0	20	4	12	4	20	20
P_1	10	2	6	2	26	26
P_2	10	2	6	2	28	28
P_3	20	4	12	4	24	24
					Average TAT = 98/4 = 24.5	

Gantt Chart



25. (6.67)



$$t = (n-1)P + n \times s$$

$$t - ns \geq (n-1)P$$

$$\frac{t-ns}{n-1} \geq P$$

t = 40 seconds
n = 4process
s = 5 second

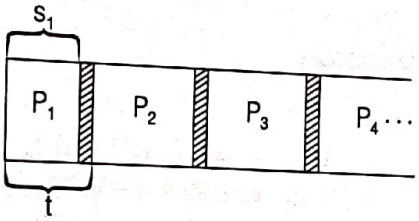
$$P \leq \frac{40-(4\times 5)}{4-1}$$

$$= \frac{40-20}{3} = \frac{20}{3}\ 6.66 = 6.67$$

Maximum value of P when

P = 6.66

26. (a)

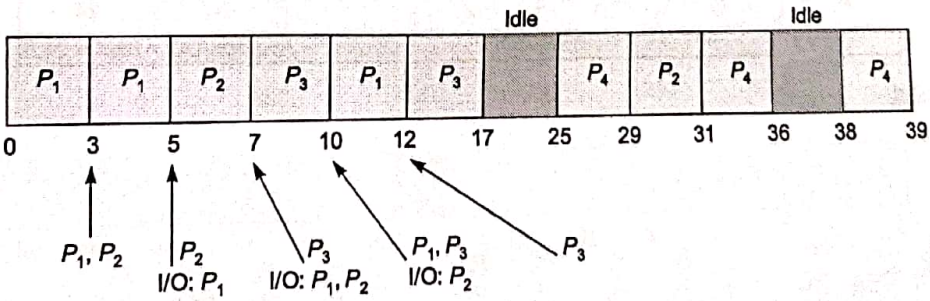


If $t > s_i$ then all processes are serviced in FCFS based.
∴ Option (a) is correct.

27. (b)

It results in FCFS.

28. (c)



$P_1 \rightarrow 12$
$P_3 \rightarrow 17$

$P_2 \rightarrow 31$
$P_4 \rightarrow 39$

29. (12)

First process 'P_0' schedule and run for = 5 units

Process 'P_0' pre-empted.

Process 'P_2' has arrived and run for = 2 unit

Process 'P_2' pre-empted.

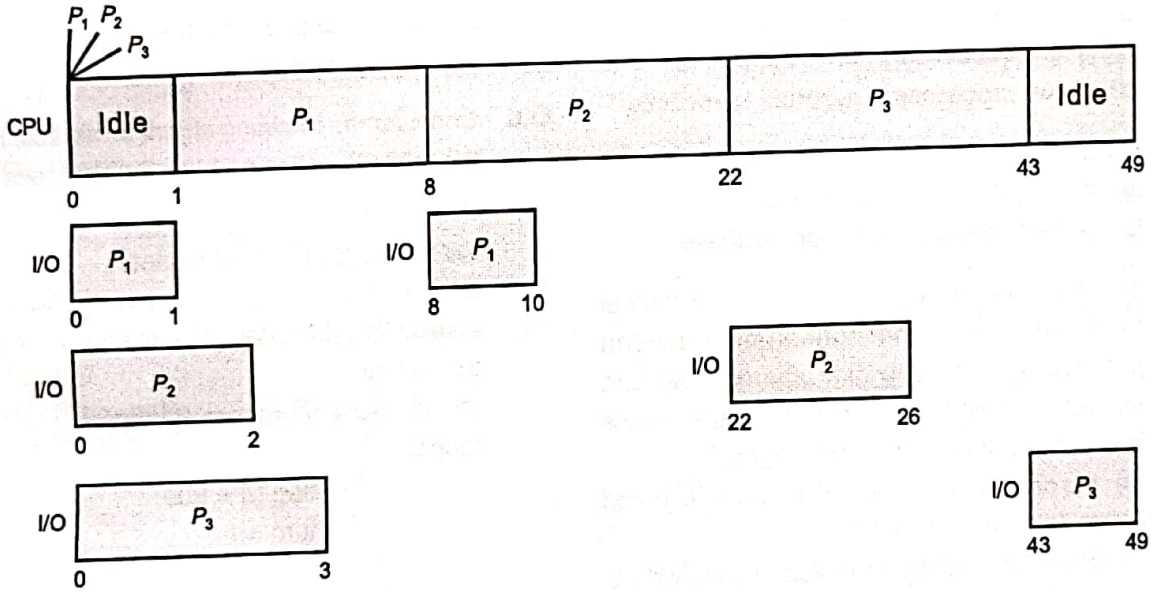
Process 'P_1' has arrived and run for = 4 unit

So the minimum burst time of process 'P_0' is = 5 + (2 + 3) + 1 ← Remaining. = 12 units

30. (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

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

■■■■

4 CHAPTER

Process Synchornization

Q.1 Spooling helps because
(a) it is a more secure method of accessing data
(b) print jobs go more smoothly with less stop and go
(c) the computer is released to do other things while still printing
(d) none of the above

Q.2 Interprocess communication
(a) is required for all processes
(b) is usually done via disk drives
(c) is never necessary
(d) allows processes to synchronize activity

Q.3 Producer consumer problem can be solved using
(a) semaphores (b) monitors
(c) both (a) and (b) (d) none of these

Q.4 Semaphores can be used to enforce mutual exclusion and synchronization between processes interacting over shared data and variables. Which of the following statements is true about semaphores in this regard?
(a) the operations SIGNAL(S) and WAIT(S) need to be atomic
(b) a process exiting the critical section will call SIGNAL(S) which will WAKEUP( ) a blocked process awaiting entry to the critical section
(c) 'busy-wait' solutions to the critical section are typically implemented using machine instructions that execute in the kernel mode
(d) all of the above

Q.5 One needs a good practical example of a "producer-consumer" type problem from among the following.
(i) a multithreaded audio server, where a thread writes digitized sound into a circular buffer and another thread reads it over to a web page.
(ii) a printer spooler where files are spooled on to the printer queue, to be taken and printed by the printer driver.
(iii) a shared data base updated by simultaneous queries.

Which of the following is/are true of such a good practical example?
(a) Only (i) is true
(b) Only (ii) is true
(c) Only (i) and (iii) are true
(d) Only (i) and (ii)

Q.6 Consider the following algorithm as a solution to the critical-section problem. The process share two variables:

Var flag: array [0...1] of boolean;
turn; 0...1;
Initially flag [0] = flag [1] = false;
turn = 0 or 1.
the structure of process P_i is like:

```
repeat
        flag [i] = true;
        turn = j
        while (CONDITION)
        do no-op;
        critical section
        flag [i] = false
        remainder section
until false;
```

What should be the CONDITION in the above algorithm so that it will meet all three requirement (Mutual Exclusion, Progress, and Bounded Waiting)?
(a) flag [j] = true and turn = j;
(b) flag [j] = false and turn = i;
(c) flag [i] = true and turn = j;
(d) flag [i] = false and turn = i;

Q.7 A solution to the Dinning philosophers problem which avoids deadlock is
(a) ensure that all philosophers pick up the left fork before the right fork
(b) ensure that all philosophers pick up the right before the left fork
(c) ensure that all odd numbered philosophers pick-up the left fork and right numbered philosopher pick-up the right fork
(d) none of the above

Q.8 Let $m[0]$ .... $m[4]$ be mutexex (binary semaphores) and $P[0]$ .... $P[4]$ be processes. Suppose each process $P[i]$ executes the following:

```
wait (m[i]);
wait (m[(i + 1) mod4]);
.....
signal (m[i]);
signal (m[(i + 1) mod 4]);
```

This could cause
(a) thrashing
(b) deadlock
(c) starvation but not deadlock
(d) none of the above

Q.9 Suppose we want to synchronize two concurrent processes P and Q using binary semaphores S, T and U.

Process P:	Process Q:
$P(S)$	W:
$P(T)$	X:
$P(U)$	Y:
print 'a';	print 'a';
print 'b';	print 'b';
$V(S)$	A:
$V(T)$	B:
$V(U)$	C:

What statements should be written at W if no deadlock should occur?
(a) $P(T)$
(b) $P(U)$
(c) $P(S)$
(d) not possible to avoid deadlock

Q.10 Suppose we want to synchronize two concurrent processes P and Q using binary semaphores S and T. The code of the processes P and Q is shown below.

Process P	Process Q
while (1) {	while (1) {
W;	Y;
print '0';	print '1';
print '0';	print '1';
X;	Z;
}	}

Synchronization statements can be inserted only at points W, X, Y and Z.

Which of the following will always lead to an output starting with '001100110011'?
(a) $P(S)$ at W, $V(S)$ at X, $P(T)$ at Y, $V(T)$ at Z, S and T initially 1
(b) $P(S)$ at W, $V(T)$ at X, $P(T)$ at Y, $V(S)$ at Z, S initially 1, and T initially 0
(c) $P(S)$ at W, $V(T)$ at X, $P(T)$ at Y, $V(S)$ at Z, S and T initially 1
(d) $P(S)$ at W, $V(S)$ at X, $P(T)$ at Y, $V(T)$ at Z. S initially 1 and T initially 0

Q.11 Critical region is
(a) a part of the operating system which is not allowed to be accessed by any process
(b) a set of instructions that access common shared resource which exclude one another in time
(c) the portion of the main memory which can be accessed only by one process at a time
(d) none of these

Q.12 At a particular time, the value of a counting semaphore is 10. It will become 7 after
1. 3 V operations
2. 3 P operations
3. 5 V operations and 2 P operations
4. 13 P operations and 10 V operations

(a) 1 and 2 (b) 2 and 3
(c) 2 and 4 (d) 4 only

Q.13 Semaphores are used to solve the problem of
1. Race condition
2. Process synchronization
3. Mutual exclusion
4. None of the above

(a) 1 and 2 (b) 2 and 3
(c) All of the above (d) None of the above

Q.14 At a particular time of computation, the value of a counting semaphore is 7. Then 20 *P* operation and '*x*' *V* operation were completed on this semaphore. If the final value of the semaphore is 5, *x* will be

(a) 15 (b) 22
(c) 18 (d) 13

Q.15 Each process *Pi*, *i* = 1, 2, 3,...., 9 is coded as follows:

```
repeat
P (mutex)
{ critical section }
V (mutex)
forever
```

The code for P10 is identical except that it uses V (mutex) instead of P (mutex). What is the largest number of processes that can be inside the critical section at any moment?

(a) 1 (b) 2
(c) 3 (d) None of these

Q.16 Process P_1 and P_2 have a producer-consumer relationship, communicating by the use of a set of shared buffers:

P_1: repeat	P_2: repeat
obtain an empty buffer fill it	obtain a full buffer empty it
return a full buffer forever	return an empty buffer forever

Increasing the number of buffers is likely to do which of the following?

1. Increase the rate at which requests are satisfied.
2. Decrease the likelihood of deadlock.
3. Increase the case of achieving a correct implementation.

(a) 1 only (b) 2 only
(c) 2 and 3 only (d) 1, 2 and 3

Q.17 Consider a queue between the two processes indicated below. *N* is the length of queue and *e*, *f* and *b* are semaphore.

init : *e* : = *N*; *f* = 0; *b* : = 1;

Process 1:	Process 2:
loop	loop
p(*e*)	*p*(*f*)
p(*b*)	*p*(*b*)
enqueue	dequeue
V(*b*)	*V*(*b*)
V(*f*)	*V*(*e*)
end loop	end loop

Which of the following statements is/are true?

1. The purpose of semaphore *f* is to ensure that dequeue is not executed on an empty queue.
2. The purpose of semaphore *e* is to ensure that deadlock does not occur.
3. The purpose of semaphore *b* is to provide mutual exclusion for queue operations.

(a) 1 only (b) 2 only
(c) 1 and 3 only (d) 2 and 3 only

Q.18 Consider the following:

```
int numreader = 0;
mutex = semaphore (1);
roomempty (1);
Reader Code          Writer Coder
r1: mutex.wait( );   w1: roomempty.wait;
r2: numreader+=1;    w2: /*Critical Writer Section*/
r3: if(numreader == 1) w3: roomempty.signal();
r4: roomempty.wait( );
r5: mutex.signal( );
r6: /*Critical Reader Section*/
r7: mutex.wait( );
r8: numreader –= 1;
r9: if(numreader == 0)
r10: roomempty.signal( );
r11: mutex.signal( );
```

The above code is a proposed solution of the Readers/Writer problem. Consider which of the following statements is/are true?

1. The code allow several writers to execute in the writers critical area at the same time
2. The code allow several readers to execute in the readers critical area at the same time.

(a) 1 only (b) 2 only
(c) 1 and 2 both (d) none of these

Q.19 Processes P1 and P2 use critical_flag in the following routine to achieve mutual exclusion. Assume that critical_flag is initialized to FALSE in the main program.

```
get_exclusive_access ()
{   if (critical_flag = = FALSE) {
        critical_flag = TRUE ;
        critical_region ( ) ;
        critical_flag = FALSE ;
    }
}
```

Consider the following statements:

1. It is possible for both P1 and P2 to access critical_region concurrently.
2. This may lead to a deadlock.

Which of the following holds?

(a) 1 is false and 2 is true
(b) Both 1 and 2 are false
(c) 1 is true and 2 is false
(d) Both 1 and 2 are true

Q.20 The following is a code with two threads, producer and consumer, that can run parallel. Further, S and Q are binary semaphore equipped with the standards P and V operations.

semaphore S = 1, Q = 0
integer x;

producer :	consumer
while (true) do	while (true) do
P(S);	P(Q);
x = produce ();	consume (x);
V(Q);	V(S);
done	done

Which of the following is TRUE about the program above?

(a) The process can deadlock
(b) One of the threads can starve
(c) Some of the items produced by the producer may be lost
(d) Values generated and stored in 'x' by the producer will always be consumed before the producer can generate a new value

Q.21 Let R_1, R_2, R_3 be reader processes and let W_1 and W_2 be writer processes requesting shared data. If R_1 is selected for access. Which of the statement is/are correct?

1. Mutual exclusion is necessary for R_2 and R_3
2. No mutual exclusion is necessary for R_2 and R_3
3. Mutual exclusion must be there for W_1 and W_2
4. No mutual exclusion must be there for W_1 and W_2

(a) 1 and 2 only (b) 2 and 3 only
(c) 3 and 4 only (d) 1 and 4 only

Q.22 Assume that 'C' is counting semaphore. Consider the following program segment:

```
C = 10;
P (C);
V (C);
P (C);
P (C);
P (C);
V (C);
P (C);
P (C);
```

What is the value of 'C' at the end of the above program execution?

(a) 2 (b) 4
(c) 6 (d) 10

Q.23 Consider the code used by the processes P and Q for accessing their critical sections. The initial values of shared Boolean variables S and T are false.

P	Q
while (S == T);	while (S !=T);
<c.s.>	<c.s.>
S = !(T);	S = T;

Which of the following statements is true?

(a) Code will violate the mutual exclusion.
(b) Process P can go into the critical section multiple times without single entry of Q into the critical section.
(c) Process Q can go into critical section after exactly one entry by process P into its critical section.
(d) None of these

Q.24 Match the following List:

List-I

A. Synchronization
B. Mutual exclusion
C. Critical Section

List-II

1. Piece of code that only one process can execute at once
2. Ensuring that only one process does a particular thing at a time.
3. Using atomic operations to ensure cooperation between processes.

Codes:

	A	B	C
(a)	1	2	3
(b)	1	3	2
(c)	3	2	1
(d)	3	1	2

Q.25 Consider the following algorithm with 2-processes (P_0 and P_1) to solve the critical problem. (i = 0 or 1)

```
flag[i] = false;
while (true){
    flag[i]=true;
    while(flag[1–i]){
        flag[i]=false;
        while (flag[1–i]){
            no-op;
        }
        flag[i]=true;
    }
    <critical section>
    flag[i]=false;
    remainder section
}
```

If P_0 and P_1 processes execute concurrently then which of the following can be guaranteed by the above algorithm.

(a) Mutual exclusion
(b) Mutual exclusion and progress
(c) Mutual exclusion and Bounded wait
(d) Mutual exclusion, progress and bounded wait

Q.26 Let U and V be two counting semaphore variables accessed by following three concurrent processes.

P_1	P_2	P_3
L_1 : wait(U)	L_2 : wait(V)	L_3 : wait(V)
print "C"	print "A"	print "D"
signal(V)	signal(V)	goto L_3
goto L_1	goto L_2	

Assume initial values of U and V are 3 and 0 respectively. Find the maximum number of C's that can be printed when the above processes are executed.

(a) 4 (b) 3
(c) 2 (d) 1

Q.27 X, Y and Z are shared semaphores. The following 4 pseudo-coded threads are started

$X = 0$
$Y = 1$
$Z = -1$

Thread1	Thread2	Thread3	Thread4
Wait(X)	Wait(Z)	Wait(Y)	Wait(X)
Print"1"	Print"2"	Print"3"	wait(Y)
Signal(Z)	Signal(Z)	Signal(X)	print"4"
Signal(Y)	Signal(Y)	Signal(X)	Signal(X)

What is output:

(a) 314 (b) 124
(c) 123 (d) 432

Q.28 Consider the below program to synchronize the two processes P_0 and P_1.

```
shared int turn = 1;
```

```
int mypid = 0; // for P0 it is '0', for P1 it is '1'
int otherpid = 1 – mypid;
```

```
while (turn != mypid)
Donothing( );
```

```
C.S.
```

```
turn = otherpid;
```

Which of the following is satisfied?

(a) M.E. but not progress
(b) Progress but not M.E.
(c) Both progress and M.E.
(d) Neither M.E. nor progress

Q.29 Consider the following 'C' code

```
x = 0;
y = 1;      } initialization
z = true;
```

P_1	P_2
while ($x < y$)	while (z)
{$x = x + 1$;}	{$y = y + 1$; $z = (x != y)$;}

Assume P_1 and P_2 are two concurrent processes and sharing the global variables x, y and z. Assignments and tests are atomic. Find which of the following is not possible with the parallel execution?

(a) P_1 and P_2 terminates
(b) P_1 terminates but P_2 does not
(c) P_2 terminates but P_1 does not
(d) Neither P_1 nor P_2 terminates

Q.30 Consider the following code for P_1 and P_2 processes.

P_1	P_2
while (true){	while (true){
n = random();	P(C);
for(i = 0; $i < n$; i++)	printf("E");
printf("M");	V(S);
for (i =0; $i < n$; i++)	}
V(C);	
for (i =0; $i < n$; i++)	
P(S);	
}	

There are two semaphores C and S. The initial value of both semaphores is zero. Assume random ( ) is a function which can return any positive integer value. Which of the following output prefix is not possible to generate?

(a) MEMEMEMEME
(b) MMMMMMEEEE
(c) MMMEEEEMME
(d) MMMEEEMMEE

Q.31 Which of the following is the guaranteed solution of avoidance of mutual exclusion?

(a) Semaphore
(b) Monitors
(c) Banker's algorithm
(d) None of these

■■■■

Answers Process Synchornization

1.	(c)	2.	(d)	3.	(c)	4.	(d)	5.	(d)	6.	(a)	7.	(c)	8.	(b)	9.	(c)
10.	(b)	11.	(b)	12.	(c)	13.	(b)	14.	(c)	15.	(c)	16.	(a)	17.	(c)	18.	(b)
19.	(c)	20.	(d)	21.	(b)	22.	(c)	23.	(c)	24.	(c)	25.	(c)	26.	(b)	27.	(a)
28.	(a)	29.	(c)	30.	(c)	31.	(b)										

Explanations Process Synchornization

8. (b)

This could cause deadlock when each process executes their 1st line and trying to execute their 2nd line.

9. (c)

If we write $P(T)$ then X or Y should have $P(S)$. in any case there is a chance that process P requires S and waits for T. And process Q acquires T and waits for S in which case the deadlock occurs. Similar is the case with $P(U)$. If it is $P(S)$ then only one of P or Q can acquire it and proceed further.

10. (b)

Process P:	Process Q:
while (1) {	while (1){
W : P(S)	Y : P(T)
print '0';	print '1';
print '0';	print '1';
X : V(T)	Z : V(S)
}	}

The initial value of S = 1 and T = 0, the scheme ensures that the process P executes first printing 00, followed by process Q printing 11, followed by process P and so on.

This happens due to initial values of S and T. The value of S initialized to 1 and that of T to 0 ensures that P(S) at W gets executed successfully, while P(T) at Y waits till V(T) at X is executed. The scheme generates the string 00 1100 11 00 11.

11. (b)

Critical region is a set of instructions that access common shared resource which exclude one another in time. Critical region is primarily centred on the usage of global shared resources (or variables) among the processes.

Hence option (b) is correct.

12. (c)

P-denotes wait operation and V denotes signal. Value of counting semaphore initially 10, it will become seven by decreasing it by '3' which can be done either by '3' wait operation or 13 'P'-operation and '10' V-operation, which effectively decreases the count by '3'.

13. (b)

Semaphores are used in deadlock avoidance by using them in interprocess communication. It is used to solve the problem of synchronisation among processes and achieves mutual exclusion.

14. (c)

Each P operation will decrease the semaphore value by 1 and V operation increases it by 1. If x is 18, then 7 P operations will make semaphore value 0. If this is followed by 7 V operations the value comes back to 7. So, after 18 P and 18 V operations, the value of the semaphore will be 7. The remaining 2 P operations result in the semaphore value 5.

15. (c)

Let the mutex be initialized to 1. Any one of the 9 processes P_i, $i = 1, 2, 3, \ldots 9$ can get into the critical section after executing P (mutex) which decrements the mutex value to 0. At this time P_{10} can enter into the critical section as it uses V (mutex) instead of P (mutex) to get into the critical section. As a result of this, mutex will be incremented by 1. Now any one of the

9 processes P_i, i = 1, 2, 3, ... 9 (excepting the one that is already inside the critical section) can get into the critical section after decrementing the mutex to 0. None of the remaining processes can get into the critical section.

If the mutex is initialized to 0, only 2 processes can get into the critical section. So the largest number of processes is 3.

16. (a)

Increasing the number of buffers may increase the rate at which the requests are satisfied. But it may not decrease the chances of deadlock. Because number of buffer will increase for both consumer and producer and thus may not increase the case of achieving a correct implementation.

17. (c)

Statement-1 is true as '*f*' semaphore is used to ensure that dequeue process is not executed on an empty queue.

Statement-2 is not true as semphore '*e*' is to ensure that enqueue process is not executed on a full queue.

Statement-3 is true as semaphore '*b*' with maximum value 1 provides mutual exclusion for queue operations.

18. (b)

The code allows several readers to execute in the readers critical area at the same time due to the numreader variable but it does not allow several writers to execute in the writers critical section at the same time.

19. (c)

Statement-1 is true as both processes can access critical region concurrently because of the condition "if(critical_flag==false)".

Both processes can execute this condition simultaneously and can enter critical section. But it cannot lead to deadlock.

20. (d)

According to the options, the process can't lead to deadlock, no threads can starve, no items produced by producer are last, but option (d) is correct.

Value generated and stored in '*x*' by the producer will always be consumed before the producer can generate a new value.

Because $P(s)$ will make $S = 0$ and then no producer can execute the code until consumer code is executed because it will execute $V(s)$.

21. (b)

Reader process don't need mutual exclusion. Writers process need mutual exclusion because write-write problem will be there.

23. (c)

Process *P* cannot go into critical section multiple times without the entry of *Q*.

24. (c)

(a) Synchronization : uses $P(\ )$ and $V(\ )$ operations (3)

(b) Mutual exclusion : Ensures only one process executes critical section at any time (2)

(c) Critical section : Sharing region (1)

25. (c)

Mutual exclusion is achieved. Progress is not achieved but bounded waiting is there. As soon as 1 process comes out of critical section and executes flag [*i*] = false, the other process waiting in while loop will come out of the loop and will enter critical section.

So, no starvation is there.

26. (b)

$U = 3$, can be decremented 3 times by P_1 execution first and prints "*C*" 3 times. If we observe in the given code there is no signal (U). When P_1 executes 4th time, it will be blocked.

$\therefore$ *C* is printed 3 times.

27. (a)

As value of x and $z \leq 0$.

So thread 1, 2 and 4 will be blocked on first attempt. Only Thread 3 can execute. After execution of thread 3 we have

$$x = 2,\ y = 0\ z = -1$$

So, only thread 1 is possible to execute.
After execution of thread 1 we have
$x = 1, y = 1, z = 0$
So only thread 4 can execute.
So result is 314.

28. (a)
Even other process is not interested in execution, A process executes the turn = otherpid after C.S. to give the chance to execute. If the same process is interested again and other is never interested then the interested process will never get the chance to execute.

29. (c)
Initially : $x = 0$, $y = 1$, b = true;
P_1: 1. while $(x = y)$
2. $\{x = x + 1\}$
P_2: 1. while (z)
2. $\{y = y + 1;$
3. $z = (x != y);\}$

(a) Both threads terminate : is possible.
P_2 : 1, 2
P_1 : 1, 2, 1, 2, 1 (terminated)
P_2 : 3, 1 (terminated)

(b) P_1 terminates but P_2 does not: is possible
P_1 : 1, 2, 1 (terminated)
P_2 : 1, 2, 3, 1, 2, 3 (infinite loop)

(c) P_2 terminates but P_1 does not : is not possible

(d) Neither P_1 nor P_2 terminates : is possible
P_2 : 1, 2, 3
P_1 : 1, 2

P_2: 1, 2, 3
P_1 1, 2
} Infinite iterations (P_2 followed by P_1)

Option (c) is correct.

30. (c)
Let $n = 3$
P_1 prints M three times
Each execution of V(C), P_1 preempts and P_2 executes one iteration to print 'E', so three times E prints but fourth E is not possible to print.
Option (c) is not possible to produce.

31. (b)
- Monitors allowed one process to execute in critical section at a time. It always satisfies the mutual exclusion properties.
- Improper use of semaphore does not lead to avoidance of mutual exclusion.
- Banker's algorithm is not related to mutual exclusion.

■■■■

5 CHAPTER
Concurrency and Deadlock

Q.1 Consider the following system state:

Process	Max	Allocated
P1	7	2
P2	6	2
P3	7	4

Total resources are 11. The system will be in a safe state if
(a) process P1 is allocated one additional resource
(b) process P2 is allocated two additional resources
(c) process P3 is allocated three additional resources
(d) process P2 is allocated one additional resource

Q.2 Fork system call creates the child process on the successful completion of fork, which of the following value is returned in the child process?
(a) pid of the parent process
(b) pid of the child process
(c) pid of the init process
(d) zero

Q.3 Minimum number of processes required for deadlock is
(a) 0 (b) 1
(c) 2 (d) None of these

Q.4 Which of the following is NOT a valid deadlock prevention scheme?
(a) release all resources before requesting a new resource
(b) number the resources uniquely and never request a lower numbered resource than the last one requested
(c) never request a resource after releasing any resource
(d) request all required resources be allocated before execution

Q.5 'm' processes share 'n' resources of the same type. The maximum need of each process doesn't exceed 'n' and the sum of all their maximum needs is always less than $m + n$. In this set up
(a) deadlock can never occur
(b) deadlock may occur
(c) deadlock has to occur
(d) none of the above

Q.6 In a certain operating system, deadlock prevention is attempted using the following scheme. Each process is assigned a unique time stamp, and is restarted with the same time stamp, when it was killed. Let P_n be the process holding a resource R, P_r be a process requesting for the same resource R and $T(P_n)$ and $T(P_r)$ be their time stamps respectively. The decision to wait or preempt one of the processes is based on the following algorithm.

if $T(P_r) < T(P_n)$ then
    kill P_r
else
wait

Which one of the following is TRUE?
(a) the scheme is deadlock free, but not starvation free
(b) the scheme is not deadlock free, but starvation free
(c) the scheme is neither deadlock free nor starvation free
(d) the scheme is both deadlock free and starvation free

Q.7 Dijkstra's banking algorithm in an operating system solves the problem of
(a) deadlock avoidance
(b) deadlock recovery
(c) mutual exclusion
(d) context switching

Q.8 Necessary conditions for deadlock are
(a) Non-preemption and circular wait
(b) Mutual exclusion and partial allocation
(c) Both (a) and (b)
(d) None of the above

Q.9 Which of the following statements is not true?
(a) Deadlock can never occur if resources can be shared by competing processes.
(b) Deadlock can never occur if resources must be requested in the same order by processes.
(c) The Banker's algorithm for avoiding deadlock requires knowing resource requirements in advance
(d) If the resource allocation graph depicts a cycle then deadlock has certainly occured.

Q.10 Suppose we have a system in which processes is in hold and wait condition then which of the following approach prevent the deadlock.
(a) Request all resources initially
(b) Spool everything
(c) Take resources away
(d) Order resources numerically

Q.11 An operating system makes use of Banker's algorithm to allocate 12 printers. Printers will be allocated to a process requesting them only if there are enough available printers to allow it to run to completion.

User 1 using 7 printers and will need at most a total of 10 printers.

User 2 is using 1 printer and will need at most 4 printers. User 3 is using 2 printers and will need at most 4 printers. Each user is currently requesting 1 more printer. Which of the following is true?
(a) The operating system will grant a printer to User 1 only
(b) The operating system will grant a printer to User 2 only
(c) The operating system will grant a printer to User 3 only
(d) The operating system will not grant any more printers until some are relinquished

Q.12 A state is safe if the system can allocate resources to each process (up to its maximum) in some order and still avoid deadlock, which of the following is/are true
1. Deadlocked state is unsafe.
2. Unsafe state may lead to a deadlock situation.
3. Unsafe state must lead to a deadlock situation.
4. Deadlock state is a subset of unsafe state

(a) 1, 2 and 3 (b) 1 and 2 only
(c) 1, 3 and 4 (d) 1, 2 and 4

Q.13 Which of the following is NOT true of deadlock prevention and deadlock avoidance schemes?
(a) In deadlock prevention, the request for resources is always granted if the resulting state is safe
(b) In deadlock avoidance, the request for resources is always granted if the resulting state is safe
(c) Deadlock avoidance is less restrictive than deadlock prevention
(d) Deadlock avoidance requires knowledge of resource requirements a priori

Q.14 An operating system implements a policy that requires a process to release all resources before making a request for another resource.
(a) Both starvation and deadlock can occur
(b) Starvation can occur but deadlock cannot occur
(c) Starvation cannot occur but deadlock can occur
(d) Neither starvation nor deadlock can occur

Q.15 An operating system contains 3 user processes each requiring 2 units of resource *R*. The minimum number of units of *R* such that no deadlock will ever occur is
(a) 3 (b) 4
(c) 5 (d) 6

Q.16 A computer system has 6 tape drives, with '*n*' processes competing for them. Each process may need 3 tape drives. The maximum value of '*n*' for which the system is guaranteed to be deadlock free is

(a) 2 (b) 3
(c) 4 (d) 1

Q.17 Consider a system having 'm' resources of the same type. These resources are shared by 3 processes A, B, C, which have peak time demands of 3, 4, 6 respectively. The minimum value of 'm' that ensures that deadlock will never occur is
(a) 11 (b) 12
(c) 13 (d) 14

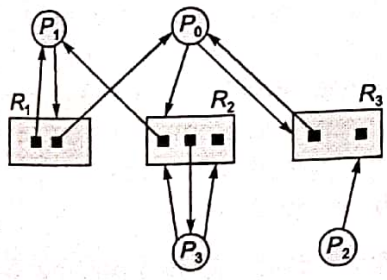
Q.18 A system has 3 processes sharing 4 resources. If each process needs a maximum of 2 units then, deadlock
(a) can never occur
(b) may occur
(c) has to occur
(d) none of these

Q.19 A system has four processes and five allocatable resources. The current allocation and maximum needs are as follows:

	Allocated	Maximum	Available
	M N O P Q	M N O P Q	M N O P Q
P_0	1 0 2 1 1	1 1 2 1 3	0 0 a 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	

What is the smallest value of 'a' for which this system is in a safe state?

Q.20 Consider the resource allocation graph in figure



This system is in a deadlock state. This remark is
(a) true
(b) false
(c) impossible to determine
(d) unpredictable

Q.21 A process executes the following code:

```
for (i = 0; i < n; i++) fork ( );
```

The total number of child processes created is
(a) n
(b) $2^n - 1$
(c) 2^n
(d) $2^{n+1} - 1$

Q.22 Consider the following code:

```
Void main ( ){
...
for (int k = 1; k < = 5; k ++){
    pid [k] = for k ( ); }
    }
```

In the given code all for k () statements executed successfully and all pid variables initialized to 0. What will be the total number of processes created by the above code?
(a) 1 (b) 5
(c) 16 (d) 32

Q.23 Consider:
$P = \{P_1, P_2, P_3\}$
$R = \{R_1, R_2, R_3, R_4\}$
One instance of resource type R_1, two instances of resource type R_2, one instances of resource type R_3 and three instances of resource type R_4. The situation is P_1 is holding one instance of R_2 and waiting for an instance R_1, P_2 is holding an instance of R_1, one instance of R_2 and waiting for an instance of R_3 and P_3 is holding an instance of R_3.
(a) Deadlock will not occur
(b) Deadlock will occur
(c) All the instances of R_4 are held by the processes
(d) All the instances of R_2 are not allotted

Q.24 Match **List-I** with **List-II** select the correct answer using the codes given below the lists:

List-I
A. Inconsistency
B. Deadlock resolution
C. Deadlock avoidance
D. Transit

List-II

1. requires the recognition of unsafe states
2. a message received by P_j from P_i that is not shown as sent by P_i
3. a message sent by P_i to P_j (mij) that does not belong to the set of messages received by P_j
4. always requires the abortion of one of more executing processes.

Codes:

	A	B	C	D
(a)	3	1	4	2
(b)	2	4	1	3
(c)	3	4	1	2
(d)	2	1	4	3

Q.25 The time complexity of Banker's algorithm to avoid deadlock having n processes and m resources is

(a) $O(m \times n)$ (b) $O(m + n)$
(c) $O(m^2 \times n)$ (d) $O(n^2 \times m)$

Q.26 How many process will be created when we run this program?

```
main ( )
{  print f("Hai");
   fork ( );
   fork ( );
}
```

(a) 2 (b) 3
(c) 4 (d) 1

Q.27 Consider a sytem with five processes P_0 through P_4 and three resource types A, B, C. Resource type A has 10 instances, resource type B has 5 instances, and resource type C has 7 instances. Suppose that, at time t_0, the following snapshot of the system has been taken:

Allocation	Max			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	0	0	3	2	2			
P_2	3	0	2	9	0	2			
P_3	2	1	1	2	2	2			
P_4	0	0	2	4	3	3			

Which of the following is the safe sequence for the above system?

(a) $<P_0, P_2, P_3, P_1, P_4>$
(b) $<P_1, P_2, P_4, P_0, P_3>$
(c) $<P_1, P_3, P_4, P_2, P_0>$
(d) We can not find a safe sequence

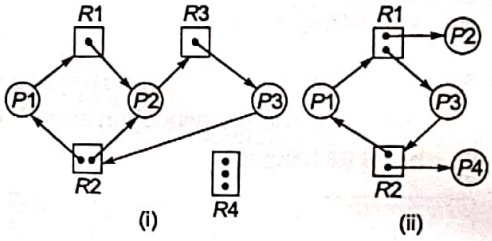
Q.28 Consider a system with three resources, tape drive, printer and disk drive. Each one of them are assigned a unique number

$R_1 = F$(Printer) = 12
$R_2 = F$(Tape Drive) = 1
$R_3 = F$(Disk Drive) = 4

Which of the following access patterns necessarily avoids deadlock?

(a) R_1, R_3, R_2 (b) R_2, R_3, R_1
(c) R_1, R_2, R_3 (d) R_3, R_2, R_1

Q.29 Which of the following is true?



(a) deadlock occurs in both the cases
(b) deadlock occurs in (i) but not in (ii)
(c) deadlock occurs in (ii) but not in (i)
(d) none of the above

Q.30 A fork system call will fail, if

(a) the previously executed statement in the program is also a fork call.
(b) after fork () both the parent and the child are executing the same program.
(c) the limit on the maximum number of processes that can be under execution by a single user exceeds.
(d) all the above.

Q.31 Which of the following statements are correct?

S_1: In fork system call, physical address of parent and child will be same.
S_2: In fork system call, virtual address of parent and child will be same.
S_3: In fork system call, child will return process id to its parent process.

(a) S_1 Only (b) S_1 and S_3 only
(c) S_2 and S_3 only (d) S_1 and S_2 only

Q.32 Consider a system has P processes. Each process need a maximum of m resources and a total of r resources available. Which of the following condition must hold to make the system deadlock free?

(a) $r = mp + 1$ (b) $r = mp + m$
(c) $r > mp + 1$ (d) $r >= p(m-1) + 1$

Q.33 Let U and V be two counting semaphore variables accessed by following three concurrent processes.

P_1	P_2	P_3
L_1 : wait(U)	L_2 : wait(V)	L_3 : wait(V)
print"C"	print"A"	print "D"
signal(V)	signal(V)	goto L_3
goto L_1	goto L_2	

Assume initial values of U and V are 3 and 0 respectively. Find the maximum number of C's that can be printed when the above processes are executed.

(a) 4 (b) 3
(c) 2 (d) 1

Q.34 Consider the following snapshot of a system with 5 processes (P_1, P_2, ..., P_5) and 4 resources (r_1, r_2, r_3, r_4).

Current Allocation				
	r_1	r_2	r_3	r_4
P_1	0	0	1	2
P_2	2	0	0	0
P_3	0	0	3	4
P_4	2	3	5	4
P_5	0	3	3	2

Max Demand			
r_1	r_2	r_3	r_4
0	0	1	2
2	7	5	0
6	6	5	6
4	3	5	6
0	6	5	2

Currently available resources are:

$r_1 = 2, r_2 = 1, r_3 = 0, r_4 = 0$

Find the safe sequence (execution order) if the system is not in deadlock?

(a) P_1, P_2, P_4, P_5, P_3
(b) P_1, P_3, P_2, P_4, P_5
(c) P_1, P_4, P_5, P_2, P_3
(d) None of these

Q.35 Consider the following code:

```
Pid = fork1 ( ) ;
If (Pid ! = 0)
fork2 ( ) ;
fork3 ( ) ;
```

The number of total process created when above code is executed is _____ (excluding parent).

Q.36 An operation system contain 8 user processes each require n unit of resource R. If the minimum number of units of R is 49 and no deadlock has occured then what is the value of n?

(a) 5 (b) 6
(c) 7 (d) 8

Q.37 A system has four processes and five allocatable resources. The current allocation and maximum needs are as follows:

	Allocated M N O P Q	Maximum M N O P Q	Available M N O P Q
P_0	1 0 2 1 1	1 1 2 1 3	0 0 a 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	

What is the smallest value of 'a' for which this system is in a safe state?

■■■■

Answers Concurrency and Deadlock

1.	(c)	2.	(d)	3.	(c)	4.	(c)	5.	(a)	6.	(a)	7.	(a)	8.	(c)	9.	(b)
10.	(c)	11.	(c)	12.	(d)	13.	(b)	14.	(d)	15.	(b)	16.	(a)	17.	(a)	18.	(a)
20.	(b)	21.	(b)	22.	(b)	23.	(b)	24.	(d)	25.	(c)	26.	(b)	27.	(c)	28.	(b)
29.	(b)	30.	(c)	31.	(c)	32.	(d)	33.	(b)	34.	(c)	36.	(c)				

Explanations Concurrency and Deadlock

2. (d)
Fork system call returns the pid of the child to the parent process and zero to the child process.

5. (a)
Using Banker's algorithm, one can show that one process has to acquire all its needed resources. This process, after completing its task, will release all its resources, there by avoiding any possible deadlock.

8. (c)
For deadlock four necessary condition are to be met
(i) Mutual exclusion
(ii) Non-preemption
(iii) Circular wait
(iv) Partial allocation [Bounded waiting]

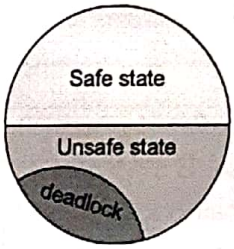
9. (b)
Deadlock also take place if resources are requested in same order by process in the scenario of circular wait.

11. (c)
Total 12 printers are there.

	Allocated	Max need	Remaining need
User-1	7	10	3
User-2	1	4	3
User-3	2	4	2
	10		

Each user is requesting 1 more printer.
Total 10 printers already allocated. As available number of printers are 2 only (12 – 10), so operating system can grant to user 3 only as its remaining need is 2 only, otherwise granting to any other user will lead to deadlock.

12. (d)



(i) Deadlock is unsafe state
(ii) The unsafe state may lead to deadlock.
(iii) Clearly deadlock state is subset of unsafe state.

Its not compulsory that unsafe state always result in deadlock.

13. (b)
Deadlock occurs when each of the 3 user processes hold one resource and make simultaneous demand for another. If there are 4 resources one of the 3 user processes will get the fourth instance of the resource and release one or both of the resource it is currently holding after using.

16. (a)
2 process can never lead to deadlock as the peak time demand of 6 (3 + 3) tape drives can be satisfied. But 3 processes can lead to a deadlock if each hold 2 drives and then demand one more.

17. (a)
Having 11 resources ensures that at least 1 process will have no pending request. This process after using will release the resources and so deadlock can never occur.

18. (a)

At least one process will be holding 2 resources in case of a simultaneous demand from all the processes. That process will release the 2 resources, thereby avoiding any possible deadlock.

19. (1)

First we need to find Need matrix.

Need = Maximum – Allocated

Need	
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

When a safety algorithm is run, the safety sequence will include P_3.

P_3 need is 0 0 1 1 1

Available is 0 0 a 2 3

$a \geq 1$

$\therefore$ Minimum value of $a = 1$

20. (b)

The system is not in deadlock state, if the following sequence of fulfillment of request of the processes is made

$$P_2 \rightarrow P_0 \rightarrow P_1 \rightarrow P_3$$

26. (b)

The number of processes created = $2^{(\text{no. of times the fork has called})} = 2^2 = 4$

But the number of child processes = $2^2 - 1 = 3$

27. (c)

Process	Need			Available		
	A	B	C	A	B	C
P_0	7	4	3	3	3	3
P_1	1	2	2			
P_2	6	0	0			
P_3	0	1	1			
P_4	4	3	1			

$<P_1, P_3, P_4, P_2, P_0>$ is the safe sequence.

28. (b)

To avoid deadlock, we must request the devices in ascending order of enumeration. This ensures that circular wait does not hold. The correct sequence is 1, 4, 12, i.e. R_2, R_3, R_1.

29. (b)

In (a) the processes P_1, P_2 and P_3 are in deadlock state. P_2 waiting on R_3. Which is held by P_3. P_1 is waiting on R_1 which is held by P_2. While P_3 in waiting on R_2 which is held by P_1 and P_2.

In (b) we can observe that P_4 can release instance of R_2, which can be allocated to either P_1 or P_3, breaking the cycle. Deadlock does not occur in (b).

30. (c)

A fork () will fail if the limit on the maximum number of processes that can be under execution by a single user exceeds.

31. (c)

In fork system call both parent and child will have same virtual address but physical address will be different.

When fork == 0 then child process is created and after completion of child process it return its process id to its parent.

32. (d)

If a process has m resources it can finish and can not be involved in a deadlock. Therefore, the worst case is where every process has $m-1$ resources and needs another one. If their is one resource left over, one process can finish and replace all its resources, letting the rest to finish.

Therefore the condition for avoiding deadlock is $r >= (m-1)p + 1$.

33. (b)

U = 3, can be decremented 3 times by P_1 execution first and prints "C" 3 times. If we observe in the given code there is no signal (U). When P_1 executes 4th time, it will be blocked.

$\therefore$ C is printed 3 times.

34. (c)

Current Allocation				
	r_1	r_2	r_3	r_4
P_1	0	0	1	2
P_2	2	0	0	0
P_3	0	0	3	4
P_4	2	3	5	4
P_5	0	3	3	2

Max Demand			
r_1	r_2	r_3	r_4
0	0	1	2
2	7	5	0
6	6	5	6
4	3	5	6
0	6	5	2

Available = 2, 1, 0, 0 (P_1 satisfies it needs, so it releases all allocated resources)

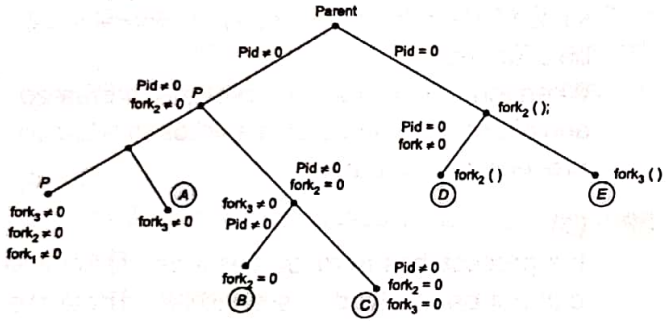
P_1 : 0 0 1 2
Available 2 1 1 2

(Now P_4 can satisfy it needs and releases its resources)

P_4 : 2 3 5 4
Available 4 4 6 6
P_5 : 0 3 3 2
Available 4 7 9 8
P_2 : 2 0 0 0
Available 6 7 9 8
P_3 : 0 0 3 4

<P_1, P_4, P_5, P_2, P_3> is execution order.

35. (5)



Total 5 process created.

36. (c)

8 users with 49 resources as minimum number.

Then $8(n-1)+1 = 49$

$8n - 8 + 1 = 49$

$8n = 49 + 7$

$n = 7$

Each process require 7 resource.

37. (1)

First we need to find Need matrix.

Need = Maximum – Allocated

Need

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

When a safety algorithm is run, the safety sequence will include P_3.

P_3 need is 0 0 1 1 1

Available is 0 0 a 2 3

$a \geq 1$

$\therefore$ Minimum value of $a = 1$

6 CHAPTER
Memory Management

Q.1 In a paging system
(a) a page frame is always larger than an incoming page
(b) a page frame is always smaller than the incoming page
(c) given the virtual memory address $v = (p, d)$, where page p is in page frame p' and the fixed page size is p_s, then the real memory address of v is $(p' + d) * p_s$
(d) when a process references a page that is not in main memory, the operating system loads the nonresident page into memory from secondary storage

Q.2 Choose the correct statement
(a) paging is virtual storage and segmentation is real storage.
(b) both paging and segmentation are forms of contiguous storage allocation.
(c) both paging and segmentation are popular schemes for virtual storage.
(d) none of these

Q.3 All process use a single page table is known as
(a) inverted page table
(b) multilevel page table
(c) hashed page table
(d) linked page table

Q.4 Assume the following free memory partitions were available

$M_1 = 100k$
$M_2 = 500k$
$M_3 = 200k$
$M_4 = 450k$
$M_5 = 600k$

How would the best-fit algorithm place the processes with following sizes?

$P_1 = 212k$
$P_2 = 417k$
$P_3 = 112k$
$P_4 = 426k$

(a) P_1-M_4, P_2-M_5, P_3-M_3, P_4-M_2
(b) P_1, M_4, P_2-M_2, P_3-M_3, P_4-M_5
(c) P_1-M_2, P_2-M_4, P_3-M_3, P_4-M_5
(d) P_1-M_2, P_2-M_5, P_3-M_3, P_4-M_4

Q.5 In a computer system where the 'best fit' algorithm is used for allocating jobs to memory partitions the following situation was encountered

Partitions sizes in kB	4 k	8 k	20 k	2 k
Jobs sizes in kB	2 k	14 k	3 k	6 k
	6 k	10 k	20 k	2 k
Times for execution	4	10	2	1
	4	1	8	6

When will the 20 k job complete?
(a) 9 units (b) 18 units
(c) 8 units (d) 19 units

Q.6 Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?

<1, 10>, <2, 500>

Note: <a, b> means <segment no, offset>
(a) 2400, 590
(b) 2310, Invalid offset
(c) 2400, Invalid offset
(d) 2310, 590

Q.7 In a system with 62 frames there are two processes running, P_1 of size 10 k and P_2 of size 127k. How many frames will be allocated to each of the processes by proportional allocation scheme?
(a) 31 frames each of P_1 and P_2
(b) 4 frames to P_1 and 57 frames to P_2
(c) 2 frames to P_1 and 60 frames to P_2
(d) none of the above

Q.8 Which of the following statements is correct about paging?
(a) paging suffers problem of external fragmentation
(b) paging suffers from internal fragmentation
(c) paging suffers from problem of internal and external fragmentation
(d) paging completely avoids all sorts of fragmentation

Q.9 A certain computer provides its users with a virtual-memory space of 2^{32} bytes. The computer has 2^{18} bytes of physical memory. The virtual memory is implemented by paging and the page size is 4096 bytes. A user process generates the virtual address 0X11123456. What is the displacement in the page?
(a) 0001 0001 0001
(b) 0001 0001 0001 0010 0011
(c) 0010 0011 0100 0101 0110
(d) 0100 0101 0110

Q.10 There are half as many holes as process, the S be the average size of process and ks be the average size of holes then the total memory M is estimated using

(a) $M = \frac{n}{2} \times S + nS$ (b) $\frac{n}{2} \times kS = M - nS$

(c) $kS \times m = \frac{n}{2} \times nS$ (d) $nS\left(\frac{k}{2} - 1\right)$

Q.11 Given memory partitions in the order below:
P_1: 100k, P_2: 500k, P_3: 800k, P_4: 300k, P_5: 600k.
How would WORST FIT algorithms place processes: 212k, 417k, 112k, and 426k (in order)
(a) P_3, P_5, P_4, P_2
(b) P_2, P_5, P_3, 426k waits
(c) P_3, P_5, P_2, 426k waits
(d) P_4, P_2, P_5, P_3

Q.12 In partitioned memory allocation scheme, the
(a) best fit algorithm is always better than the first fit algorithm
(b) first fit algorithm is always better than the best fit algorithm
(c) superiority of the first fit and best-fit algorithms depend on the sequence of memory requests.
(d) None of the above

Q.13 If there are 32 segments, each of size 1 k bytes, then the logical address should have
(a) 13 bits (b) 14 bits
(c) 15 bits (d) 16 bits

Q.14 In a paged segmented scheme of memory management, the segment table itself must have a page table because
(a) the segment table is often too large to fit in one page
(b) each segment is spread over a number of pages
(c) segment tables point to page tables and not to the physical location of the segment
(d) the processor's description base register points to a page table

Q.15 A computer installation has 1000 K of main memory. The jobs arrive and finish in the following sequence.
Job 1 requiring 200 K arrives
Job 2 requiring 350 K arrives
Job 3 requiring 300 K arrives
Job 1 finishes
Job 4 requiring 120 K arrives
Job 5 requiring 150 K arrives
Job 6 requiring 80 K arrives
Among best fit and first fit, which performs better for this sequence?
(a) first fit
(b) best fit
(c) both perform the same
(d) none of the above

Q.16 Suppose that a certain computer with paged virtual memory has 4 KB pages, a 32-bit byte-addressable virtual address space and, 30 bit byte-addressable physical address space. The system manages an inverted page table where each entry includes the page number plus 12 overhead bits. How big is the basic inverted page table including page number and overhead bits?

(a) 2^{10} B (b) 2^{20} B
(c) 2^{30} B (d) 2^{32} B

Q.17 A system contains 16 MB of primary memory and fixed partitions, all of the size 65536 bytes. What is the minimum number of bits needed in an entry in the process table to record all of the partitions.

(a) 8 bits (b) 16 bits
(c) 32 bits (d) 64 bits

Q.18 Consider a paging system with 16 MB of physical memory, 256 pages of logical address space and a page size of 1 KB. A page frame needs how many memory space?

(a) 1 KB (b) 2 KB
(c) 16 KB (d) 3 KB

Q.19 Consider an address of 16 bit, with 4 bit as segment number, then maximum possible segment size is _____.

(a) 16 (b) 65536
(c) 4096 (d) 12

Q.20 In a simple paging system with 2^{24} bytes of physical memory, 256 pages of logical address space, and a page size of 2^{10} bytes, how many bits are in logical address.

(a) 4 (b) 14
(c) 10 (d) 18

Q.21 Assume a machine with 64 MB physical memory and 32 bits virtual address space. If the page size is 4 KB, what is the approximate size of the page table?

(a) 16 MB (b) 1 MB
(c) 24 MB (d) 2 MB

Q.22 On a system using simple segmentation, following is the segment table

Segment	Base	Length
0	330	124
1	876	211
2	111	99
3	498	302

Compute the physical address for the logical address 3,222

(a) 720 (b) 498
(c) 302 (d) 800

Q.23 Consider the 23 bit logical address specified as

Segment	Page	Words
5 bits	9 bits	9 bits

The size of the largest segment is

(a) 64 K words (b) 128 K words
(c) 256 K words (d) 512 K words

Q.24 Consider virtual address space of 32 bits and page size is of 4 KB. Consider RAM of 128 KB. Then what will be ratio of page table and inverted page table size if each entry in both is of 4 B?

(a) 2^{15} : 1 (b) 2^{20} : 1
(c) 2^{10} : 1 (d) None of these

Q.25 On a system the average process size be 128 KB and each page entry requires 8 bytes then what will be the optimal page size.

(a) 1024 bytes (b) 1448 bytes
(c) 512 bytes (d) 2048 bytes

Q.26 Match **List-I** with **List-II** select the correct answer using the codes given below the lists:

List-I

A. Next fit
B. Buddy system
C. Best fit
D. First fit

List-II

1. The memory is divided into a number of small size memory and required number of memories are allocated to the needed segment.
2. When process and holes are kept on a list sorted by address then, this algorithm is faster.

3. It chooses the hole from the list, from the place where it left off last time while searching hole.
4. When process and holes are kept on a list sorted on size then this algorithm is faster.
5. We choose the hole into which needed segment will fit.

Codes:

	A	B	C	D
(a)	2	5	1	4
(b)	3	2	4	1
(c)	2	1	4	5
(d)	3	1	4	2

Q.27 In a paged memory, the page hit ratio is 0.35. The time required to access a page in secondary memory is equal to 100 ns. The time required to access a page in primary memory is 10 ns. The average time required to access a page is

(a) 3.0 ns (b) 68.0 ns
(c) 68.5 ns (d) 78.5 ns

Q.28 Consider a system using paging and segmentation. The virtual address space consist of up to 8 segments and each segment is 2^{29} bytes long. The hardware pages each segment into 2^8 byte pages.
How many bits in the virtual address specify the offset within page?

(a) 8 bits (b) 16 bits
(c) 32 bits (d) 64 bits

Q.29 A system uses a two level page table has 2^{12} bytes pages and 32-bit virtual addresses. The first 8-bit of the address serve as a index into the first level page table.
How many bits specify the second-level index?

(a) 8 bits (b) 10 bits
(c) 12 bits (d) 16 bits

Q.30 A computer system supports 32-bit virtual addresses as well as 32-bit physical addresses. Since the virtual address space is of the same size as the physical address space, the operating system designers decide to get rid of the virtual memory entirely.
Which one of the following is true?

(a) Efficient implementation of multi-user is no longer possible
(b) The processor cache organized can be made more efficient now
(c) Hardware support for memory management is no longer needed
(d) CPU scheduling can be made more efficient now

Q.31 In a paging system

(a) a page frame is always larger than an incoming page
(b) a page frame is always smaller than an incoming page
(c) given the virtual memory address $v = (p, d)$, where page p is in address of v is $(p'+d)*P_s$
(d) when a process references a page that is not in main memory, the operating system loads the non-resident page into memory from secondary storage

Q.32 There are 5 processes having an average size of 100 k and the average size of hole is 60 k. The percentage of memory wasted in holes is

(a) 76 (b) 37
(c) 23 (d) none of these

Q.33 Consider a machine with 48-bit virtual address and a page size of 16 MB. During a program execution the TLB contains the following valid entries (in hexadecimal)

Virtual Page No.	Physical Frame No.
0x0	0x1
0xAA	0x2
0xAA9	0x3
0xAA98	0x4
0xAA987	0x5
0xAA9876	0x6

Translate the following virtual address into a 52-bit physical address (in hexadecimal)?
'virtual address = 0x 00AA9876 AB01'

(a) 0x 000000476AB01
(b) 0x000000056AB01
(c) 0x 000003876AB01
(d) None of these

Q.34 Which of the following is false?

(a) Larger disk block results in greater internal fragmentation.

(b) Smaller page size results in less internal fragmentation.

(c) Fixed partitioning scheme results in external fragmentation.

(d) Variable partitioning scheme results in internal fragmentation.

Q.35 Consider the following information:

- TLB hit rate is 90% and TLB access time is 1 cycle.
- Cache hit rate is 80% and cache access time is 1 cycle.
- When TLB and cache both get miss, page fault rate is 2%.
- Main memory access time is 20 cycles.
- Hard drive access time is 500 cycles.
- TLB access and cache access are sequential.
- Page table is always kept in memory.

Compute the average memory access time when the cache is virtually addressed.

(a) 4.2 (b) 5

(c) 5.6 (d) 5.8

Q.36 Consider a system with 2-level paging with TLB Support. The page table has divided into 2 k pages each of size 4 k words. If physical address space has 64 words which is divided into 16 k frames. The TLB access time is 20 ns while main m/m access time is 120 ns. If a CPU finds 140 page references in TLB out of total 200 then the effective memory access time using TLB is _____ (ns).

Q.37 Consider a system using segmented paging architecture. The segment is divided into 1 K pages. Each page having 128 entries. Segment table is divided into 512 pages and each page having 1024 entries. The size of page table entries are 2 words. If 15 bits are required to represent the frames of physical memory, the size of page table of segment table is _____ (in words) [assume memory is word addressable].

■■■■

Answers Memory Management

1.	(d)	2.	(c)	3.	(a)	4.	(b)	5.	(d)	6.	(b)	7.	(b)	8.	(b)	9.	(d)
10.	(b)	11.	(c)	12.	(c)	13.	(c)	14.	(b)	15.	(a)	16.	(b)	17.	(a)	18.	(a)
19.	(c)	20.	(b)	21.	(d)	22.	(a)	23.	(c)	24.	(a)	25.	(b)	26.	(d)	27.	(c)
28.	(a)	29.	(c)	30.	(c)	31.	(d)	32.	(c)	33.	(a)	34.	(d)	35.	(d)		

Explanations Memory Management

4. (b)

Best fit algorithm tries to select the free memory partition that can be best suited for the memory needed for a process. Among the available free memory partitions, P_1 can fit in M_2, M_4 and M_5, But the algorithm select the smallest memory partition from the above three. Hence it select M_4 for P_1 and so on.

5. (d)

The 20 k partition will be occupied by the 14 k job which will require 10 units of time for execution. After its completion the 10 k job will be occupying this 20 k partition. And on its competition 20 k will be taking up the partition.
Hence total time = 10 + 1 + 8 = 19 units.
Other jobs will be occupying the other partitions.

6. (b)

For <1, 10>:

a : 10 < 14 (from 2nd entry of the segment table)

b : Hence the logical physical address

= Base + offset

= 2300 + 10 = 2310

For <2, 500>:

a : 500 > 100 (from 3rd entry of the segment table)

b : Hence this is invalid

7. (b)

Suppose total number of frames are M, and S_i be the memory requirement of its process, then the total memory requirement is

$$S = \sum_{1}^{k} S_i$$

The proportional allocation scheme, the amount of frames allocated to process p_i with size S_i is

$= \frac{S_i}{S} \times M$ for the given data

$M = 62$, $S = 10 + 127 = 137$

The process P_1 gets $(10/137) \times 62 = 4$ frames

The process P_2 gets $(127/137) \times 62 = 57$ frames

8. (b)

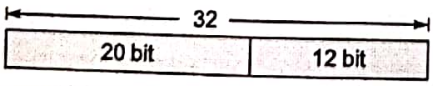
Paging avoids problem of external fragmentation. However it suffers from problem of internal fragmentation.

9. (d)

Size of page = Size of frame = 4096 = 2^{12} bytes

So entry inside the page table = $\frac{2^{32}}{2^{12}} = 2^{20}$

So logical address can be divided into



given address in hexadecimal

0001 0001 0001 0010 0011 0100 0101 0110

last 12 bit represent displacement within a page.

So in this case displacement is 0100 0101 0110.

10. (b)

$$\frac{n}{2} \times ks = M - nS$$

⇒ Number of space occupied by holes

= Total space – Number of space occupied by process

i.e. $M = \frac{n}{2} \times ks - nS = nS\left(\frac{k}{2} + 1\right)$

11. (c)

Principle of Best Fit allocation

Allocate the largest hole that is available. We must search the entire list, unless the list in kept ordered by size.

212 k is put in 800 k partition

417 k is put in 600 k partition

112 k is put in 500 k partition

426 k have to wait because there is no partition free that is big enough for 426 k.

12. (c)

In partitioned memory allocation scheme, it is not always necessary that best fit ot first fit algorithm will be better than any of these. It depends on the sequence of requests that have been made by processes.

13. (c)

To specify a particular segment, 5 bits are required (since $2^5 = 32$). Having selected a page, to select a particular byte one needs 10 bits (since 2^{10} = 1K byte). So, totally 5 + 10 = 15 bits are needed.

15. (a)

The memory configuration after the arrival of the jobs 1, 2 and 3 and the termination of job 1 can be depicted as :

FREE–200 JOB 2–350 JOB3–300 FREE–150

First fit algorithm will allocate the FREE-200 slot for job 4. But best fit algorithm will allocate the FREE-150 slot for job 4. The memory configuration for the first fit and best fit will be

JOB4–120 FREE–80 JOB2–350 JOB3–300 FREE–150 and FREE–200 JOB2–350 JOB3–300 JOB4–120 FREE–30

respectively. When job 5 arrives, it will be allotted the FREE–150 slot by the first fit algorithm and the FREE–200 slot by the best fit algorithm. The memory allocation table for the first fit and best fit will be

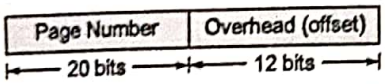
JOB4–120 FREE–80 JOB2–350 JOB3–300 JOB5–150
and
JOB5–150 FREE–50 JOB2–350 JOB3–300 FREE–30

When job 6 arrives, it will be allotted the FREE-80 slot by the first fit algorithm. The best fit algorithm will find no room to store Job 5 as the needed 80 K, is not available contiguously. So, it has to wait till a job terminates. So, the first-fit algorithm performs better in this case.

16. (b)
Virtual address space = 2^{32} Bytes
Physical address spaces = 2^{30} Bytes
Page size = 4 KB = 2^{12} Bytes

No. of frames in physical memory = $\frac{2^{30}}{2^{12}} = 2^{18}$

No. of Bits in each page table entry



Page table size = No. of frames × (20 + 12) bits
= 2^{18} × 32 bits = 2^{18} × 32/8 bytes ≈ 2^{18} × 2^2 bytes
≈ 2^{20} Bytes

17. (a)
Primary memory size = 16 MB = 2^{24} Bytes
Partition size = 65536 Bytes = 2^{16} Bytes

No. of partitions = $\frac{2^{24}\text{ Bytes}}{2^{16}\text{ Bytes}} = 2^8$

Hence number of bits needed in an entry in the process table to record all of the partitions = 8 bits.

18. (a)
No. of bits to identify each frame in physical memory are equal to the no. of bits to identify each page in logical memory, because page size is equal to frame size.
Hence correct option is 1 KB.

19. (c)
Since four bits are used as segment number out of 16 bits, a maximum of 12 bits can participate in the segment size, which is equal to 2^{12} = 4096.

20. (d)
Physical memory size = 2^{24} bytes
Space size = 2^{10} bytes
Logical address space = 256 pages
= 256 × 2^{10} = 2^{18}
No. of bits required = 18 bits

21. (d)
Physical memory size = 2^{28} bytes
Virtual memory size = 2^{32} bytes
Page size = 2^{12} bytes
No. of pages in virtual memory

$= \frac{2^{32}}{2^{12}} = 2^{20}$ pages

No. of pages in main memory

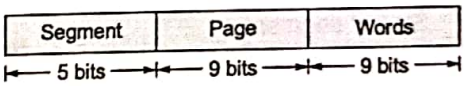
$= \frac{2^{28}}{2^{12}} = 2^{14}$ frame

Page table size = No. of page in virtual memory × No. of bits used to identifie frame

$= 2^{20} \times 14$ bit $= 2^{20} \times \frac{14}{8}$ bytes ≈ 2 MB

22. (a)
The physical address for the logical address 3, 222 represents that 222th location byte of segment 3
= 222 + base address of segment 3
= 222 + 498 = 720

23. (c)



Here segment no. is occupying 5 bits out of total of 23 bits, the remaining bits = 23 – 5 = 18 bits. The size of largest segment is hence = 2^{18} words which equal to 256 K words.

24. (a)
Page table size = No. of pages in logical Address space × each entry size

$= \frac{2^{32}}{2^{12}} \times 4 = 2^{20} \times 4$

Inverted page table size = No. of pages in main memory × each entry size

$$= \frac{2^{17}}{2^{12}} \times 4 = 2^5 \times 4$$

$$\text{Ratio} = \frac{2^{20} \times 4}{2^5 \times 4} = 2^{15} : 1$$

26. (d)

In **next fit** algorithm, the memory chunk is chosen from the place where it left off last time while searching chunk.

In buddy system, the traditional way of division of memory into number of small size memory is done and the small chunks are allocated according to the need of processes.

In best fit algorithm, the chunks are allocated according to the process size after sorting the list of process size and memory chunks.

In first fit algorithm, the chunks are allocated according to their sorted address.

27. (c)

Hit ratio = 0.35

Time (secondary memory) = 100 ns

T(main memory) = 10 ns

Average access time = $h(T_m) + (1 - h)$

$= 0.35 \times 100 + (0.65) \times 100 = 3.5 + 65 = 68.5$ ns

28. (a)

Virtual address space = $8 \times 2^{29} = 2^{32}$ bytes

Number of pages within each segment

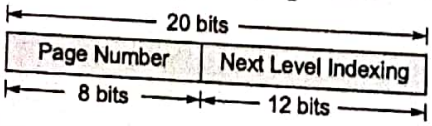
$$= \frac{\text{Size of segment}}{\text{Size of page}} = \frac{2^{29}}{2^8} \text{ Bytes} = 2^{21} \text{ Bytes}$$

Offset within the page = 8 bits (for 2^8 bytes)

29. (c)

$$\text{Number of pages} = \frac{2^{32}\text{B}}{2^{12}\text{B}} = 2^{20}$$

Now since first 8 bits of the address serve as a index into the first level page table



Number of bits specify the second level index = 12 bits.

30. (c)

Since the physical memory is equal to the virtual memory and the virtual memory is to be removed hence the hardware support for memory management is no longer needed for the conversion of virtual address to physical address.

32. (c)

% of memory wasted in holes = fraction of memory in holes × 100

$$= \frac{\text{Total memory occupied by holes}}{\text{Total memory}} \times 100$$

$$= \frac{\frac{n}{2} \times kS}{nS\left(\frac{k}{2}+1\right)} \times 100 = \left(\frac{kS}{k+2}\right) \times 100$$

$= 0.23 \times 100 = 23\%$

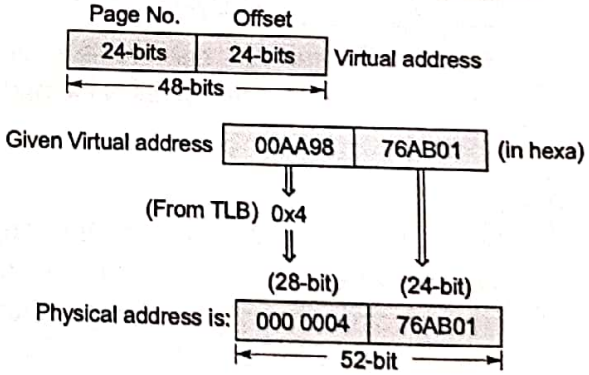
$kS = 60$

$S = 100$

$$k = \frac{3}{5}$$

33. (a)

Page size = 16 MB = 2^{24} bytes ⇒ hexa digits for offset



∴ Ox 000 000476AB01 is physical address.

34. (d)

Variable partition or segmentation not results in internal fragmentation. But which may suffer from external fragmentation.

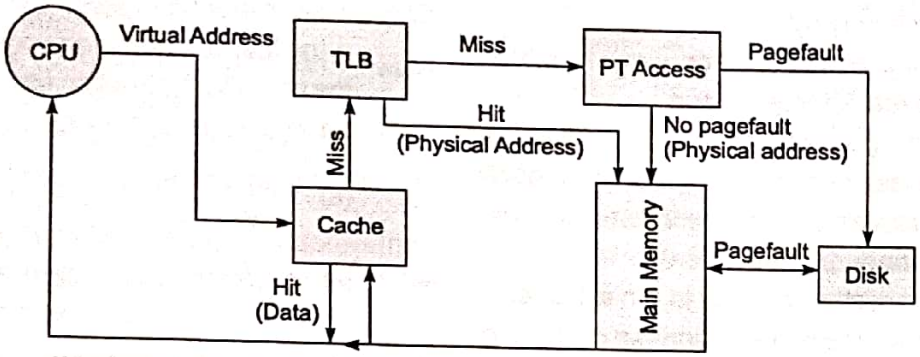
35. (d)

Virtually addressed:

T_{avg} = Cache hit × T_{cache} + Cache miss × ([TLB_{hit} × [T_{cache} + T_{TLB} + T_{Mem}]] + [TLB_{miss} × No page fault × [T_{cache} + T_{TLB} + T_{Mem} + T_{Mem}] + TLB_{Miss} × Page fault × [T_{cache} + T_{TLB} + T_{Mem} + T_{Mem} + T_{disk}]])

$T_{avg} = 0.8 \times 1 + 0.2 \times ([0.9 \times [1 + 1 + 20]] + [0.1 \times 0.98 \times [1 + 1 + 20 + 20]] + 0.1 \times 0.02 \times [1+1+20+20+500])$

$= 0.8 + 0.2 \times (19.8 + 4.116 + 1.084) = 5.8$



Note: If there is cache miss, TLB miss then there is chance of pagefault.

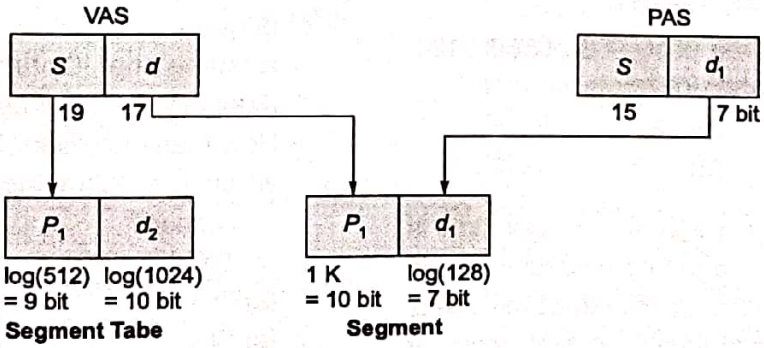
36. (212)

$$\text{Hit ratio of TLB} = \frac{\text{\# Reference found in TLB}}{\text{\# Total reference}} = \frac{140}{200} = 0.70$$

The system uses 2 - level paging with TLB

$$\text{EMAT} = x(c + m) + (1 - x)(c + 3m)$$
$$= 0.70(20 + 120) + (1 - 0.70)(20 + 360)$$
$$= 0.70(140) + (0.30)(380)$$
$$= 98 + 114 = 212 \text{ ns.}$$

37. (1024)



Page table size of segment table = Number of entries of segment table × Segment table entry size

$= 2^9 \times 2$ words

= 1024 words = 1 K words

■■■■

7 CHAPTER

Virtual Memory

Q.1 In a virtual memory system
(a) each virtual address space is much smaller than the real address space
(b) determining how to map virtual addresses to physical addresses is of great importance
(c) MMUs convert physical addresses to virtual addresses during execution
(d) when the system is ready to run a process, the system loads the process's code and data from main memory into cache

Q.2 Page fault frequency in an operating system is reduced when the
(a) size of pages is reduced
(b) processes tend to be I/O-bound
(c) processes tend to be CPU-bound
(d) locality of reference is applicable to the process

Q.3 The address sequence generated by tracing a particular program executing in a pure demand paging system with 100 records per page with 1 free main memory frame is recorded as follows. What is the no. of page faults?

0100, 0200, 0430, 0499, 0510, 0530, 0560, 0120, 0220, 0240, 0260, 0320, 0370.

(a) 13 (b) 8
(c) 7 (d) 10

Q.4 Assume the average page-fault service time is 25 milliseconds and a memory access time is 100 nanoseconds. What is the effective access time in nanoseconds when the probability of page-fault is p?
(a) 100 + 25000000 p
(b) 100 + 24999900 p
(c) 25000000 – 24999900 p
(d) 25000000 + 24999900 p

Q.5 If an instruction takes 'i' microseconds and a page fault takes an additional 'j' microseconds, the effective instruction time if on the average a page fault occurs every 'k' instruction is:

(a) $i+\frac{j}{k}$ (b) $i + j * k$

(c) $\frac{i+j}{k}$ (d) $(i + j) * k$

Q.6 The address sequence generated by tracing a particular program executing in a pure demand paging system with 100 records per page with 1 free main memory frame is recorded as follows. What is the number of page-faults?

0100, 0200, 0345, 0430, 0499, 0510, 0530, 0120, 0199, 0240, 0260, 0350, 0599

(a) 9 (b) 13
(c) 8 (d) 10

Q.7 A demand paged virtual memory system uses 16 bit virtual address, page size of 256 bytes, and has 1 byte of main memory.

Currently the main memory contains the following pages 1, 17, 56.
Assume that Optimal Page Replacement algorithm has been used.
How many number of page-faults will happen when the following addresses has been referenced?

0X00FF, 0X010D, 0X10FF, 0X11B0

(a) 1 (b) 2
(c) 0 (d) 3

Q.8 Assume we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified, and 20 milliseonds if the replaced

page is modified. Memory access time is 100 ns. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200ns?

(a) 6.1×10^{-6} (b) 7.3×10^{-6}
(c) 3.4×10^{-4} (d) none of these

Q.9 Page fault occurs when
(a) the page is corrupted by application software
(b) the page is in main memory
(c) the page is not in main memory
(d) the CPU tries to divide a number by 0

Q.10 The page replacement policy that sometimes leads to more page faults when the size of the memory is increased is
(a) FIFO
(b) LRU
(c) no such policy exists
(d) none of the above

Q.11 Thrashing
(a) reduces page I/O
(b) decreases the degree of multiprogramming
(c) implies excessive page I/O
(d) improves the system performance

Q.12 Given reference to the following page by a program: 0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7, 8, 2, 3, 8, 3

If the program contains 3 page frames. How many page fault will occur in optimal page replacement policy?

(a) 4 (b) 5
(c) 6 (d) 7

Q.13 Four page frames and page references in the order : 0 2 1 3 2 1 0 1 2 1.

By using LRU page replacement algorithm the least recently used page will be
(a) 0 (b) 1
(c) 2 (d) 3

Q.14 Determine the page faults when FIFO is applied to the following string with 3 frames 2, 4, 3, 1, 5, 3, 2, 5, 3, 1, 4, 2.

(a) 10 (b) 12
(c) 9 (d) 7

Q.15 Consider the following page addresses stream frequency by executing the program with 3 frames. 1 2 3 1 2 5 8 7

By using optimal page replacement, number of page fault will be _____ and the number of page hit will be _____.

(a) 3 6 (b) 5 6
(c) 2 6 (d) 6 2

Q.16 In paged memory, the page hit ratio is unity. The time required to access a page in primary memory is 20 ns, then the average time required to access page is
(a) 14 ns (b) 20 ns
(c) 6 ns (d) 10 ns

Q.17 Determine the number of page faults when references to pages occur in the order: 1, 2, 4, 5, 2, 1, 2, 4. Assume that the main memory can accommodate 3 pages and the main memory already has the pages 1 and 2, with page 1 having been brought earlier than page 2 (assume LRU algorithm is used).

(a) 3 (b) 5
(c) 4 (d) None of these

Q.18 A process refers to 5 pages, *A*, *B*, *C*, *D* and *E* in the order: *A*, *B*, *C*, *D*, *A*, *B*, *E*, *A*, *B*, *C*, *D*, *E*.

If the page replacement algorithm is FIFO, the number of pages which transfer with an empty internal store of 3 frames is
(a) 8 (b) 10
(c) 9 (d) 7

Q.19 The address sequence generated by tracing a particular program executing in a pure demand paging system with 100 records per page, with 1 free main memory frame is recorded as follows. What is the number of page faults?

0100, 0200, 0430, 0499, 0510, 0530, 0560, 0120, 0220, 0240, 0260, 0320, 0370

(a) 13 (b) 8
(c) 7 (d) 10

Q.20 A memory page containing a heavily used variable that was initialized very early and is in constant use is removed, when the page replacement algorithm used is

(a) LRU (b) FIFO
(c) LFU (d) None of these

Q.21 If page fault service time is 50 milliseconds and memory access time is 100 nanoseconds, then what will be the effective access time, if the probability of page fault is p.

(a) $(500000 + 100\,p)$ nanoseconds
(b) $(100 + 500000 \times p)$ nanoseconds
(c) $10^{-7} - 10^{-7}\,p\,(500000)$ seconds
(d) $10^{-7} + 49.9 \times 103\,p$ seconds

Q.22 The minimum number of page frames that must be allocated to a running process in a virtual memory environment is determined by

(a) the instruction set architecture
(b) page size
(c) physical memory size
(d) number of process in memory

Q.23 There are five virtual pages, numbered from 0 to 4. The page are referenced in the following order

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

If FIFO page replacement algorithm is used then find

1. Number of page replacement faults that occur if 3 page frames are present.
2. Number of page faults that occur if 4 page frames are present.

(a) 10, 9 (b) 9, 10
(c) 9, 8 (d) 10, 11

Q.24 Given references to the following pages by a program: 0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7, 8, 2, 3, 8, 3.

How many page faults will occur if the program has been three page frames available to it and uses an optimal replacement?

(a) 7 (b) 8
(c) 9 (d) None of these

Q.25 A demand paging system takes 100 time units to service a page fault and 300 time units to replace a dirty page. Memory access time is 1 time unit. The probability of a page fault is p. In case of page fault, the probability of page being dirty is also p. It is observed that the average access time is 3 times units. Then the value of p is

(a) 0.194 (b) 0.233
(c) 0.514 (d) 0.981

Q.26 Consider the main memory with five page frames and the following sequence of page references: 9, 8, 7, 9, 3, 0, 2, 9, 8, 3, 9, 2, 0, 9. What will be result of $(X + Y)$ [where X is number of page faults using LRU policy and Y is number of page faults using FIFO policy]?

(a) 14 (b) 15
(c) 16 (d) 18

Q.27 The number of page faults with a given string S using LRU algorithm is N and considering the reverse of string S^R the number of page fault's will be

(a) N (b) $\frac{N}{2}$
(c) $2N$ (d) None of these

Q.28 Consider the following page reference string:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6

How many page faults would occur for the LRU replacement algorithm with 3 available free frames in physical memory?

Q.29 Consider a system using demand paging architecture where it takes 6 ms to service the page fault if either empty frame is available or replaced page is not to be modified. It takes 15 ms if the replaced page is modified. Assuming the main memory access time = 1 ms and 70% of time page to be replaced is modified. The maximum acceptable page fault rate to get EMAT (effective memory access time) not more than 3 ms is _____ % (upto 2 decimal places).

Answers Virtual Memory

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

Explanations Virtual Memory

3. (c)
When it tries to access 0100, it results in a page fault as the memory is empty right now. So, it loads the second page (which has the addresses 100-199). Trying to access 200 will result in a page fault as it is not in memory right now. So the third page with the addresses from 200 to 299 will replace the second page in memory. Trying to access 430 will result in another page fault. Proceeding this way, we find trying to access the addresses 0510, 920, 0220 and 0320 will all result in page faults. So, altogether 7 page faults.

4. (b)
Effective access time = (probability of success) × (memory access time) + (probability of page fault) × (average page-fault service time)
$= (1 - P) \times (100) + (P) \times (25000{,}000)$
$= 100 + 24999900\,P$

5. (a)
$$\left(1-\frac{1}{k}\right)i+\left(\frac{1}{k}\cdot i+\frac{1}{k}\cdot j\right)$$

6. (a)
A page fault happens when we refer the following addresses: 0100, 0200, 0345, 0430, 0510, 0120, 0240, 0350, 0599.

7. (b)
The memory references are given in hexadecimal, so first convert then into binary equivalent.
00FF ⇒ 0000 0000 1111 1111 ⇒ 255 in decimal
010D ⇒ 0000 0001 0000 1101 ⇒ 269
10FF ⇒ 0001 0000 1111 1111 ⇒ 4351
11B0 ⇒ 0001 0001 1011 0000 ⇒ 4522
As the page size is 256 bytes so page–0 will contain 0 to 255 bytes.

Page–0	0 to 255
Page–1	256 to 511
Page–17	4352 to 4607

00FF will cause page fault, because it lies in page 0 which is not in memory and it will replace page 56. 010D belongs to the page 1, hence no page fault. 10FF will again cause page fault because it lies in page 16 which is not in memory and can replace any of page 0 or page 1. 11B0 will not care page fault because it belongs to page 17. Which in already in MM.

8. (a)
Effective access time = memory access time + page fault rate × (probability of empty or unmodified page × time to access empty or unmodified page + probability of page modification× time to access modified page)
Substituting these values in the formula, we get
$0.2 = 0.1 + P(0.3(8000) + 0.7\,(20{,}000))$
$P = 6.1 \times 10^{-6}$

9. (c)
Page fault occurs when required page that is to be exists not found in the memory. When the page is found in the memory it is called page hit otherwise miss. It is not necessary to page in main memory it can either in cache memory or any other level of memory.

10. (a)
It is due to Belady's Anomaly in which
More frames ⇒ more page fault

12. (d)
In optimal page replacement policy we check for the page frame that will not be used in the near future and is replaced. For the given sequence 0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7, 8, 2, 3, 8, 3,

	Page hit		Page Miss	
~~0~~ 8.....8	0	8	0	3
~~3~~ 7.....3	1	2	9	
~~1~~ 2.....2	8	7	1	
	8	8	8	
	7	2	7	
	1	8	2	
		3		

13. (d)

For the sequence

0, 2, 1, 3, 2, 1, 0, 1, 2, 1

0
2
1
3

the least recently used page will be 3

Hence option (d) is correct.

14. (c)

Page frames

~~2~~	~~1~~	~~2~~	1
~~4~~	~~5~~	~~5~~	4
~~3~~	~~3~~	~~3~~	2

Page sequence
2, 4, 3, 1, 5, 3, 2, 5, 3, 1, 4, 2

Page faults
2, 4, 3, 1, 5, 2, 1, 4, 2

Page hit
3, 5, 3

∴ Number of page faults = 9.

16. (b)

Page hit ratio = 1.

M_a = 20 ns.

Since page hit ratio = 1.

∴ Memory access time for one page = 20 ns.

Hence, option (b) is correct.

17. (c)

1, 2, 4, 5, 2, 1, 2, 4

Since 1 and 2 are already in the memory which can accomodate 3 pages

~~1~~	~~5~~	4	
~~2~~	~~2~~	2	No. of page faults = 4
~~4~~	1		

Hence option (c) is correct

18. (c)

The first 3 reference *A*, *B*, *C* fills the internal storage with *A*, *B*, *C* in 3 page transfers. Now the next reference *D* results in a page fault. So page *A* is downloaded and *D* takes its place after a page transfer. So, the internal store has *D*, *E* and *C*. The next reference is A - results in a page fault. So, a page transfer takes place and swaps *B* and *A*. Continuing this way, we find totally 9 page transfers are necessary.

19. (c)

When it tries to access 0100, it results in a page fault as the memory is empty right now. So, it loads the second page (which has the addresses 100 – 199). Trying to access 200 will result in a page fault, as it is not in memory right now. So the third page with the addresses from 200 to 299 will replace the second page in memory. Trying to access 430 will result in another page fault. Proceeding this way, we find trying to access the addresses 0510, 0120, 0220 and 0320 will all result in page faults. So, altogether 7 page faults.

20. (b)

Since the page is used heavily then it can not be removed if LFU used, since it is in constant used, hence it can't be removed if LRU is used. Using FIFO only it can be removed effectively.

21. (c)

Effective access time = (probability of success)× (memory access time) + (probability of page fault)× (average page fault service time

$= ((1 - P) \times 100 + (P) \times 50000000)$ ns

$= (10^{-7} - 10^{-7} P + 50000000 P)$ ns

$\approx 10^{-7} - 10^{-7} P(500000)$

23. (b)

If 3 page frames are used

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

Page	Sequence	
~~0~~	~~1~~	~~2~~
~~3~~	~~0~~	~~1~~
~~4~~	0	1
~~2~~	~~3~~	4

Page fault : 0, 1, 2, 3, 0, 1, 4, 2, 3
Page hit : 0, 1, 4.
∴ No. of page fault = 9.
If 4 pages frames are used

0	1	2	3
1	0	4	
2	1		
3	2		

Page Sequence			
0	1	2	3
0	1	4	0
1	2	3	4

Page faults : 0, 1, 2, 3, 4, 0, 1, 2, 3, 4
Page hit : 0, 1
∴ Number of page faults = 10
Hence, (b) is the correct option.

24. (a)

Page sequence: 0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7, 8, 2, 3, 8, 3.
Given 3 page frames are available and optimal replacement policy.

0.8	Page faults : 0, 9, 1, 8, 7, 2, 3
9 7 3	Page hit : 0, 1, 8, 8, 7, 1, 8, 2, 7, 8, 2, 8, 3
1 2	∴ No. of page fault = 7

Hence, the correct option is (a).

26. (b)

By using LRU policy:

9	8	7	9	3	0	2	9	8	3	9	2	0	9
					0	0	0	0	0	0	0	0	0
				3	3	3	3	3	3	3	3	3	3
		7	7	7	7	7	7	8	8	8	8	8	8
	8	8	8	8	8	2	2	2	2	2	2	2	2
9	9	9	9	9	9	9	9	9	9	9	9	9	9
F	F	F	H	F	F	F	H	F	H	H	H	H	H

7 faults

By using FIFO policy:

9	8	7	9	3	0	2	9	8	3	9	2	0	9
					0	0	0	0	0	0	0	0	0
				3	3	3	3	3	3	3	3	3	3
		7	7	7	7	7	7	8	8	8	8	8	8
	8	8	8	8	8	8	9	9	9	9	9	9	9
9	9	9	9	9	9	2	2	2	2	2	2	2	2
F	F	F	H	F	F	F	F	F	H	H	H	H	H

8 faults

So, [LRU + FIFO] page fault: 7 + 8 = 15

28. (15)

F	F	F	F		F	F	F	F	F		F	F	F		F	F			F
1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3
		1	2	3	4	2	1	5	6	6	1	2	3	7	6	3	3	1	2

⇒ 15 page faults

29. (17.70) [17.00-18.00]

$3\text{ ms} \le P[0.7 \times 15 + 0.3 \times 6] + (1-P) \times 1\text{ ms}$
$3\text{ ms} \le P[12.3] + (1-P) \times 1$
$3\text{ ms} \le 11.3P + 1$
$2\text{ ms} \le 11.3P$

$$P = \frac{2}{11.3} = 0.177$$

In (%) ⇒ 0.177 × 100 = 17.7%

■■■■

8 CHAPTER

File System

Q.1 Each inode in a file system has 5 direct pointers to disk blocks, 3 single-indirect pointers to disk blocks, 2 double-indirect pointers to disk blocks and nothing else. A disk block is 500 bytes. A pointer to a disk block is 10 bytes. The entire disk contains 15,000,000 bytes, at most. What is the maximum size in bytes, of a file in this file system?

(a) 2552500 bytes (b) 2575000 bytes
(c) 2575000 bytes (d) 2577500 bytes

Q.2 The data block of a very large file in the Unix file system are allocated using

(a) contiguous allocation
(b) linked allocation
(c) indexed allocation
(d) an extension of indexed allocation

Q.3 In which of the following directory systems, is it possible to have multiple complete paths for a file, starting from the root directory?

(a) Single level directory
(b) Two level directory
(c) Three structured directory
(d) Acyclic graph directory

Q.4 A unix-style Inode has 10 direct pointers, 1 double indirect, 1 triple indirect and 1 quadruple indirect pointers. Disk block size is 1 kbyte. If the maximum number of bytes of a file that can be accessed by quadruple pointer is 4 terabytes using the inode then how many bits are required to address disk block?

(a) 8 (b) 16
(c) 24 (d) 32

Q.5 In a particular Unix OS, each data block is of size 1024 bytes, each node has 10 direct data block addresses and three additional addresses: one for single indirect block, one for double indirect block and one for triple indirect block. Also, each block can contain addresses for 128 blocks. Which one of the following is approximately the maximum size of a file in the file system?

(a) 512 MB (b) 2 GB
(c) 8 GB (d) 16 GB

Q.6 Which of the following statement is false in the context of the allocation methods?

(a) Contiguous file allocation leads to external fragmentation.
(b) Linked allocation supports random access to disk block.
(c) Linked allocation does not exhibit external fragmentation.
(d) In indexed file allocation, for each file one block is used as an index to store block pointers.

Q.7 Which of the following statements are true?

S_1: Using a larger block size in fixed block size file system leads to better disk throughput but poor disk space utilization.

S_2: In index allocation of block to a file, the maximum possible size of file depends on the size of blocks, the number of blocks used for the index, and the size of the address of blocks.

(a) Only S_1 (b) Only S_2
(c) Both S_1 and S_2 (d) Neither S_1 nor S_2

Q.8 In particular Unix OS, each data block is 256 bits, each node has 4 direct data block addresses and 3 additional addresses: One for single indirect block, one for double indirect block and one for triple indirect block. Each block is addressed with 64-bit. The total size of a file possible in the file system (in k-bits) is _____.

■■■■

Answers File System

1. (c) 2. (d) 3. (d) 4. (d) 5. (b) 6. (b) 7. (c)

Explanations File System

1. **(c)**
Each pointer is 10 bytes, so each disk block can contain 500/10 = 50 pointers.
An Inode can reference 5 blocks directly, each single-indirect pointer references a block which contains 50 pointers, so a total of 150 blocks can be referenced by the 3 single-indirect pointers.
Similarly, each double-indirect pointer references a block of 50 pointers, each of which references of a block of 50 pointers, for 2,500 blocks in total. There are 2 double-indirect pointers, so a total of 5,000 blocks referenced by them
Total: 5 + 150 + 5000 = 5155 blocks
Which is 5155 × 500 = 2,577,500 bytes

3. **(d)**
In Acyclic Graph directory file can be accessed using different paths from the same root directory, this can be done with the helps of soft links.

5. **(b)**
If 2 indirect pointer then entry will be 2 times and 3 indirect pointer then entry will 3 times.
No. of entries = 128 = 2^7 and block size = 1 KB
Now maximum file size
= 10×1 KB + $2^7 \times 2^7 \times 1$ KB + $2^7 \times 2^7 \times 2^7 \times 1$ KB
(It is too big then other will neglect)
= $2^{21} \times 2^{10}$ B = 2 GB

6. **(b)**
- Contiguous file allocation results in external fragmentation because it could be the case that there are many small unallocated fragments between files and they can't be used to allocate a file whose size is greater than the size of any fragment but less than total size of free space.
- Linked allocation supports sequential access to disk block but not random access.
- Linked allocation does not suffer from external fragmentation.
- In indexed file allocation, one block is used for each file as an index to store only block pointers.

7. **(c)**
S_1 : For better throughput larger block size used in fixed block size file system but it results in decrease in the space utilization.
S_2 : Number of DBA's possible in one disk block
$= \frac{\text{DBA Size}}{\text{DBA}}$.

8. **(22)**
Total file size

$$= \left[\text{Direct DBA} + \#\left(\frac{\text{Data block size}}{\text{DBA}}\right) + \#\left(\frac{\text{Data block size}}{\text{DBA}}\right)^2 + \#\left(\frac{\text{Data block size}}{\text{DBA}}\right)^3\right] \times \text{Data block size}$$

Data block size = 256 bits

$\left(\frac{\text{Data block size}}{\text{DBA}}\right)$ = # Disk block address store inside one block
Maximum file size
= [4+1*(4)+1*$(4)^2$ + 1*$(4)^3$] × 256 bits
= 22 kbits

■■■■

9 CHAPTER
Input-Output System

Q.1 Which of the following is not a disk scheduling algorithm?
(a) FCFS (b) LRU
(c) SSJF (d) SCAN

Q.2 Suppose that a disk drive has 5000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending request, in FIFO order is 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130 starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests, for each of the following disk scheduling algorithms?
- FCFS
- SSTF
- SCAN

(a) 7081, 1745, 9769 (b) 7081, 1745, 7081
(c) 1745, 1004, 9600 (d) 9769, 1745, 7081

Q.3 Consider a program to be run on a computer system in round-robin CPU scheduling. The size of the program is 100k. It is given that hard disk has a transfer rate of 1 megabyte per second. Assume that there are no head seeks and average latency is 8 milliseconds. What could be the acceptable time quantum for effective CPU utilization?
(a) 2.048 sec (b) 0216 sec
(c) 0.108 sec (d) 0.100 sec

Q.4 Match **List-I** with **List-II** select the correct answer using the codes given below the lists:

List-I	List-II
A. Disk scheduling	1. Round robin
B. Batch processing	2. SCAN
C. Time sharing	3. LIFO
D. Interrupt processing	4. FIFO

Codes:

	A	B	C	D
(a)	3	4	2	1
(b)	4	3	2	1
(c)	2	4	1	3
(d)	2	1	4	3

Q.5 Disk requests come to a disk driver for cylinders 98, 183, 37, 122, 14, 124, 65, 67 in that order at a time when the disk drive is reading from cylinder 53. The total seek time, if the disk arm scheduling algorithm is shortest seek time first, is
(a) 236 (b) 331
(c) 299 (d) None of these

Q.6 Suppose that the head of a moving head disk with 192 tracks, numbered 0 to 191, is currently serving a request at track 80 and has just finished at track 62. The queue of the request is kept in the FIFO order: 119, 58, 114, 28, 111, 55, 103, 30, 75. What is the total number of tracks traversed by head movements needed to satisfy these request for the Elevator disk-scheduling algorithms?
(a) 143 (b) 130
(c) 177 (d) 547

Q.7 Disk requests come to a disk driver for cylinders in the order 10, 22, 20, 2, 40, 6 and 38, at a time when the disk drive is reading from cylinder 20. The seek time is 6 ms per cylinder. The total seek time, if the disk arm scheduling algorithm is first-come-first-served is
(a) 360 ms
(b) 850 ms
(c) 900 ms
(d) None of these

Q.8 A certain moving arm disk storage with one head has following specifications:

Number of tracks/recording surface = 200
Disk rotation speed = 2400 rpm
Track storage capacity = 62500 bits

The average latency time (assume that the head can move from one track to another only by traversing the entire track) is

(a) 2.5 s (b) 2.9 s
(c) 3.1 s (d) 3.6 s

Q.9 On a disk with 1000 cylinders, numbers 0 to 999, compute the number of tracks the disk arm must move to satisfy all the requests in the disk queue. Assume the last request serviced was at tracks 345 and the head is moving towards track 0. The queue in FIFO order contains requests for the following tracks. 123, 874, 692, 475, 105, 376. Perform the computation using C-SCAN scheduling algorithm. What is the total distance?

(a) 1219 (b) 1009
(c) 1967 (d) 1507

Q.10 A disk has 200 tracks (numbered 0 through 199). At a given time, it was servicing the request of reading data from track 120, and at the previous request, service was for track 90. The pending requests (in order of their arrival) are for track numbers.

30, 70, 115, 130, 110, 80, 20, 25

How many times will the head change its direction for the disk scheduling policies SSTF (Shortest Seek Time First) and FCFS (First Come First Serve)

(a) 2 and 3 (b) 3 and 3
(c) 3 and 4 (d) 4 and 4

Q.11 A hard disk system has the following parameters:

Number of track = 500
Number of sectors/track = 100
Number of bytes/sector = 500

Time taken by the head to move from one track to adjacent track = 1ms

Rotation speed = 600 rpm.

What is the average time taken for transferring 250 bytes from the disk?

(a) 300.5 ms (b) 255.5 ms
(c) 255 ms (d) 300 ms

Q.12 Consider the disk with read/write head is currently located at track 45 (of track 0 – 255) and moving in the positive direction. Assume that the following track requests have been made in the order: 40, 67, 11, 240, 87.

What is the total seek time using C-SCAN (assume one cylinder to adjacent cylinder time is 2 msec) ?

(a) 1015 (b) 1010
(c) 1020 (d) 1030

Q.13 A manufacturer wishes to design an array of hard disks in a server with a capacity of 512 GB (using the standard definition of 1 GB = 2^{30} bytes). If the technology used to manufacture the disks allows 2048 byte/sectors, 4096 sectors/track, and 8192 tracks/platter, how many minimum disks are required, assuming two platters/disk? (Assume a fixed number of sectors per track).

Q.14 Disk request come into disk driver for cylinders 10, 22, 20, 2, 6, 40 and 31 in that order. The disk head is currently positioned over 30. A seek takes 6 msec per cylinder moved. The total seek time using shortest seek time first is _____ (in msec) [If the difference between requests are matching then read/write header moves towards positive direction].

Q.15 Consider the following characteristics of a disk system.

- 2 GB disk rotate at 5000 rpm.
- Data transfer rate of 10^6 bytes/sec.
- Average seek time is 10 msec.
- Block size is 32 kB.

What is the average service time to retrieve a single block from a random location on the disk (Approx)?

(a) 30 msec (b) 31 msec
(c) 32 msec (d) 48 msec

■■■■

Answers Input-Output System

1. (b) 2. (a) 3. (b) 4. (c) 5. (a) 6. (d) 7. (d) 8. (a) 9. (c)
10. (c) 11. (d) 12. (b) 15. (d)

Explanations Input-Output System

2. (a)
The total seek time is sum of difference between successive head positions.

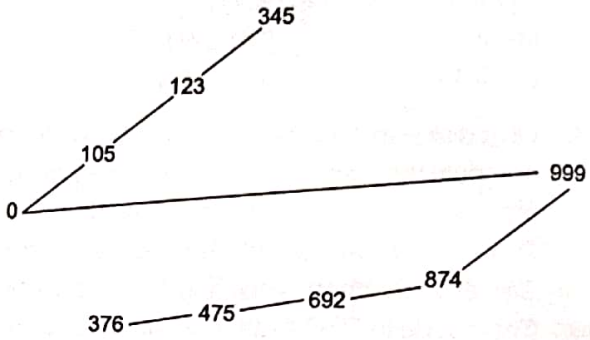
3. (b)
Whenever a program is to be executed, it needs to be brough in to the memory from disk, where it is residing. When the program needs to be swapped out to disk, again some time is required.
For effective CPU utilization, the time quantum must be substantially greater than the context switch time.
Time required to transfer process from memory = 100k/1000k = 1/10 seconds = 100 ms
Add average latency to above it becomes 108 milliseconds for one way transfer between disk and memory for the given program. The time required two way transfer is 216 ms.

5. (a)
Disk requests are
98, 183, 37, 122, 14, 124, 65, 67
Total seek time = Time covered by the disk arm starting from 53.
= (65 – 53) + (67 – 65) + (67 – 37) + (37 – 14) + (98 – 14)+(122 – 98) + (124 – 122)+(183 – 124)
= 236

7. (d)
The disk drive has to traverse totally 146 cylinders.
So, seek time is 6 × 146 = 876 ms.

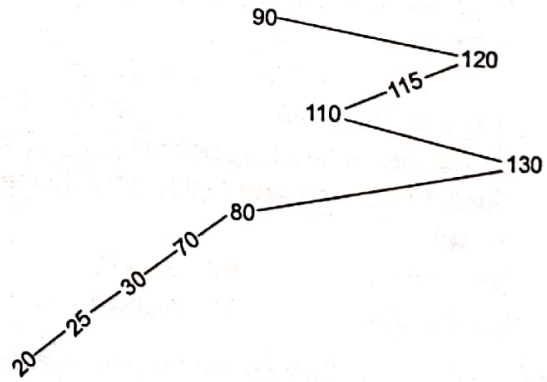
8. (a)
To cover 2400 × 62500 bits, 60 s are needed.
Average latency time is the time needed to traverse 100 tracks i.e. 100 × 62500 bits, which is 2.5 s.

9. (c)



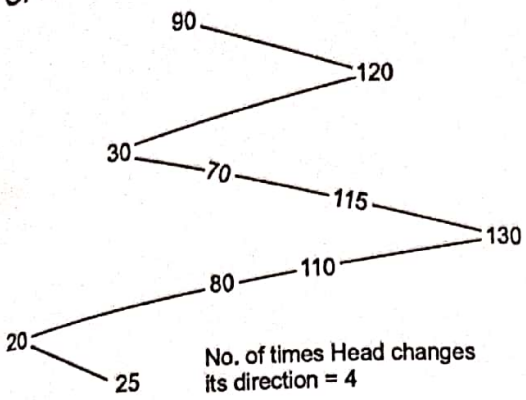
∴ Total distance covered by the head
= (345 – 123) + (123 – 105) + (105 – 0) + (999 – 0) + (999 – 874)+(874 – 692)+(692 – 475)+(475 – 376)
= 222 + 18 + 105 + 999 + 125 + 182 + 217 + 99
= 1967
Hence (c) is correct option.

10. (c)
SSTF



∴ Head changes its direction for SSTF = 3

FCFS



Hence, (c) is the correct option.

11. (d)

track = 500, sector = 100

capacity of sector = 500

$$\text{Rotational speed} = \frac{600}{60}\text{r/sec}$$

$$\text{1 second rotation} = \frac{60}{600}$$

Seek time = 1 ms

$$\text{Average seek time} = \frac{499}{2} = 249.5$$

$$\text{Data transfer time} = \frac{\text{Track capacity}}{\text{one complete rotation time}}$$

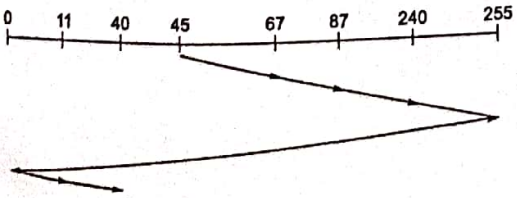
$$= \frac{100\times500}{\left(\frac{60}{600}\right)} = 5\times10^5 \text{ byte/sec}$$

$$\text{Block transfer time} = \frac{250}{5\times10^5 \text{ byte/sec}} = 0.5 \text{ ms}$$

Total transfer time = Seek time + data transfer time + block transfer time

= 249.5 + 0.5 + 50 = 300 ms

12. (b)



Total seek time = [(67 – 45)+(87 – 67)+(240 – 87) + (255 – 240) + (255 – 0) + (11 – 0)+(40 – 11)] × 2 msec

= [22 + 20 + 153 + 15 + 255 +11 + 29] × 2 msec

= [505] × 2 msec

= 1010 msec

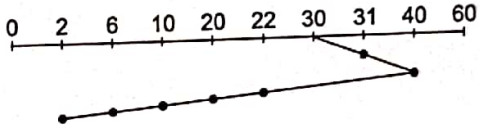
13. (4)

Multiplying bytes per sector, the sectors per track and the tracks per platter gives a capacity of 64 GB (8×2^{30} bytes) per platter.

i.e., $2040\times4096\times8192$ bytes = $2^{11}\times2^{12}\times2^{13}$ bytes = 2^6 GB = 64 GB

Therefore, 2 platters give 128 GB per disk. Number of disks in a server = 4 or more to give a total server capacity of 512 GB.

14. (288)



= ([31 – 30] + [40 – 31] + [40 – 22] + [22 – 20] + [20 – 10] + [10 – 6] + [6 – 2]) × 6

= (1 + 9 + 18 + 2 + 10 + 4 + 4) × 6 msec

= 288 msec

15. (d)

$$T_{seek} = 10 \text{ msec}$$

$$T_{\text{rotational time}} = \frac{1}{2}(\text{Rotational latency})$$

$$= \frac{1}{2}\left(\frac{60}{5000}\right) = \frac{3}{500}\text{sec} = 6 \text{ msec}$$

$$T_{transfer} = \frac{32\times1024}{10^6} = 32 \text{ msec}$$

$$T_{service} = T_{seek} + T_{rotational} + T_{transfer}$$

= 10 msec + 6 msec + 32 msec

= 48 msec

■■■■