Branch : CSE & IT Batch : Hinglish

Operating Systems

Intro & Background

DPP 01

[MCQ]

1. Consider the following statements:

(i) Operating system implementation has architectural constraints.

(ii) Operating system is a set of utilities.

Which of the following is true?

(a) Only (i) is correct.

(b) Only (ii) is correct.

(c) Both (i) and (ii) are correct.

(d) Neither (i) nor (ii) are correct

[MCQ]

2. Operating system working on strict deadlines are _____.

(a) Time-sharing OS.

(b) Real-time OS.

(c) Network operating system.

(d) Centralized OS.

[MCQ]

3. What is/must be the primary goal for hard real time operating system?

(a) Robustness.

(b) Reliability.

(c) Efficiency.

(d) Convenience.

[MSQ]

4. Which of the following statement is/are correct?

(a) During Booting, OS is loaded into disk from main memory.

(b) The area of memory where OS is stored, is known as system area.

(c) Uni-programming can load and execute a single program in memory.

(d) Uni-programming suffers from idleness of CPU.

[MCQ]

5. Throughput is _____.

(a) Executing multiple programs in memory.

(b) Total number of programs completed per unit time.

(c) Total number of programs loaded into main memory.

(d) None of the above.

[MCQ]

6. Pre-emptive processes have _____.

(a) Forceful deallocation.

(b) Better response time.

(c) Both (a) and (b).

(d) None of the above.

[MCQ]

7. Consider the following statements:

(i) Non-preemptive process can lead to starvation.

(ii) Non-preemptive process has good response time.

(iii) Non-preemptive process can release CPU voluntarily.

Which of the following is correct?

(a) (i) and (ii) are correct.

(b) (ii) and (iii) are correct.

(c) (i) and (iii) are correct.

(d) All (i), (ii) and (iii) are correct.

[MSQ]

8. Function of operating system includes _____.

(a) Resource Management.

(b) Reliability.

(c) Security.

(d) Control over system performance.

Answer Key

1. **(c)**
2. **(b)**
3. **(c)**
4. **(b, c, d)**
5. **(b)**
6. **(c)**
7. **(c)**
8. **(a, c, d)**

Hints & Solutions

1. (c)

Operating system is implemented purely for a particular architecture. For example, Smartphones OS cannot be implemented over a desktop. Operating system is a set of utilities which simplifies application development.

So, option (c) is correct.

2. (b)

Real-time OS operates on strict deadlines and have time constraints. For example:

Fire alarm system, heart pacemaker, air traffic control system, Missile control, etc.

3. (c)

Efficiency is the primary goal of real time OS, if system is little bit inefficient it can cause heavy loss and destruction.

4. (b, c, d)

During Booting, OS is loaded into main memory from the disk, and the area where OS is stored in main memory is known as system area. So, option (a) is incorrect and option (b) is correct.

Uniprogramming OS has the ability to load and execute a single program in memory. so, option (c) is correct.

In Uniprogramming if one program is loaded into main memory, then no other program can be loaded and CPU can remain idle if that single process is busy in performing IO. So, option (d) is also correct.

5. (b)

Executing multiple programs in memory is multiprogramming.

So, option 'a' is incorrect.

Total number of programs/applications completed per unit is throughput.

So, option (b) is correct.

6. (c)

Pre-emptive process has forceful deallocation and it improves response time of the processes.

So, (c) is correct.

7. (c)

(i) Non-preemptive process can lead to starvation. Correct.

(ii) Non-preemptive process has good response time. Incorrect.

(iii) Non-preemptive process can release CPU voluntarily. Correct.

$\therefore$ (c) is correct option.

8. (a, c, d)

Functions of operating system include resource management, security, controlling and monitoring overall system's health to improve performance.

Whereas reliability is a goal of operating system.

Branch : CSE & IT Batch : Hinglish

Operating Systems
Intro & Background

DPP 02

[MSQ]

1. Which of the following are allowed in kernel mode only?

(a) Enabling / Disabling interrupts.
(b) Reading the system's time.
(c) Context switching.
(d) Clear the memory.

[MSQ]

2. Which of the following is/are correct?

(a) To switch from kernel mode to user mode, the mode bit should be changed to 1.
(b) To switch from kernel mode to user mode, the mode bit should be changed to 0.
(c) To switch from user mode to kernel mode, the mode bit should be changed to 1.
(d) To switch from user mode to kernel mode, the mode bit should be changed to 0.

[MCQ]

3. Match the following:

(i)	User mode	1)	Atomic
(ii)	Kernel mode	2)	Mode bit 0
		3)	Privileged
		4)	Preemptive

(a) (i) – 2, 3; (ii) – 1, 4
(b) (i) – 2, 1; (ii) – 3, 4
(c) (i) – 2, 1, 4; (ii) – 3
(d) (i) – 4; (ii) – 1, 2, 3

[MCQ]

4. System call is used to access_____.

(a) Application functionality.
(b) I/O functionality.
(c) Operating system functionality.
(d) None of these.

[MCQ]

5. Consider the following statements:

(i) Pre-defined functions start executing in kernel mode.
(ii) User-defined functions start executing in user mode.

Which of the following is correct?

(a) Only (i) is correct.
(b) Only (ii) is correct.
(c) Both (i) and (ii) are correct.
(d) Both (i) and (ii) are incorrect.

[NAT]

6. Consider the following program:

```
main()
 {
    fork()
    fork()
    fork()
    printf("GATE "2023");
    return 0;
 }
```

How many times GATE 2023 will get printed?

[MCQ]

7. Mode bit is present in _____.

(a) Main Memory
(b) Disk
(c) Cache
(d) Register

[MCQ]

8. At system boot time, the hardware starts in _____.

(a) Kernel mode
(b) User mode
(c) Operating mode
(d) Disk mode

Answer Key

1. **(a, c, d)**
2. **(a, d)**
3. **(d)**
4. **(c)**
5. **(b)**
6. **(8)**
7. **(d)**
8. **(a)**

Hints & Solutions

1. (a, c, d)

The instructions that run in kernel mode are known as privileged instructions.
Privileged instructions include:
Enabling/ disabling of interrupts,
Context switching
Clear the memory or remove any process from memory.
Whereas, reading the system's time is a non-privileged instruction and can be performed in user-mode.
∴ (a), (c) and (d) are correct.

2. (a, d)

To switch from kernel mode to user mode, the mode bit should be 1, and
To switch from user mode to kernel mode, the mode bit should be 0.
Therefore, option (a) and (d) are correct.

3. (d)

User mode is non-atomic, preemptive and non-privileged mode with mode bit set to 1.
Kernel mode is atomic non-preemptive and privileged mode with mode bit set to 0.
∴ option (d) is correct

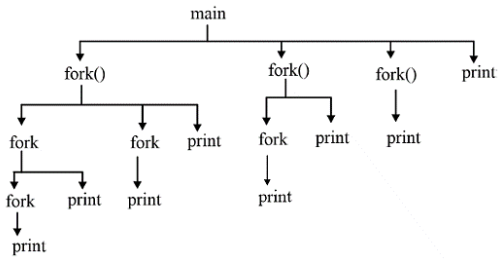
4. (c)

System call is used to access operating system's functionality.

5. (b)

All user-defined and pre-defined function start executing in user-mode.

6. (8)



Total 8 times printf is executed and "GATE2023" is printed.

7. (d)

Mode bit is stored in program status word (PSW) register.

8. (a)

During booting, hardware starts in kernel mode.

Branch : CSE & IT Batch : Hinglish

Operating System

Process Management

DPP 01

[MCQ]

1. Consider the following statements:

(i) In Uni-programming, CPU can run only one program at a time.

(ii) In Multi-programming, CPU can run multiple program at a time.

Which of the following is correct?

(a) (i) is correct (ii) is incorrect.
(b) (i) is incorrect (ii) is correct.
(c) Both (i) and (ii) are correct.
(d) Both (i) and (ii) are incorrect.

[MSQ]

2. Which of the following is/are incorrect?

(a) Multiprogramming enhances CPU throughput.
(b) Non-preemptive processes may have forceful deallocation.
(c) Non-preemptive multiprogramming is equivalent to uni-programming.
(d) Multiprogramming is of two types preemptive and non-preemptive.

[MCQ]

3. (Fork) system call is converted by compiler and also know as _____.

(a) Supervisory call.
(b) Privileged instruction.
(c) Software interrupt instruction.
(d) All of the above.

[MCQ]

4. Consider the following statement:

(i) User program gets blocked when fork call is executed.

(ii) Dispatch table contains information regarding all system call.

(iii) Fork system call can execute non-atomically in system.

Which of the following is/are correct?

(a) Only (i) is correct.
(b) (i) and (ii) are correct.
(c) (i) and (iii) are correct.
(d) (ii) and (iii) are correct.

[MSQ]

5. Which of the following is/are correct?

(a) Changing mode from user to kernel need an interrupt.
(b) Changing mode from kernel to user needs an interrupt.
(c) ISR (Interrupt Service Routine) is responsible for mode shifting (user to kernel)
(d) ISR (Interrupt Service Routine) is responsible for changing the mode kernel to user.

[MCQ]

6. Consider the following statements:

(i) Static data have fixed size and memory allocated during run-time.

(ii) Dynamic data have different size and memory allocated during load-time.

Which of the following is/are correct?

(a) Only (i) is correct.
(b) Only (ii) is correct.
(c) Both (i) and (ii) are correct.
(d) Both (i) and (ii) are incorrect.

[MCQ]

7. Match the following

(i) Program	1. Active entity
(ii) Process	2. Resides in main memory
	3. Passive entity
	4. Resides in disk

(a) (i) – 1, 3 (ii) – 2, 4
(b) (i) – 2, 4 (ii) – 1, 3
(c) (i) – 1, 2 (ii) – 1, 4
(d) (i) – 3, 4 (ii) – 1, 2

[MCQ]

8. Program counter holds _____.

(a) Address of previous executed instruction.
(b) Address of current executing instruction.
(c) Address of next instruction to be executed.
(d) None of these.

Answer Key

1. **(a)**
2. **(b, c)**
3. **(d)**
4. **(b)**
5. **(a, c)**
6. **(d)**
7. **(d)**
8. **(c)**

Hint & Solutions

1. (a)

(i) In uni-programming, CPU can load and run only one program at a time.

(ii) In multi-programming, CPU can load multiple program but only one program can run at a time.

∴ (i) is correct and (ii) is incorrect.

2. (b, c)

(a) Multiprogramming enhances CPU throughput. **Correct**.

(b) Non-preemptive processes do not have forceful deallocation. So, 'B' is **Incorrect.**

(c) Non-preemptive multiprogramming is not equal to uni-programming. **Incorrect**.

(d) Multiprogramming is of two types preemptive and non-preemptive.

3. (d)

(Fork) System call is converted by compiler for execution and it is also known as supervisory call, privileged instruction, or software interrupt instruction. Therefore, option (d) is correct.

4. (b)

Fork is a system call and has more privilege, therefore user program gets blocked when CPU is executing fork. So 'i' is correct. Dispatch table has the information regarding all system calls along with their address. So, (ii) is correct.

Fork is a system call and executed in kernel mode, so it has to execute atomically. So, (iii) is incorrect.

5. (a, c)

Changing mode from user to kernel mode needs an interrupt and it blocks the user process, then execute the system service.

ISR is responsible for mode shifing from user to kernel. Whereas, there is no need of any interrupt to change the mode form kernel to user as the system is in already kernel mode and to change the mode from kernel to user, last instruction of system call is used.

∴ (a) and (c) are correct.

6. (d)

Static data have fixed size and memory is allocated before run-time or during load time.

Dynamic data is size is unknown and memory is allocated during run-time.

∴ Both (i) and (ii) are incorrect.

7. (d)

Program is a passive entity and resides in disk.

Process is an active entity and resides in main memory.

8. (c)

Program counter holds address of the next instruction to be executed.

Branch : CSE & IT Batch : Hinglish

Operating System

Process Management

DPP 02

[MCQ]

1. Consider the following statements:

(i) Program when utilizing computer's resources is known as process.

(ii) A program performing, I/O operation is considered as a process.

Which of the following is correct?

(a) Only (i) is correct.
(b) Only (ii) is correct.
(c) Both (i) and (ii) are correct.
(d) Both (i) and (ii) are incorrect.

[MSQ]

2. Which of the following is/are correct?

(a) A process is an instance of a program.
(b) A program loaded into main memory but it is not running currently is known as process.
(c) Program is created from a process.
(d) Process is created from a program.

[MCQ]

3. Match the following:

(i)	Context switching	1. Dispatcher
(ii)	Message passing	2. Long term Scheduler
(iii)	Degree of multiprogramming	3. Process creation
(iv)	Fork()	4. Inter process communication

Select the correct code from the following:

(a) (i)- 4, (ii)- 1, (iii)- 2, (iv)- 3
(b) (i)- 4, (ii)- 1, (iii)- 3, (iv)- 2
(c) (i)- 1, (ii)- 4, (iii)- 3, (iv)- 2
(d) (i)- 1, (ii)- 4, (iii)- 2, (iv)- 3

[MCQ]

4. The memory area where all instructions of a program is stored is known as_____.

(a) Heap (b) Program Block
(c) Code section (d) Instruction space.

[MCQ]

5. Runtime stack contains _____ of function calls.

(a) Code
(b) Activation record
(c) Dynamic data
(d) Instructions

[NAT]

6. How many of the following operations can be performed on a process?

(i) Creation
(ii) Dispatch
(iii) Execute
(iv) Block
(v) Terminate

[MSQ]

7. Which of the following is/are correct?

(a) A suspended process resides in memory.
(b) A blocked process resides in memory.
(c) A suspended process resides on disk.
(d) A blocked process resides on disk.

[MCQ]

8. Information stored inside a process control block is known as_____ of process.

(a) Schedule
(b) Details
(c) Context.
(d) Data

Answer Key

1. **(c)**
2. **(a, b, d)**
3. **(d)**
4. **(c)**
5. **(b)**
6. **(5)**
7. **(b, c)**
8. **(c)**

Hint & Solutions

1. (c)

A program under execution is known as a process. So when a program is utilizing computer's resources it means the program is under execution and it will be known as a process.

Similarly, I/O is a computer's resource and if a program performing I/O it means the program is under execution and hence it is considered as a process.

Therefore, both (i) and (ii) are correct.

2. (a, b, d)

Program is like a class, and process is like an object. So, "A" is correct.

A program loaded into main memory but it is not running currently is known as a process. When program is loaded from disk to main memory by operating system, it becomes process. So, "B" is correct.

Program is not created from any process. Process is created from a program. So, "C" is incorrect and "D" is correct.

3. (d)

Context Switching: Whenever a process is dispatched to the running state by the dispatcher, we need to perform context switching (loading PCB).

Degree of Multiprogramming: Long term scheduler is responsible for creating new processes to main memory so it controls the degree of multiprogramming.

Message passing: Inter-process communication (IPC) uses message passing method to communicate with each other.

Fork (): System call is used to create a new child process.

4. (c)

The memory area where all instructions of a program is stored is known as code section or text section.

5. (b)

Runtime stack contains activation records of function calls. Activation record further contains memory for formal parameters, local variables, return address.

6. (5)

All are operations performed on a process.

(i) Creation- Creation of the process or allocating resources to the process.

(ii) Dispatch- Dispatch/scheduling the process. Selecting which process will run on CPU.

(iii) Execute- Execute/ running the process. When process is executing or implementing it's instruction in the CPU.

(iv) Block- When process needs to perform I/O, then it is blocked and it is in wait state.

(v) Terminate- When process has completed its instruction's execution. Resource deallocation is also known as termination operation. When process has completed its execution in CPU.

7. (b, c)

A blocked process resides in memory- A process when blocked remains in memory and perform operation like I/O, etc.

Whereas, when a process is suspended it resides on disk. And this is the difference between block and suspend operation.

8. (c)

Information stored inside a process control block is known as context of process.

Branch : CSE & IT Batch : Hinglish

Operating System

Process Management

DPP 03

[MCQ]

1. The information about a process is maintained in _____.

(a) Process Context block which is implemented using an Array.
(b) Process Control block which is implemented using a Stack
(c) Process Context block which is implemented using a Linked list
(d) Process Control block which is implemented using a Doubly Linked list

[MSQ]

2. Consider a computer with a CPU and N processes, which of the following is correct regarding the processes:

(a) There can be maximum of N processes in the block state.
(b) There can be maximum of N processes in the ready state.
(c) There can be atmost 1 process in the running state.
(d) There can be atmost N processes in the running state.

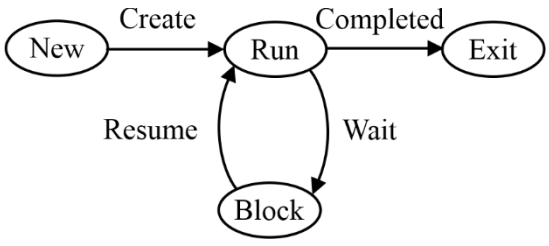
[MCQ]

3. If "Dispatch" operation is performed on the process, which transition is possible from the following?

(a) Running to Ready
(b) Ready to Running
(c) Blocked to Ready
(d) Running to Blocked.

[MCQ]

4. The given process state transition diagram represents:



(a) UNIX operating system.
(b) Multiprogramming operating system.
(c) Uni-programming operating system.
(d) None of these.

[MSQ]

5. If the process is in Main memory, then, it can be in

(a) Ready state
(b) Running state
(c) Block state
(d) Suspend state

[MCQ]

6. If a process is suspended from Running state, it is moved to _____.

(a) Ready state in main memory.
(b) Block state in secondary memory.
(c) Suspend ready state in main memory.
(d) Suspend ready state in secondary memory.

[MSQ]

7. Which of the following scheduling queues are present on the disk?

(a) Ready queue
(b) Block queue
(c) Suspend queue
(d) Input queue

[MSQ]

8. Which of the following process state transition/ transitions is/are present in multi-programming OS but not in uni-programming OS?

(a) Running to Exit
(b) Ready to Running
(c) Block to Ready
(d) Running to Ready

Answer Key

1. **(d)**
2. **(a, b, c)**
3. **(b)**
4. **(c)**
5. **(a, b, c)**
6. **(d)**
7. **(c, d)**
8. **(b, c ,d)**

Hint & Solutions

1. (d)

The information about a process is maintained in a Process Control Block. Process Control Block is generally implemented using Doubly-linked list. So, option D is correct answer.

2. (a, b, c)

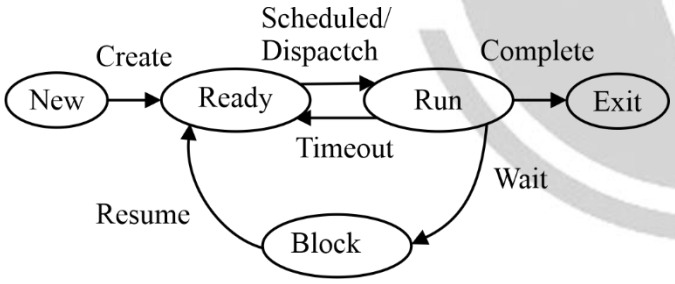
There can be N number of process present in block state. So, A is correct.

Ready state comprises a queue to store and schedule the processes on the CPU, therefore, there can be a maximum of N processes in the ready state. Hence, option B is correct.

When process is actually executing on the CPU that time the process is in running state. A single CPU can run at most 1 process at a time. Therefore, C is the correct option and D is incorrect.

3. (b)

In the process transition diagram, if the process is moving from ready to running state, means the process is dispatched or scheduled on the CPU.



Therefore. Option B is correct.

4. (c)

The given operating system do not have "READY state" in it, so a single process is created and loaded into CPU for execution. The above process state transition diagram represents Uni-programming operating system.

Unix operating system is also a multi-programming operating system, and the multi-programming operating system, has a "READY state" to schedule multiple processes to the CPU.

Therefore, option "C" is the correct answer.

5. (a, b, c)

If the process is in Ready state, Running state, Block state, then it is in main memory. Whereas if the process is in suspend state, it means it is in the disk until the main memory gets free.

Therefore, option a, b, c are correct.

6. (d)

When a process is suspended it is moved to suspend ready state in secondary memory on disk.

7. (c, d)

Suspend queue and Input/Job queue are present on disk. Ready queue and Block queue are present on main memory.

8. (b, c, d)

As, Ready state is not present in the uni-programming operating system, so

- Ready to Running
- Block to Ready
- Running to Ready

All these transitions are present in Multi-programming operating system but not in Uni-programming OS.

Branch : CSE & IT Batch : Hinglish

Operating System

Process Management

DPP 04

[MCQ]

1. Long term scheduler operates on _____.

(a) Input queue (b) Job queue
(c) Ready queue (d) Device queue

[MSQ]

2. Which of the following scheduler works between disk and main memory?

(a) Long term scheduler
(b) Medium term scheduler
(c) Short term scheduler
(d) CPU scheduler

[MCQ]

3. The time spent by a process inside a CPU is known as ____.

(a) Waiting time
(b) Turnaround time
(c) Completion time
(d) Burst time

[MCQ]

4. Context switching is done by ____.

(a) Operator (b) Scheduler
(c) Dispatcher (d) Loader

[NAT]

5. Consider a system with 5 CPU's and 20 processes. Suppose X is the maximum number of processes that can be in the running state, Y is the minimum number of processes in the ready state, Z is the maximum number of processes in the block state. Calculate X + Y*Z?

[MCQ]

6. Which scheduler plays important role in "swapping"?

(a) Long term scheduler
(b) Medium term scheduler
(c) Short term scheduler
(d) None of these

[MSQ]

7. _____, is the time when process first enters into ready queue.

(a) Submission time.
(b) Arrival time
(c) Dispatch time
(d) Load time

[MCQ]

8. If n is the total time spent by a process since its arrival till its completion, and s amount of time process spent within CPU, it also requires to perform some input-output activity and there the process spent x amount of time. Which of the following is correct expression for the total time spent by process in Ready queue?

(a) s – (n+ x) (b) x + (s- n)
(c) n + (s- x) (d) n – (s+ x)

Answer Key

1. **(b)**
2. **(a, b)**
3. **(d)**
4. **(c)**
5. **(5)**
6. **(b)**
7. **(a, b)**
8. **(d)**

Hint & Solutions

1. (b)

Long term scheduler operates on Job queue and decides which program to be loaded in main memory. Long term scheduler decides which program should be fetched into main memory from Job queue.

Input queue: It is a collection of processes in storage that are waiting to be brought into memory to run a program.

Job queue: This queue keeps all the processes in the system.

Ready queue: This queue keeps a set of all processes residing in main memory, ready and waiting to execute.

Device queue: This queue contains the processes which are waiting for the completion of I/O request.

2. (a, b)

A long-term scheduler is a scheduler that is responsible for bringing processes from the JOB queue (or secondary memory) into the READY queue (or main memory). In other words, a long-term scheduler determines which programs will enter into the RAM for processing by the CPU.

Medium term scheduler operates between Suspend queue (in disk) and block/wait state or ready state (in main memory).

Short term scheduler or CPU scheduler works between ready queue (in main memory) and running state (in main memory).

Therefore, a, b are correct option.

3. (d)

The time spent by a process inside a CPU or the time a process run on the CPU is known as Burst time.

4. (c)

Context switching is an activity carried out by Dispatcher that involves saving the PCB of process leaving the CPU and loading the PCB of next ready process onto CPU. Total time required to load and save the PCB is known as context switching time.

5. (5)

Maximum 5 processes can run on 5 CPUs at a time. So, X = 5

Minimum there can be 0 or no process present in ready state. So, Y= 0

Maximum there can be 20 processes present in the block queue. So, Z = 20

Therefore, X + Y*Z = 5 + 0*20 = 5

6. (b)

Medium term scheduler is also known as "Swapping scheduler". Its main objective is to swap out the suspended processes from the main memory. It operates on Suspend queue to swap-out and swap-in processes.

7. (a, b)

Arrival time/ Submission time is the time when process first enters the system. It is the arrival time when process is selected by Long term scheduler from new state and sent to ready queue. Arrival time is also known as submission time, therefore a, b both are correct.

8. (d)

The total time spent by a process since its arrival till its completion is known as turn around time. So, Turnaround time = n.

Amount of time spent within CPU is known as burst time. So, burst time = s.

Amount of time spent by a process in performing I/O activity is known as IO Burst time. So, IOBT = x

The question is asking, the total time spent by process in Ready queue, which means total waiting time.

WT = Turnaround time – (Burst time + IOBT)

WT = n – (s+ x)

Therefore, option D is the correct answer.

Any issue with DPP, please report by clicking here:- https://forms.gle/t2SzQVvQcs638c4r5

For more questions, kindly visit the library section: Link for web: https://smart.link/sdfez8ejd80if

PW Mobile APP: https://smart.link/7wwosivoicgd4

Branch : CSE & IT Batch : Hinglish

Operating System

CPU Scheduling

DPP 01

[MCQ]

1. Process can complete its execution from ____

(a) Ready state (b) Running state

(c) Block State (d) All of the above

[MCQ]

2. Consider a system with n number of processes, s is time when each process terminates, and t is the time each process arrives to the ready queue. What will be the average time for n processes to complete their execution?

(a) $\sum_{i=1}^{n} \frac{(t_i - s_i)}{n}$ (b) $\sum_{i=1}^{n} \frac{(n_i - t_i)}{s}$

(c) $\sum_{i=1}^{n} \frac{(s_i - n_i)}{t}$ (d) $\sum_{i=1}^{n} \frac{(s_i - t_i)}{n}$

[MSQ]

3. Which of the following statement is/ are INCORRECT regarding total number of schedules if the system has n number of processes?

(a) Total number of schedules possible in non-preemptive system are n.

(b) Total number of schedules possible in non-preemptive system are n^2.

(c) Total number of schedules possible in preemptive system are n^2.

(d) Total number of schedules possible in non-preemptive system are n!.

[NAT]

4. Consider a system with three non-preemptive processes, first process arrives at time unit 4 and executes for 12 ms, while first process was executing second process arrives at time 6 with burst time 8 ms and third process arrives at time 19 with burst time 8 ms. Assume X is the schedule length of the above system and Y is the amount of time CPU is idle and waiting for processes, calculate X + Y?

Note: CPU time started at 0.

[MSQ]

5. Which of the following statements are CORRECT regarding process state transition diagram?

(a) A process can move from ready to suspend blocked state.

(b) A blocked process can move to suspend ready state.

(c) A blocked process can move to running state.

(d) A running process can move to blocked state.

[NAT]

6. Consider the following processes, with their respective arrival time and burst times:

Processes	Arrival Time	Burst Time
P1	3	4
P2	5	6
P3	2	5
P4	0	3

Calculate the average turnaround time of all the processes scheduled using First come first serve algorithm. [Upto two decimal places]

Note: Scheduling overhead is negligible

[NAT]

7. Consider the following processes, with their respective arrival time and burst times:

Processes	Arrival Time	Burst Time
P1	2	5
P2	5	4
P3	4	6
P4	0	4

Calculate the sum of average waiting time and average turn around time of all the processes scheduled using First come first serve algorithm . [Upto two decimal places]

Note: Scheduling overhead is negligible.

[NAT]

8. Consider a pre-emptive system with 6 processes, each process executes for 2 ms and leaves the system, and again enters the system(in any order) after all other processes has executed its first 2 ms. Each process has a burst time of 5 ms. What will be the total schedule length. Assuming no process went in block state.

Answer Key

1. **(b)**
2. **(d)**
3. **(a, b, c)**
4. **(32)**
5. **(b, d)**
6. **(7.75)**
7. **(13.25)**
8. **(30)**

Hint & Solutions

1. (b)

Process can terminate or complete its execution from running state only. The termination may be normal or abnormal, but a process will always terminate from running state only.

2. (d)

Given,

n number of processes

s is time when each process terminates = Completion time

t is the time each process arrives to the ready queue = Arrival time

Average time for n processes to complete their execution= Average turnaround time

Average Turnaround time for n processes is given as

$\sum_{i=1}^{n} \frac{(s_i - t_i)}{n}$

Therefore, option D is the correct answer.

3. (a, b, c)

For n processes, the total number of schedules possible in non-pre-emptive system are n! And Total number of schedules possible in pre-emptive system are infinite.

Therefore, option A, B, C all are incorrect statements and correct options.

4. (32)

Gantt Chart:

IDLE	P1	P2	P3
0	4	16	24 32

Schedule length = completion time of last process - arrival time of first process.

X = 32- 4

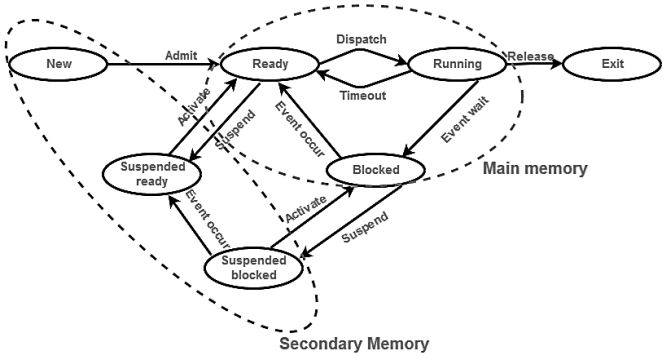
= 28

CPU is idle for 4 units from time 0 to 4. Y = 4

Therefore, X+ Y = 32.

5. (b, d)

According to process transition diagram



A blocked process can move to suspend ready state. And, A running process can move to blocked state.

Therefore, option B, D are correct.

6. (7.75)

P4	P3	P1	P2
0	3	8	12 18

Processes	Arrival Time	Burst Time	Completion Time	Turn Around Time
P1	3	4	12	9
P2	5	6	18	13
P3	2	5	8	6
P4	0	3	3	3

Average Turnaround time = (9 + 13+ 6 + 3)/ 4 = 7.75

7. (13.25)

P4	P1	P3	P2
0	4	9	15 19

Processes	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting time
P1	2	5	9	7	2
P2	5	4	19	14	10
P3	4	6	15	11	5
P4	0	4	4	4	0

Average Waiting time = (2+ 10+ 5+ 0)/ 4 = 4.25

Average Turnaround time = (7+ 14+11+4)/ 4 = 9

Sum of average waiting time and average turnaround time of all the processes = 9 + 4.25 = 13.25

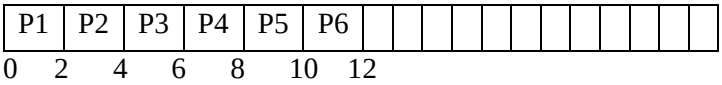
8. (30)

Schedule length is defined as the difference between completion time of last process and arrival time of first process.

There are total 6 processes in the system and it is given that each process will enter the system for 2 ms.

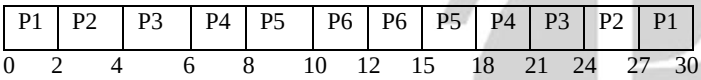
We can assume 6 processes as P1, P2, P3, P4, P5, P6

First all the processes completed their 2 ms.



Now, all the processes has remaining 3ms burst time because the condition is over, and now we can execute processes in any order.

Assuming now, we are executing processes as P6, P5, P4, P3, P2, P1



So, the total schedule length is 30.

Branch : CSE & IT Batch : Hinglish

Operating System

CPU Scheduling

DPP 02

[MCQ]

1. Consider arrival time and execution time for the following processes.

Process	Arrival time	Burst time
P0	2	5
P1	7	9
P2	8	3
P3	10	4

Assume, each context switch requires 1ms of time (excluding first and last context switch), then calculate the sum of schedule length, turnaround time of P3, and waiting time for P1 using SRTF algorithm?

(a) 39 (b) 34

(c) 40 (d) 25

[NAT]

2. Consider arrival time and execution time for the following processes that need to be scheduled on a single CPU.

Process	Arrival time	Burst time
P0	1	4
P1	6	3
P2	7	9
P3	5	5

Assume X is the total number of contexts switching and Y is the throughput of the system using SJF scheduling algorithm then find the value of (X + Y)? (Exclude start and end context switching and round-off up to 2 decimal).

[NAT]

3. Consider the following set of processes with the arrival times and the CPU burst times given in milliseconds:

Process	Arrival time	Burst time
P0	0	4
P1	2	3
P2	3	4
P3	5	5
P4	1	2

What does the average turnaround time for these processes with the shortest remaining process time first (SRTF) algorithm? (If burst time is matching them follow lowest arrival time and round-off up to 1 decimal).

[MCQ]

4. Five jobs are waiting to be run. Their expected running time are 12, 5, 7, 3 and "X". Which of the following order will minimize the average completion time using the shortest job first (SJF)?

(a) 3, X, 5, 7, 12

(b) 3, 5, X, 7, 12

(c) 3, 5, 7, X, 12

(d) 3, 5, 7, 12, X

[NAT]

5. Consider a pre-emptive SJF scheduling technique followed by three processes P_1, P_2, P_3. All these 3 processes arrive at time t = 0 and their total execution time is 30, 40, and 50 units respectively. Each process spends the first 10% of execution doing I/O, 80% of CPU and the rest 10% 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, round off upto 2 decimal).

[MCQ]

6. Consider the following processes:

Process	Arrival time	Burst time
P0	3	2
P1	6	3
P2	12	3
P3	5	2
P4	25	4

What is the throughput using the first come first serve(FCFS) algorithm, where scheduling overhead requires 2 unit (excluding first and last context switch)?

(a) 0.16 (b) 0.17

(c) 0.18 (d) 0.28

[MSQ]

7. Match List I with List II, and select the correct answer using the code below:

List I	**List II**
(i) SJF	1. Premptive
(ii) SRTF	2. Non-Preemptive
(iii) FCFS	3. Starvation

(a) i) - 1, ii)- 2, iii)-3
(b) i)- 3, ii)- 1, iii)- 2
(c) i)- 2, ii)-1, iii) 2
(d) i)-3, ii)-3, iii) 2

Answer Key

1. **(c)**
2. **(3.19)**
3. **(7.6)**
4. **(a)**
5. **(7.69)**
6. **(b)**
7. **(b, c, d)**

Hint & Solutions

1. (c)

Gantt Chart for the above process:

IDLE	P0	CS	P2	CS	P3	CS	P1

0 2 7 8 11 12 16 17 26

Process	Arrival Time	Burst Time	Completion Time	Turn-around Time	Waiting Time
P0	2	5	7	5	0
P1	7	9	26	19	10
P2	8	3	11	3	0
P3	10	4	16	6	2

Schedule length = 26 – 2 = 24
Turnaround time of P3 = 6
Waiting time of P1 = 10
So, 24 + 6 + 10 = 40
Therefore, option C is correct.

2. (3.19)

Gantt Chart for the above process:

IDLE	P0	P3	P1	P2

0 1 5 10 13 22

Total number of context switches:
Throughput = Number of processes/ Schedule length
= 4/ (22 – 1)
= 4/21
= 0.190
So, X + Y = 3 + 0.190 = 3.19

3. (7.6)

Gantt Chart for the above process:

Ready queue: P0 P4 P0 P1 P2 P3

P0	P4	P0	P1	P2	P3

0 1 3 6 9 13 18

Process	Arrival time	Burst time	Completion time	Turnaround Time
P0	0	4	6	6
P1	2	3	9	7
P2	3	4	13	10
P3	5	5	18	13
P4	1	2	3	2

Average TAT = (6 + 7 + 10 + 13 + 2)/ 5 = 7.6

4. (a)

For shortest job first (SJF), to get the minimum average completion time the value of x should be as minimum as possible.
In the option a x is placed in the second position so, the value of x must be ≤ 3. Now if we put the x value 3 then the minimum average completion time will be:
(3 + 6 + 11 + 18 + 30)/5 = 13.6

5. (7.69)

Scheduling algorithm is preemptive SJF = SRTF

Process	Burst Time	I/O	CPU	I/O
P1	30	3	24	3
P2	40	4	32	4
P3	50	5	40	5

IDLE	P1	P2	P3	IDLE

0 3 27 59 99 104

Hence,
CPU idle time = (3 – 0) + (104 – 99) = 3 + 5 = 8
Total time taken to complete all processes = 104
CPU idle percentage = (8 / 104) * 100 = 7.69%

6. (b)

IDLE	P0	CS	P3	CS	P1	CS	P2	CS	IDLE	CS	P4

0 3 5 7 9 11 14 16 19 21 25 27 31

Throughput = Total processes/ Schedule length
= 5 / (31 – 3)
= 5/28
= 0.17
Therefore, B is the correct answer.

7. (b, c, d)

SJF is non- preemptive in nature and has starvation.
SRTF is preemptive in nature and has starvation.
FCFS is non-preemptive in nature and do not have starvation.
Therefore, option b, c, and d are correct.

Branch : CSE & IT Batch : Hinglish

Operating System

CPU Scheduling

DPP 03

[MCQ]

1. Consider the following statements:

(i) SJF is the most optimal algorithm among all other scheduling algorithms.

(ii) SRTF is practically non-implementable because it is pre-emptive in nature.

(iii) SJF is practically implementable because of it's non-premptive nature.

Which of the following is CORRECT?

(a) Only (i) and (ii).
(b) Only (ii) and (iii).
(c) All (i), (ii), and (iii).
(d) None of these.

[MCQ]

2. Consider the set of processes.

Process	Arrival time	Burst time
A	2	3
B	4	17
C	2	4
D	1	2
E	7	5

Assume that the processes are scheduled using the non-pre-emptive Shortest Job First(SJF) algorithm. What are the average turnaround time and average waiting time?

(a) 13.5, 5.25 (b) 10.0, 3.8
(c) 12.2, 5.6 (d) 8.75, 4.25

[MCQ]

3. A Scheduler that selects partially executed processes from a secondary storage device is called_____.

(a) Medium-term scheduler.
(b) Short term scheduler.
(c) Long Term scheduler.
(d) None of these.

[NAT]

4. Consider arrival time and execution time for the following processes that need to be scheduled on a single CPU.

Process	Arrival time	Burst time
P0	0	3
P1	2	5
P2	3	4
P3	4	2
P4	6	1
P5	6	2

What is the sum of throughput of the system and average waiting time if SRTF Scheduling is used? (upto 2 decimal)

[MCQ]

5. Consider the given process table:

Process	Arrival time	Burst time
P0	6	8
P1	0	6
P2	2	4
P3	3	6
P4	4	1
P5	5	2

Consider the context switching overhead of 1 unit. What is the percentage of CPU overhead activity while using SRTF scheduling for the processes given in the above table? Include the context switching to load the first process, and save the last process.

(a) 20.5% (b) 23.5%
(c) 25% (d) 27.6%

[MSQ]

6. Which of the following statement(s) is/are correct in the context of CPU scheduling?

(a) Turnaround time includes waiting time.
(b) SRTF can be used even when the CPU time required by each of the processes is not known apriori.
(c) Implementing preemptive scheduling may need hardware support.
(d) The goal is to only maximize CPU utilization and minimize throughput.

[NAT]

7. Consider the following five processes, with their arrival times and execution times given in milliseconds.

Process	Arrival time	Burst time
P0	1	3
P1	0	2
P2	4	4
P3	3	6
P4	6	2

Using Non - pre-emptive Shortest Job First, what will be the average waiting time? [upto 1 decimal point]

[MCQ]

8. Which of the following scheduler can control degree of multiprogramming?

(a) Medium-term scheduler.
(b) Short term scheduler.
(c) Long Term scheduler.
(d) None of these.

Answer Key

1. **(d)**
2. **(b)**
3. **(a)**
4. **(3.01)**
5. **(c)**
6. **(a, c)**
7. **(2.6)**
8. **(c)**

Hint & Solutions

1. (d)

(i) SJF is the most optimal algorithm among all other scheduling algorithms. CORRECT.

(ii) SRTF is practically non-implementable because it is pre-emptive in nature. SRTF is non-implementable because predicting burst time of a process before it's execution is not possible. INCORRECT

(iii) SJF is practically implementable because of it's non-premptive nature. INCORRET. SJF is practically non-implementable because predicting burst time of a process before it's execution is not possible.

Hence, option D is correct.

2. (b)

Gantt Chart:

IDLE	D	A	C	E	B

0 1 3 6 10 15 32

Process	Arrival time	Burst time	Completion time	Turnaround Time	Waiting Time
A	2	3	6	4	1
B	4	17	32	28	11
C	2	4	10	8	4
D	1	2	3	2	0
E	7	5	15	8	3

Average Waiting time = (1 + 11+ 4 + 0 + 3)/ 5 = 3.8
Average Turnaround time = (4 + 28 + 8 + 2 + 8)/ 5 = 10

3. (a)

The function performed by the medium-term scheduler is called "Swapping".

- When the degree of multi-programming increases up to a certain point then CPU performance start decreasing due to page fault and context switching.
- Now, the medium-term scheduler starts swapping processes from "Ready state" and "waiting/block state" to "suspend Ready state" and "Suspend wait/block State" respectively to decrease the degree of multi-programming.
- "Suspend Ready State" and "Suspend wait/block State" are in secondary memory So, the Medium Term Scheduler selects the partially executed process from the secondary storage device and bring it back for execution.

4. (3.01)

Gantt Chart:

P0	P2	P3	P4	P5	P2	P1

0 3 4 6 7 9 12 17

Process	Arrival time	Burst time	Completion time	Turnaround Time	Waiting Time
P0	0	3	3	3	0
P1	2	5	17	15	10
P2	3	4	12	9	5
P3	4	2	6	2	0
P4	6	1	7	1	0
P5	6	2	9	3	1

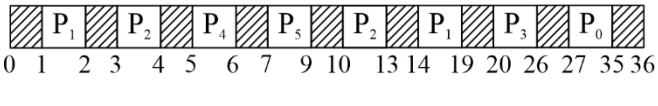
Average waiting time = (0 + 10 + 5 + 0 + 0 + 1)/ 6
= 2.66

Throughput of the system = Number of processes/ Schedule Length.
= 6/(17 – 0)
= 6/ 17
= 0.35

So, Sum of Throughput and average waiting time
= 2.66 + 0.35 = 3.01

5. (c)

Gantt Chart:



Total Context Switches = 9

Schedule Length = 36

% of CPU overhead activity = 9/36 = 0.25 ≈ 25%

6. (a, c)

(a) Turnaround time includes waiting time. Correct; Turnaround time = CPU Burst time + IO burst time + **waiting time**

(b) SRTF can be used even when the CPU time required by each of the processes is not known apriori. Incorrect. SRTF cannot be implemented if CPU time required by each process is unknown, without knowing CPU burst time it is not possible to schedule process with shortest remaining time.

(c) Implementing preemptive scheduling needs hardware support. Correct. Preemptive scheduling may require timer for scheduling processes.

(d) The goal is to only maximize CPU utilization and minimize throughput. Incorrect. The goal of CPU scheduling is to maximize CPU utilization and maximize throughput as well.

7. (2.6)

Gantt Chart:

P1	P0	P2	P4	P3

0 2 5 9 11 17

Process	Arrival time	Burst time	Completion time	Turnaround Time	Waiting Time
P0	1	3	5	4	1
P1	0	2	2	2	0
P2	4	4	9	5	1
P3	3	6	17	14	8
P4	6	2	11	5	3

Average waiting time = (1+ 0 + 1+ 8 + 3)/ 5 = 2.6

8. (c)

Long term scheduler or Job scheduler selects the processes from the pool (or the secondary memory) and then maintains them in the primary memory's ready queue. The multiprogramming degree is mostly controlled by the long-term scheduler.

Branch : CSE & IT Batch : Hinglish

Operating System

CPU Scheduling

DPP 04

[MSQ]

1. Which of the following statements is/are correct regarding HRRN scheduling algorithm?

(a) It reduces waiting time for longer processes.

(b) HRRN is preemptive in nature.

(c) HRRN's response ratio can be calculated as (S + W)/S, where S is the service time and W is the waiting time.

(d) HRRN favours job with shorter burst time.

[MSQ]

2. Consider the below table

Process	Arrival time	Burst time
P1	0	7
P2	2	4
P3	4	1
P4	5	4

Schedule the above processes using SRTF and SJF, and select the correct option from the following.

(a) TAT(SJF) > TAT(SRTF).

(b) Waiting time P4(SJF) > Waiting time P3 (SRTF)

(c) TAT(SRTF)> TAT(SJF)

(d) Waiting time P2(SRTF)< Waiting time P1(SJF)

[NAT]

3. How many of the following scheduling algorithms may cause starvation?

i. First-come-first-serve

ii. Priority

iii. Shortest process next

iv. Shortest remaining time first

[NAT]

4. Consider the following processes:

Process	Arrival time	Burst time
P1	0	5
P2	1	5
P3	2	5
P4	3	5

Calculate the sum of waiting time of process P4 and average turnaround time using LRTF scheduling algorithm? (if burst time of two process is same schedule the process with higher arrival time)

[NAT]

5. Consider the set of 5 processes whose arrival time and burst time are given below:

Process	Arrival time	Burst time
P0	0	4
P1	3	5
P2	4	6
P3	5	5
P4	8	3

If the CPU scheduling policy is Highest Response Ratio Next, X is response ratio of last process scheduled, Y is the average turnaround time, and Z is the average waiting time. Calculate X * Y + Z? (upto 1 decimal point)

[MCQ]

6. Match the following groups.

Group-I	Group-II
A. FCFS	1. Used in time sharing system
B. SJF	2. Used in real time system
C. Priority scheduling	3. Minimum average waiting time
	4. Convoy effect

(a) A-2, B-1, C-3

(b) A-4, B-3, C-2

(c) A-2, B-3, C-1

(d) A-4, B-2, C-3

[MCQ]

7. Consider the three processes for a particular system P1, P2, P3. The arrival time of the processes are 0ms, 5ms and 8ms respectively. And the processing times are 15ms, 12ms and 5ms respectively. The three processes

are pre-emptively scheduled on a single-CPU system using the shortest remaining processing time first scheduling policy. Which of the following shows the order in which processes are completed?

(a) P2, P1, P3 (b) P3, P1, P2

(c) P3, P2, P1 (d) P1, P2, P3

[NAT]

8. Consider the following set of processes with the arrival times and burst times. Processes are scheduled using highest response ratio next.

Process	Arrival Time	Burst Time
A	0	4
B	2	1
C	4	3
D	5	5

(All time in milliseconds)

The average waiting time of processes are _____ (ms). (Upto 1 decimal place)

Answer Key

1. **(a, c, d)**
2. **(a, b)**
3. **(3)**
4. **(26)**
5. **(34.6)**
6. **(b)**
7. **(b)**
8. **(1.5)**

Hint & Solutions

1. (a, c, d)

(a) HRRN reduces waiting time for longer processes. Correct

(b) HRRN is preemptive in nature. Incorrect- HRRN is non-preemptive in nature.

(c) HRRN's response ratio can be calculated as (S + W)/S, where S is the service time and W is the waiting time. Correct

(d) HRRN favors job with shorter burst time. Correct

Therefore, A, C, D are correct statements.

2. (a, b)

SRTF:

P1	P2	P3	P2	P4	P1

0 2 4 5 7 11 16

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	7	16	16	9
P2	2	4	7	5	1
P3	4	1	5	1	0
P4	5	4	11	6	2

Average Turn Around Time = (16 + 5 + 1 + 6)/ 4

= 28/4 = 7

SJF:

P1	P3	P2	P4

0 7 8 12 16

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	7	7	7	0
P2	2	4	12	10	6
P3	4	1	8	4	3
P4	5	4	16	11	7

Average Turn Around Time = (7 + 10 + 4 + 11)/ 4

= 32/4 = 8

Therefore, only option a, b are correct.

3. (3)

(i) First-come-first-served- No starvation

(ii) Priority- Starvation

(iii) Shortest process next- Starvation

(iv) Shortest remaining time first- Starvation

4. (26)

Gantt Chart:

P1	P2	P3	P4	P3	P2	P1	P4	P3

0 1 2 3 5 6 7 8 9 10

P2	P1	P4	P3	P2	P1	P4	P3	P2	P1

10 11 12 13 14 15 16 17 18 19 20

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	5	20	20	15
P2	1	5	19	18	13
P3	2	5	18	16	11
P4	3	5	17	14	9

Waiting time of P4 = 9

Average TAT = (20 + 18 + 16 + 14)/ 4 = 17

Waiting time of P4 + Average TAT = 26

5. (49.2)

Gantt Chart:

P0

0 4

At time 4, process P1 and P2 arrived, calculating response ratio for P1 and P2.

P1 = (W + S)/ S = (1 + 5)/5 = 6/5 = 1.2

P2 = (W + S)/ S = (0 + 6)/6 = 6/6 = 1

So, P1 will be scheduled next.

P0	P1

0 4 9

At time 9, process P3 and P4 arrived and P2 was already waiting, calculating response ratio for P2, P3, and P4.

P2 = (W + S)/ S = (5 + 6)/6 = 11/6 = 1.8

P3 = (W + S)/ S = (4 + 5)/5 = 9/5 = 1.8

P4 = (W + S)/ S = (1 + 3)/3 = 4/3 = 1.3

So, P2 will be scheduled next.

P0	P1	P2

0 4 9 15

At time 15, calculating response ratio for P3, and P4.

P3 = (W + S)/ S = (10 + 5)/5 = 15/5 = 3

P4 = (W + S)/ S = (7 + 3)/3 = 10/3 = 3.3

Now, P4 will be scheduled and then P3 will be scheduled

P0	P1	P2	P4	P3

0 4 9 15 18 23

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P0	0	4	4	4	0
P1	3	5	9	6	1
P2	4	6	15	11	5
P3	5	5	23	18	13
P4	8	3	18	10	7

Response ratio of last process scheduled, last process scheduled was P3, its response ratio was 3, so X = 3.

Average waiting time = (0 + 1 + 5 + 13 + 7)/ 5

= 26/5 = 5.2

Average Turnaround time = (4 + 6 + 11 + 18 + 10)/5

= 49/5 = 9.8

So, X * Y + Z = 34.6

6. (b)

FCFS: Suffers from Convoy effect.

SJF: Minimum average waiting time

Priority scheduling: Used in real time system.

7. (b)

Gantt Chart:

P1	P3	P1	P2

0 8 13 20 32

Process completion order: P3, P1, P2. So, option b is correct answer.

8. (1.5)

Gantt Chart:

A

0 4

At time 4, process B and C are in the system, calculating response ratio

B = (2 + 1)/1 = 3

C = (0 + 3)/ 3 = 1

So, B will be scheduled next

A	B

0 4 5

At time 5, process C and D are in the system, calculating response ratio

C = (1 + 3)/3 = 1.333

D = (0 + 5)/ 5 = 1

So, C will be scheduled next, and after that D will be scheduled.

A	B	C	D

0 4 5 8 13

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
A	0	4	4	4	0
B	2	1	5	3	2
C	4	3	8	4	1
D	5	5	13	8	3

Average waiting time = (0 + 2 + 1 + 3)/ 4 = 1.5

Branch : CSE & IT Batch : Hinglish

Operating System

CPU Scheduling

DPP 05

[NAT]

1. Consider the following table

Process	P1	P2	P3	P4
Arrival Time	0	1	2	3
Service Time	6	3	1	4

Find the average turn around time when processes are scheduled with SRTF. (Upto 2 decimal point)

[MCQ]

2. Consider the following statements:

(i) If all jobs arrive at the same point in time, a SJF and an SRTF scheduler will behave the same.

(ii) If all jobs arrive at the same point in time and have identical run lengths, a FIFO and a SJF scheduler will behave the same.

Which of the following statements is true?

(a) Only (i) is true.
(b) Only (ii) is true.
(c) Both (i) and (ii) are true.
(d) Both (i) and (ii) are false.

[MCQ]

3. Consider the following process table with Arrival and Burst time

Process	Arrival Time	Burst Time
P_1	1	2
P_2	6	4
P_3	4	10
P_4	5	6

Average waiting time of these processes by using round robin scheduling, where time quantum is 2 unit.

(a) 2.25 (b) 6.25
(c) 4.25 (d) 8.25

[MCQ]

4. Which of the following scheduling algorithm is better than others in forms of response time.

(a) SJF (Shortest job first)
(b) SRTF (Shortest remaining time first)
(c) FCFS (First come first serve)
(d) RR (Round Robin)

[MCQ]

5. Consider the 3 processes, P1, P2 and P3 shown in the table:

Process	Arrival Time	Burst Time
P1	0	5
P2	4	7
P3	6	4

The completion order of the 3 processes under the policies FCFS and RR2 (round robin scheduling with CPU quantum of 2-time units) are

(a) FCFS: P1, P2, P3; RR2: P1, P2, P3
(b) FCFS: P1, P3, P2; RR2: P1, P3, P2
(c) FCFS: P1, P2, P3; RR2: P1, P3, P2
(d) FCFS: P1, P3, P2; RR2: P1, P2, P3

[MSQ]

6. Suppose a system has 5 processes A, B, C, D, E all processes are arrived at time 0 and have burst time 4, 5, 7, 2, 5 respectively. If the time slice for each process is 2 unit and each context switch requires one unit of time (excluding first and last context switch), then which of the following statement is correct?

(a) At time 25, Process B is running on the system.
(b) Process E finishes execution at time 31.
(c) Total CPU overhead activity is 33%.
(d) Process B finishes its execution before process E

[MCQ]

7. In the Round Robin scheduling policy if the time quantum used is more than the maximum time required to execute any process. Then the policy will ____.

(a) Behave same as FCFS.
(b) Behave same as SJF.
(c) Behave same as HRRN.
(d) None of these.

[MCQ]

8. Which of the following scheduling algorithm treat every process equally?

(a) Shortest Job First
(b) Shortest remaining Time first (SRTF)
(c) Longest remaining Time first (LRTF)
(d) Round Robin

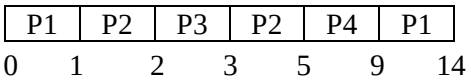
Answer Key

1. **(6.25)**
2. **(c)**
3. **(b)**
4. **(d)**
5. **(c)**
6. **(d)**
7. **(a)**
8. **(d)**

Hint & Solutions

1. (6.25)

Gantt Chart:



Average Turn Around time:

TAT = Completion Time – Arrival Time

TAT(P1) = 14 – 0 = 14

TAT(P2) = 5 – 1 = 4

TAT(P3) = 3 – 2 = 1

TAT(P4) = 9 – 3 = 6

Average Turn Around time: (14 + 4 + 1 + 6)/4 = 6.25

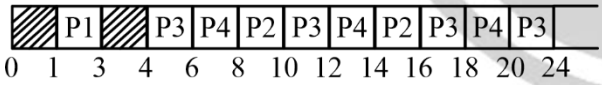
2. (c)

Since all jobs arrive at the same point in time, a SJF and an SRTF scheduler run all process for same amount of time.

All are of identical run lengths FIFO and a SJF scheduler will behave the same, Since no shortest job concept will be there.

3. (b)

Gantt Chart



Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P_1	1	2	3	2	0
P_2	6	4	16	10	6
P_3	4	10	24	20	10
P_4	5	6	20	15	9

Average waiting time = $\frac{0+6+10+9}{4}$

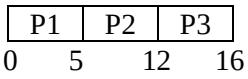
$= \frac{25}{4} = 6.25$

4. (d)

Round Robin scheduling algorithm is best in terms of response time. Due to fixed time quantum each process has equal chance of visiting CPU.

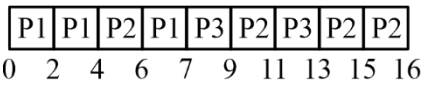
5. (c)

FCFS:



Completion order: P1, P2, P3

Round Robin:



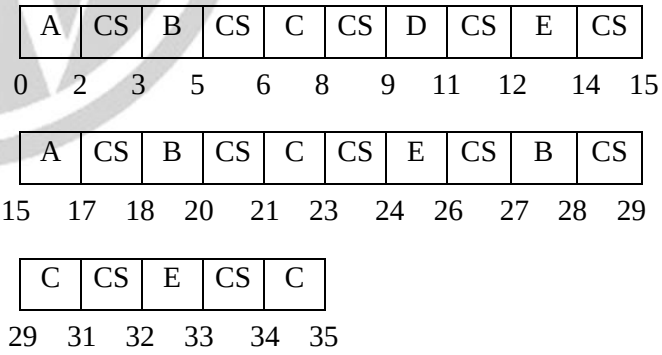
Completion order for RR: P1, P3, P2

Therefore, option C is correct answer.

6. (d)

Time quantum = 2

Process	Arrival Time	Burst Time
A	0	4
B	0	5
C	0	7
D	0	2
E	0	5



(a) At time 25, Process B is running on the system. Incorrect. At time 25 process E is running on CPU.

(b) Process E finishes execution at time 31. Incorrect. Process E finishes its execution at time 33.

(c) Total CPU overhead activity is 33%. Incorrect. CPU overhead activity is = 12/ 35 = 34.2%

(d) Process B finishes its execution before process E. Correct. Process B finishes its execution at time 28 whereas process E finishes its execution at time 33.

7. (a)

If the time quantum used in the Round Robin scheduling algorithm is more than the maximum burst time then it will degenerate to a first come first serve (FCFS) algorithm. So, option a is correct.

8. (d)

In round robin process scheduling, all process are given an equal amount of time for execution despite of their longer/ shorter/ remaining burst time. In round robin scheduling policy, every process is treated equally and given equal time to execute on CPU. In SJF and SRTF process with shorter burst time has priority over process with longer burst time, and in LRTF process with longer burst time has priority over process with shorter burst time.

Therefore, option d is the correct answer.

Branch : CSE & IT Batch : Hinglish

Operating Systems

Process Synchronization/ Coordination

DPP 01

[MCQ]

1. Consider the following statements:

S_1: If all jobs have identical run lengths, a RR scheduler provides better average turnaround time than FIFO.

S_2: With a MLFQ scheduler, high priority jobs have longer time-slices than low priority jobs.

Which of the following is true?

(a) Only S_1
(b) Only S_2
(c) Both S_1 and S_2
(d) None of these

[MCQ]

2. Suppose there is a system operating upon round-robin scheduling, If e denotes time that is been needed to do a process switch and if w denotes round-robin time quantum and if n denotes average time that a process is required to run. Then what will be the CPU efficiency under following circumstances. If round-robin time quantum is greater than the average time i.e. (w > n).

(a) n
(b) 1/(n + e)
(c) n/(n + e)
(d) None of these

[NAT]

3. A process spends 20% of its execution time waiting for completion of I/O operation. If there are 8 processes in memory at once, then the probability of CPU time utilized is _____ %. (Assume all I/O operations are overlapped). (Upto 2 decimal places)

[NAT]

4. Considering the exponential average behavior used to predict the next CPU burst. If $\alpha = 0.70$ and $\tau_0 = 30$ ms and previous (T_0 , T_1, T_2, T_3) runs were as 10, 12, 15, 20. The predicted value of τ_4 _____

[MSQ]

5. What are the limitations of Single- Ready queue?

(a) All processes have to use same scheduling technique.
(b) Lots of searching time required to select single process.
(c) Multiple processes cannot run simultaneously.
(d) In a single-ready queue only single process can reside.

[MSQ]

6. Which of the following is/are correct regarding MLQ(Multi-level queue) scheduling?

(a) Some processes may suffer from starvation.
(b) Processes are divided into categories and scheduled on different ready queue.
(c) Multiple scheduling algorithms can be implemented simultaneously.
(d) It has minimal scheduling overhead.

[MCQ]

7. Which process can be affected by other processes executing in the system?

(a) cooperating process
(b) child process
(c) parent process
(d) independent process

[MSQ]

8. A race condition _____

(a) Occurs when two threads enter critical section at the same time.
(b) Occurs when one thread is in critical section and another thread cannot access critical section.
(c) Occurs when two threads access a shared variable at the same time.
(d) None of these

Answer Key

1. **(d)**
2. **(c)**
3. **(1.00)**
4. **(19)**
5. **(a, b)**
6. **(a, b, c, d)**
7. **(a)**
8. **(c)**

Hints & Solutions

1. (d)

S_1 False: If all jobs are identical, RR is horrible for turnaround time because all jobs will complete at nearly the same time.

S_2 False: Since jobs that do a lot of computation (long CPU burst) are given low priority.

2. (c)

We know that,
Efficiency = Useful Time/ Total Time
Useful time in this problem= time taken by a process to run (**n**)
Total Time= time taken by a process to run + time that is been needed to do a process switch (e)

= **n+e**
So,
CPU efficiency = n/ (n+e)
Therefore, option C is the correct answer.

3. (1.00)

n = total number of processes

p = fraction of time a process is waiting for I/O

Probability (all processes waiting for I/O) = p^n

CPU utilization = $1 – p^n$
So, in this p = 20% = 0.2; n = 8
CPU time utilization= $(1- (0.2)^8) = 1.00000256$

4. (19)

Exponential average (Aging) , $\tau_{n+1} = \alpha\, t_n + (1 - \alpha)\tau_n$
T_n is predicted burst time of n^{th} process.
t_n = actual burst time of n^{th} process.
α = is smoothing factor and $0 <= \alpha <= 1$
Given,
$t_0 = 30$
$t_1=10$
$t_2=12$
$t_3=15$
$t_4=20$
$\alpha= 0.7$
So,

$\tau_1 = 0.7(10) + (0.3)\,(30) = 16$

$\tau_2 = 0.7(12) + (0.3)\,(16) = 13.2$

$\tau_3 = 0.7(15) + (0.3)\,(13.2) = 14.46$

$\tau_4 = 0.7(20) + (0.3)\,(14.46) = 18.338 = 19.0$(approx)

5. (a, b)

(a) All processes have to use same scheduling technique. True, in single ready queue all the processes have to use only one ready queue.

(b) Lots of searching time required to select single process. True, if ready queue has n processes, then we need to search n processes to select a single process for scheduling. If there are 1000 processes in ready queue and we need to execute an interactive process, so we need to first search for the category of interactive processes and then select a single process from that category, this require lot of searching time.

(c) Multiple processes cannot run simultaneously. False, On a single CPU, single process can run/execute at a time, so this is not the limitation of ready queue.

(d) In a single-ready queue only single process can reside. False, on a single ready queue, total processes present can be equal to size of ready queue. If ready queue is full than processes are sent to suspend ready state.

Therefore, option A and B are correct.

6. (a, b, c, d)

(a) Some processes may suffer from starvation. Correct, MLQ has starvation. So, this is correct statement.

(b) Processes are divided into categories and scheduled on different ready queue. Correct, different processes are on different ready queues, like there will be queue for system processes, another queue for interactive processes, etc.

(c) Multiple scheduling algorithms can be implemented simultaneously. Correct, On multiple queue's multiple scheduling algorithms can be implemented simultaneously.

(d) It has minimal scheduling overhead. Correct, As MLQ assigns permanent queues to the processes therefore, it has the advantage of low scheduling overhead.

7. (a)

A cooperating process can be affected by other processes executing in the system. Also it can affect other processes executing in the system. A process shares data with other processes, such a process is known as a cooperating process.

8. (c)

A race condition **occurs when two threads access a shared variable at the same time**. Therefore, option c is correct.

Branch : CSE & IT Batch : Hinglish

Operating System

Process Synchronization/Coordination

DPP 02

[MCQ]

1. Intra Process–communication is_____.

(a) Two processes within same system sharing resources.

(b) Two entities within same process communicating with each other.

(c) Two variables of two processes communicating with each other.

(d) None of these.

[MSQ]

2. In IPC, synchronization is required to eliminate_____.

(a) Inconsistency

(b) Deadlock

(c) Progress

(d) Data-loss

[MCQ]

3. Consider the following statements:

(i) User mode execution is non-atomic.

(ii) User process can be preempted after completion of any instruction.

Which of the following correct?

(a) Only (i) is correct

(b) Only (ii) is correct

(c) Both (i) and (ii) are correct

(d) None of these

[MSQ]

4. Preemption during execution can lead to_____.

(a) Inconsistency

(b) Correct result

(c) Data loss

(d) Incorrect result

[MCQ]

5. Consider the snippet following two processes:

P_1

```
{
    int x;
    int y;
    y = x + 1;
    x + 1;
}
```

P_2

```
{
    int p;
    int y;
    y = p – 1;
    p – 1;
}
```

What is the shared variable in both processes?

(a) x (b) y

(c) p (d) all of these

[MCQ]

6. Critical section is_____.

(a) Part of the program which does not access shared resource.

(b) Complex part of program which cannot be translated by complier.

(c) Such section will always cause deadlock.

(d) Part of the program where shared resources are accessed.

[NAT]

7. Consider the following code of producer consumer problem:

```
# define N 1000
int Buffer [N]
int count = 0
void producer (void)
{
    int itemp, in = 0;
    while (1)
    {
        itemp = Produce_item();
        while (count = = N);
        Buffer[in] = itemp;
        in= (in + 1) % N;
        count = count +1;
    }
```

```
void consumer (void)
{
int itemc, out = 0;
while (1)
{
    while (count = =0);
    itemc = Buffer [out];
    out = (out + 1) % N;
    count  = count – 1;
    process_item(itemc);
}
```

}

How many variables from the above code belong to critical section?

[MSQ]

8. Necessary condition for synchronization problems to occur in Inter–process communication environment are

(a) Critical section

(b) Non-critical section

(c) Race condition

(d) Preemption

Answer Key

1. **(b)**
2. **(a, b, d)**
3. **(c)**
4. **(a, c, d)**
5. **(b)**
6. **(d)**
7. **(2)**
8. **(a, c, d)**

Hint & Solutions

1. (b)
Intra-process communication is when two entities (function) within same process communicating with each other using parameter passing technique or global variable.

2. (a, b, d)
Inconsistency (incorrectness), data-loss, and deadlock can occur in the absence of synchronization.

3. (c)
User mode is non-atomic or preemptive and in user mode processes can get pre-empted after completion of any instruction.
Therefore, option 'c' correct.

4. (a, c, d)
Preemption during the execution of process can lead to inconsistency, data loss and incorrect result.

5. (b)
'y' is the shared variable between P_1 and P_2 as both P_1 and P_2 can modify value of 'y'.

6. (d)
By definition, critical section is a part of the program where shared resources are accessed.

7. (2)
Count and Buffer are two shared variable and belong to critical section.

8. (a, c & d)
Critical section, Race– condition and Preemption (preemptive processes) are necessary condition for occurrence of synchronization problems

Branch : CSE & IT Batch : Hinglish

Operating System

Process Synchronization/Coordination

DPP 03

[MSQ]

1. Producer Consumer problem has_____.

(a) Competition (b) Co-operation
(c) Inconsistency (d) Data loss

[MCQ]

2. Purpose of "entry section" in synchronization mechanism is.

(a) To allow all processes to access shared resource concurrently.
(b) To allow one process at a time to enter into critical section.
(c) To indicate the program is started
(d) None of these

[MSQ]

3. What are the requirements of critical section problem?

(a) Mutual exclusion (b) Bounded waiting
(c) Deadlock (d) Consistency

[MCQ]

4. Mutual exclusion is_____.

(a) Any number of process can enter into critical section if it is free.
(b) No process present in Non-CS should block the other process from entering into CS.
(c) No two process may be simultaneously present in CS.
(d) None of these.

[MCQ]

5. Consider two processes.

Process X	Process Y
while(true){	while (true){
x = l;	y = 1;
while(y= =l){	while (x = =1){
// critical section	// critical section
x = 0;	x = 0;
}	}
}	}

Assume x and y shared variables and initialized to 0.

Which of the given condition are satisfied by the above code?

(a) Mutual exclusion and progress
(b) Mutual exclusion
(c) Progress
(d) No mutual exclusion

[MCQ]

6. If Bounded waiting condition is not satisfied, it can cause.

(a) Inconsistency (b) Data-loss
(c) Starvation (d) Deadlock

[MSQ]

7. Which of the following is/are incorrect?

(a) Process cannot be pre-empted from CS.
(b) Process can enter CS without going into entry section.
(c) Process can complete/leave CS without going into exit section.
(d) Process can be preempted from entry, exit and critical section.

[NAT]

8. How many of the following are software type solution for synchronization?

(i) Lock variable (ii) Monitor
(iii) Peterson's solution (iv) Semaphore
(v) Test and set Lock instruction set. (vi)Strict algorithm
(vii) Swap

[NAT]

9. Consider the following program segments for two different processes (P_1 & P_2) executing concurrently; 'a' & 'b' are not shared variables, 'x' is shared and starts at '0'.

P_1	P_2
for (a = 1; a< =4; ++a)	for (b =1; b< =4;++b)
x = x +1;	x = x + 1;

If P_1 and P_2 execute only once and concurrently, then the final minimum possible value of x is_____.

Answer Key

1. **(a, b, c & d)**
2. **(b)**
3. **(a, b)**
4. **(c)**
5. **(d)**
6. **(c)**
7. **(a, b & c)**
8. **(2)**
9. **(4)**

Hint & Solutions

1. (a, b, c & d)

Both co-operation and competition is present in producer consumer problem. Producer is incrementing the value of count variable and consumer is decrementing the value. If they both execute at same time then both processes (producer & consumer) are competing for shared variable "count".

Therefore, this leads to inconsistent result and data loss. Option, a, b, c & d all are correct.

2. (b)

Main purpose of "entry section" in synchronization mechanism is to allow only one process from N processes to enter into critical section at a time.

3. (a, b)

The critical section problem needs a solution to synchronize the different processes. The solution to the critical section problem must satisfy the following conditions:

(1) Mutual exclusion
(2) Progress
(3) Bounded waiting

Therefore, a, b are correct option.

4. (c)

Mutual exclusion implies that only one process can be inside the critical section at any time. If any other process require the CS, it must wait until CS is free. No two processes may be simultaneously present in CS.

5. (d)

Here, both processes can enter into critical section simultaneously, so no mutual exclusion.

Therefore, option (d) is correct.

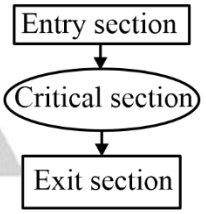
6. (c)

If Bounded wait condition is not satisfied, it means one process is getting chance to enter CS again and again, and other processes are starving for CS, so this can lead to starvation for other processes.

7. (a, b & c)

Process can be pre-empted from critical section, entry section and exit section. So, option 'a' is incorrect and option 'd' is correct.

Process cannot enter into critical section without going to entry section.



Process is said to have leave/left "CS" only if it completes exit section.

Therefore, option (b) and (c) are incorrect.

8. (2)

Lock variable and Peterson algorithm are software type solution. Strict alternation is also software type solution.

Monitor and semaphore are OS based solution.

TSL instruction set hardware type solution.

Strict algorithm and swap are no solution.

9. (4)

If P_1 gets pre-empted during its first iteration and then completing 4 iterations of P_2 and completing first iteration of P_1, followed by it's remaining iterations.

Then, we would have executed only 4 iteration of P_1 and thus, the final minimum possible value of x is 4.

Branch : CSE/IT Batch : Hinglish

Operating System

Process synchronization / Coordination

DPP-04

[MCQ]

1. Consider the following program:

```
Bool Lock = False;
void Process (int i)
{
    while(1)
    {
    < Non-critical section >
    while(TSL(&Lock)= = True)
    < Critical Section >
    Lock = False;
    }
}
```

Which of the following are correct regarding given code?

(a) It guarantees mutual exclusion but not progress.

(b) It gurantees mutual exclusion and progress but not bounded waiting.

(c) It guarantees progress but not mutual exclusion.

(d) It guarantees progress and bounded waiting but not guarantees mutual exclusion.

[MSQ]

2. Which of the following is/are correct regarding "SWAP"?

(a) It is a software-based synchronization mechanism.

(b) It is atomic instruction.

(c) It is based on "Lock Key" mechanism.

(d) It is non-privileged instruction.

[MCQ]

3. Match the following:

1)	TSL	(i)	Mutual exclusion
2)	SWAP	(ii)	Progress
		(iii)	Bounded waiting

Which of the following option is correct?

(a) 1 – (i); 2 – (ii); 2 – (iii)

(b) 1 – (ii); 1 – (iii); 2 – (ii)

(c) 1 – (i); 2 – (i); 2 – (ii)

(d) 1 – (iii); 2 – (i); 1 – (i)

[MSQ]

4. Which of the following is/are correct?

(a) TSL has busy waiting and wastes CPU cycle.

(b) TSL suffers from priority - inversion.

(c) SWAP has busy waiting and utilizes CPU cycle efficiently.

(d) SWAP suffer from priority-inversion.

[MCQ]

5. Priority inversion problem can be solved by _____.

(a) Preemption.

(b) Priority Inheritance

(c) Preemption followed by priority exchange.

(d) No solution, and priority inversion problem can cause deadlock.

[MCQ]

6. Consider the following program of producer from producer-consumer problem implemented using sleep() and wakeup ().

```
void Producer(void)
```

```
{
    int itemp, in = 0;
    while(1)
    {
     itemp = Produce.item();
    X;
    Buffer [in] = itemp;
    in = (in + 1) % N;
    Count = Count + 1;
    Y
    }
}
```

What will be the value of X and Y respectively?

(a) X = if(count = = 1) sleep();
    T = if(count = = N) wakeup(consumer);

(b) X = if(count = = N) sleep();
    Y = if(count = = N -1) wakeup(consumer);

(c) X = if(count = = N) sleep();
    Y = if(count = = 1) wakeup(consumer);

(d) X = if(count = = N) wakeup(consumer);
    Y = if(count = = 1) sleep();

[MCQ]

7. Operations performed on semaphores are _____.

(a) Wait and signal
(b) Sleep and wakeup
(c) Increment and decrement
(d) All of the above.

[MSQ]

8. Which of the following are the types of semaphore?

(a) Binary
(b) Bounded
(c) Counting
(d) Incremental

Answer Key

1. **(b)**
2. **(b, c)**
3. **(c)**
4. **(a, b, d)**
5. **(b)**
6. **(c)**
7. **(a)**
8. **(a, c)**

Hints and solutions

1. (b)

The given program is of Test and Set lock instruction and it generates mutual exclusion and progress but it does not guarantee bounded waiting.

2. (b, c)

SWAP is hardware-based synchronization mechanism. It is privileged/atomic instruction.

It is a lock-based mechanism and has "Lock-key" mechanism.

So, option (b), (c) are correct.

3. (c)

Both TSL and SWAP guarantees Mutual Exclusion and progress but not bounded waiting.

4. (a, b, d)

TSL and SWAP both have busy waiting which wastes CPU cycle and they also has priority inversion.

Therefore (a), (b) and (d) are correct.

5. (b)

Priority inheritance is the solution to priority inversion problem.

6. (c)

Producer's program in

Producer-consumer implementation using sleep() and wakeup() is as follows:

```
void producer(void)
{
    int itemp; in = 0;
    while(1)
    {
    itemp = Produce.item();
    if(count = = N) sleep();
    Buffer[in] = itemp;
    in = (in + 1)%N
    count = count + 1;
    if(count = = 1) wakeup(consumer);
    }
}
```

Therefore, option c is correct.

7. (a)

wait and signal are operations performed on semaphores.

8. (a, c)

Semaphores are of two types.

Binary and counting.

In Binary semaphore the value is either 0 or 1.

Counting semaphore can take any value between - ∞ to + ∞.

Branch : CSE & IT Batch : Hinglish

Operating Systems

Process Synchronization/Coordination

DPP 05

[MCQ]

1. Choose the hardware type solution for the synchronization problem from the following

(a) Lock variables
(b) Strict alteration
(c) Peterson's solution
(d) Test and set instructions

[NAT]

2. Consider the following two-process synchronization solution

X	Y

while (turn = = 1);
C.S
turn =1;
	while (turn = = 1);
	C.S
	turn =1;

The shared variable turn is initialized to zero. Then the number of incorrect statements about above solution is/are?

(I) It violates mutual exclusion.
(II) It violates progress.
(III) It violates bounded wait.

[MSQ]

3. The software type solution: strict alteration satisfies which of the following?

(a) Mutual exclusion
(b) Progress
(c) Bounded wait
(d) All of the above

[MSQ]

4. Choose the correct statements regarding peterson's algorithm from the following

(a) Peterson's algorithm satisfies only mutual exclusion but not progress.
(b) Peterson's algorithm is a solution for 2 processes.
(c) Peterson's algorithm satisfies mutual exclusion, progress and bounded wait.
(d) Peterson's algorithm is a solution for atleast 2 processes.

[MCQ]

5. Which of the following is correct regarding solution to synchronization problem:

(a) Test and set is a 2-process, software based solution.
(b) Lock variable is a 2-process, hardware based solution.
(c) Strict alteration is a 2-process, hardware based solution
(d) None of these.

[MSQ]

6. Choose the characteristics for test and set synchronization solution.

(a) It ensures mutual exclusion.
(b) It is deadlock free.
(c) It does not guarantee bounded wait.
(d) It may cause starvation.

[MCQ]

7. For a implementing synchronization for atleast 2 processes (that should satisfy mutual exclusion, progress and bounded wait), which algorithm is best recommended?

(a) Decker's algorithm
(b) Strict alteration
(c) Peterson's solution
(d) None of the above

[MSQ]

8. Which of the following is/are OS based synchronization mechanism?

(a) Sleep() and wakeup( )
(b) Wait() and signal()
(c) Monitor
(d) Swap

Answer Key

1. **(d)**
2. **(0)**
3. **(a, c)**
4. **(b, c)**
5. **(d)**
6. **(a, b, c, d)**
7. **(d)**
8. **(a, c, d)**

Hints & Solutions

1. (d)

Software type solutions for synchronization problem are lock variables, strict alteration, and peterson's solution.

Hardware type solution is test and set instructions.

2. (0)

The synchronization solution violates mutual exclusion and deadlock may occur.

3. (a, c)

Strict alteration satisfies mutual exclusion and bounded wait but not progress.

4. (b, c)

Peterson's solution satisfies all mutual exclusion, progress, and bounded wait. It is a solution for 2 processes.

5. (d)

Test and set is a hardware based solution while lock variables and strict alteration are software based solution.

6. (a, b, c, d)

All the options, mention the characteristics of the test and set synchronization solution.

7. (d)

Decker's algorithm, strict alteration do not satisfy progress while peterson's solution satisfy all M.E, progress, and bounded wait but it is limited to only two processes.

8. (a, c, d)

Sleep() and wakeup(), wait() and signal(): semaphore, and monitors are OS based synchronization mechanism. Swap is a hardware based synchronization uses lock and key to implement synchronization.

Branch : CSE & IT Batch : Hinglish

Operating Systems

Process Synchronization/Coordination

DPP 06

[NAT]

1. A counting semaphore S is initialized to 2 then following sequence of operation are performed over S.
V V V P V P P P V P P V P V V P
What will be the value of S after operations?

[MSQ]

2. Recall the readers-writers problem. Consider the following solution to this problem.

```
Writer
Wait (wrt);
        .....
        writing is performed
        .....
        Signal (wrt);
Reader
        wait (mutex);
        readcount = readcount + 1;
        if readcount = 1 then wait (wrt);
        signal (mutex);
                .....
                Reading is performed
                .....
        wait (mutex);
        readcount = readcount – 1;
        if readcount = 0 then signal (wrt);
        signal (mutex);
```

Which of the following is/are correct reading above solution?

(a) Multiple readers can read together
(b) The reader are not starved of access because of priority of the writers and vice versa.
(c) A writer gets exclusive access, i.e., while a writer is writing. No one can write or read.
(d) Deadlock is possible.

[MCQ]

3. At a particular time of computation, the value of a counting semaphore is 9. Then 20 P operation and xV operations were completed on this semaphore. If the final value of the semaphore is 5, x will be?

(a) 19 (b) 18
(c) 16 (d) 20

[MCQ]

4. A counting semaphore is initialized to 5. Then, 15 P operations and 20 signal operations are performed on S. What will be the final value of S?

(a) 0 (b) 20
(c) 5 (d) 10

[MSQ]

5. Which of the following condition must be satisfied in the classical reader-writer problem?

(a) Only one writer may write a file at a time.
(b) Only one reader may read a file at a time.
(c) If a reader is reading a file, no writer may write to it.
(d) Any number of the reader can read at a time.

[MCQ]

6. A thread that is blocked on a semaphore is awakened when another thread:

(a) Tries to block the same semaphore
(b) Tries to decrement a semaphores value ≤ 0.
(c) Tries to increment the semaphore value ≥ 0.
(d) None of these

[MSQ]

7. The strict alternation _____.

(a) Does not guarantee bounded waiting
(b) Does not guarantee progress.
(c) Does not guarantee Mutual exclusion
(d) All of these

[MCQ]

8. The bounded buffer problem is also known as

(a) Readers – writing problem
(b) Producer – consumer problem
(c) Dining – Philospher problem
(d) None of these

Answer Key

1. **(2)**
2. **(a, b, c)**
3. **(c)**
4. **(d)**
5. **(a, c, d)**
6. **(c)**
7. **(b)**
8. **(b)**

Hints & Solutions

1. (2)

S = 2

V	V	V	P	V	P	P	P	V	P	P	V	P	V	V	P
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
3	4	5	4	5	4	3	2	3	2	1	2	1	2	3	2

2. (a, b, c)

The solution efficiently synchronizes multiple readers and writers such that multiple readers can read together but a writer gets exclusive access. This implements a solution to this problem and ensures the readers are not starved of access due to priority of the writers and vice versa.

3. (c)

Initial value of semaphore = 9
Signal operation = xV
Wait operation = 20 P
Final value = 5
So,
5 = 9 + x V + 20 P
5 = 9 + x (+1) + 20 (–1)
5 = 9 + x – 20
x = 5 – 9 + 20
x = 16

4. (d)

$5 - (15 \times 1) + (20 \times 1)$

$\Rightarrow 5 - 15 + 20$

$\Rightarrow 10$

5. (a, c, d)

For the classical reader-wirter problem, the following condition must be satisfied.

→ Any number of readers may simultaneously read a file.

→ If the reader is reading a file, no writer may write it.

→ If the writer is writing a file, no reader may read it.

→ Only one writer is allowed to write the file at a time.

6. (c)

A thread that is blocked on a semaphore is awakened when another thread tries to increment the till the semaphore value becomes equal to or above 0.

7. (b)

The strict alternation guarantees mutual exclusion and bounded waiting but does not guarantee progress. Therefore, option B is correct.

8. (b)

Producer consumer problem is also known as bounded buffer problem. It is a classical example of concurrent access to shared resource.

Branch : CSE & IT Batch : Hinglish

Operating Systems

Process Synchronization/Coordination

DPP 07

[MCQ]

1. What is a compulsory step before using semaphore?

(a) Deciding final value of semaphore
(b) Initialization of semaphore
(c) Defining number of operations to be performed
(d) All of the above.

[MCQ]

2. Consider the following statements, which of the following is correct?

(a) Semaphore are atomic in nature and implemented in user's mode.
(b) Semaphore are atomic in nature and implemented in kernel mode.
(c) Semaphore are non-atomic in nature and implemented in user's mode.
(d) Semaphore are non-atomic in nature and implemented in kernel mode.

[MCQ]

3. If a semaphore's value is "– 3", then what does magnitude of "– ve" value indicate _____?

(a) Number of successful up operation.
(b) Number of successful down operation.
(c) Number of blocked processes.
(d) Number of unblocked processes.

[MCQ]

4. Processes x_1 and x_2 uses flag_critical in the following function to achieve mutual exclusion. Assuming flag_critical is initialized FALSE initially.

```
get_access
{
if (flag_critical = = FALSE)
      {
      flag_critical = TRUE;
      Critical_section();
      flag_critical = FALSE;
      }
}
```

Consider the following statement:

(i) The above routine may lead to deadlock.
(ii) It is possible for processes x_1 and x_2 to access critical section concurrently.

(a) (i) is true and (ii) is false.
(b) (ii) is true (i) is false.
(c) Both (i) and (ii) are true.
(d) Both (i) and (ii) are false.

[MCQ]

5. Consider the code given below, used by the processes x_1 and x_2 to access critical section.

The initial value of shared Boolean variable P and Q are false

x_1	x_2
while (P = = Q);	while (P! = Q);
<critical section>	<critical section>
P = !(Q);	P = Q;

Select the true statements from the following:

(i) Process x_2 can go into critical section just after one entry by process x_1 into its critical section.
(ii) Mutual exclusion is not ensured.
(iii) Process x_1 can go into critical section many times without single entry of x_2 into its critical section.
(iv) None of the above
(a) (i) & (ii)
(b) (ii) & (iii)
(c) Only (i)
(d) (i), (ii), & (iii)

[NAT]

6. Consider the two function P_i and P_j that share a variable Q with an initial value '3' execute concurrently:

$P_i()$	$P_j()$
{	{
R = Q * 2:	S = Q + 1:
Q = R;	Q = S;
}	}

What are the different possible value for variable Q at the end of execution of both process P_i and P_j?

[MCQ]

7. Match the following statements

List I	List II
A. Critical section	1. Ensuring that only one process can execute C.S.
B. Synchronization	2. atomic operation are used to ensure co-operation between processes.
C. Mutual exclusion	3. Section of code that only one process can access at once.

Matches:

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

[NAT]

8. Let S be a binary semaphore variable. Let S = 1 initially.
Assume that no blocked processes exist in the system.
The following operations are performed on semaphore S.
6 P, 8 V, 12 P, 11 V, 19 P
The number of blocked processes after executing these operations are _____.

Answer Key

1. **(b)**
2. **(b)**
3. **(c)**
4. **(b)**
5. **(c)**
6. **(4)**
7. **(b)**
8. **(19)**

Hints & Solutions

1. (b)

Semaphore initialization is compulsory step before using it. Without initializing the semaphore's value it is not possible to perform further operations on it.

2. (b)

Semaphore are atomic and are implemented in system's kernel. The semaphore values are kept in a table stored in kernel memory. A semaphore is identified by a number corresponding to a position is this table.

3. (c)

Magnitude of "–ve" value indicates the number of blocked processes.

4. (b)

(ii) is true because, both the processes x_1 and x_2 can access critical region concurrently because of

if(flag_critical = = FALSE)

x_1, x_2 can execute the above condition simultaneously and can enter C.S without leading to deadlock.

5. (c)

Process x_1 cannot go into critical section multiple times without entry of x_2.

6. (4)

I. It is given the process $P_i()$ and process $P_j()$ is executing concurrently. So, assign the unique number to operations of both the processes.

$P_i()$: $I_1 \Rightarrow R = Q * 2$; $I_2 \Rightarrow Q = R$

$P_j()$: $I_3 \Rightarrow R = Q * 2$; $I_4 \Rightarrow Q = R$

II. Now, perform the operation to find the distinct values of Q

1. I_l, I_2, I_3, $l_4 = 7$
2. I_1, I_3, I_2, $I_4 = 4$
3. I_3, I_1, I_4, $I_2 = 6$
4. I_3, I_4, I_1, $I_2 = 8$
5. I_3, I_1, I_2, $I_4 = 4$
6. I_1, I_3, I_4, $I_2 = 6$

Therefore, we have total 4 distinct values of Q: {7, 4, 6, 8}

7. (b)

- Synchronization uses p() and v() operation.
- Critical section is a section of code that only one process can access at a time.
- Mutual exclusion ensures that only one process can execute CS at any time.

8. (19)

Initially S = 1

6 P: 5 blocked processes

8 V: S = 1 & 0 blocked process

(As all six blocked processes will be resumed and 7^{th} V will make S = 1 and 7^{th} V will continue to keep S = 1)

12 P: 11 blocked processes

11V : S = 0 and 0 blocked process

19 P : S = 0 and 19 blocked processes

Branch : CSE & IT Batch : Hinglish

Operating Systems

Process Synchronization/Coordination

DPP 08

[MCQ]

1. In dining philosopher's problem, deadlock can occur when ____

(a) Two philosopher's pick there left fork and get preempted before picking right fork.

(b) Two philosopher's sitting in front of each other pick there left fork and then right fork.

(c) Four philosophers are sitting and doing nothing.

(d) All philosopher's pick there left fork and preempted before picking right fork.

[MCQ]

2. Consider the following code:

```
# define N 6
void p (int i)
{
    while (1)
    {
        T(i);
        T_f(i);
        T_f((i + 1)%N);
        x(i)
        p_f(i);
        p_f((i + 1)%N);
    }
}
```

T is a think function, T_f is a take fork function, x is a eat function, p_f is a put fork function, and i represents the philosopher. Which of the following is true about above code?

(a) Prevents Deadlock but has starvation.

(b) Has deadlock but do not have starvation.

(c) Has deadlock and starvation both.

(d) Do not have deadlock and starvation.

[NAT]

3. How many of the following instructions can run independently?

I_1: p = q * r;

I_2: q = p + z;

I_3: l = m + k;

I_4: n = a + b;

I_5: m = z + s;

[MCQ]

4. Consider the following instructions:

I_1: a = b + c;

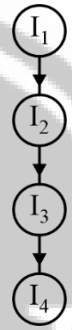
I_2: d = e * f;

I_3: k = a + d;

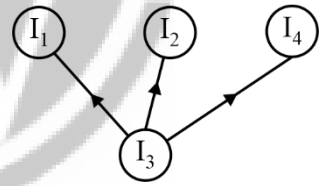
I_4: l = k + m;

Which of the following is correct precedence graph for the above instructions?

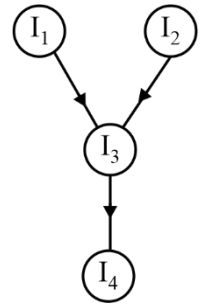
(a)



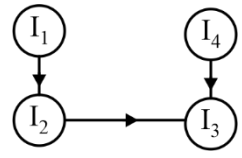
(b)



(c)



(d)



[MSQ]

5. Solve the following code by co-begin and co-end

x = 4;
co-begin
x = x + 5;
x = x + 6;
co-end;
What could be the possible final value of x?

(a) 5 (b) 9
(c) 10 (d) 15

[MCQ]

6. Busy waiting is _____.

(a) When a process polling on a variable.
(b) When a process periodically checks a variable.
(c) When a process issues an interrupt.
(d) When a process continuously checks a variable.

[NAT]

7. Consider the following concurrent program, how many levels are there in it's precedence graph?

S_1:
Parbegin
begin;
S_2; S_3;
end;
begin
S_4;
begin
S_5; S_6;
end;
Parbegin
S_7
begin
S_8; S_9
end;
Parend;
end;
Parend;
S_{10}

Answer Key

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

Hints & Solutions

1. (d)

In dining philosopher's problem if all philosophers get hungry and take their left fork and gets pre-empted before picking its right fork. Then, deadlock can occur.

2. (c)

The given code is example of classical IPC problem dining philosopher and it can suffer from deadlock and/or starvation.

Therefore, option (c) is correct.

3. (3)

I_1: p = q * r;

I_3: l = m + k;

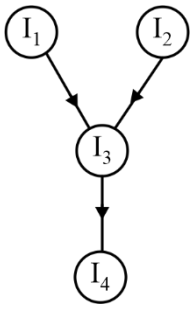
I_4 : n = a + b;

I_1, I_3, and I_4 can execute / run independently.

4. (c)

As, I_3 has 'a' and 'd' so it depends on I_1 and I_2, and I_4 has 'k' so it depends on I_3.

Therefore,



is the correct precedence graph.

5. (b, c, d)

If first we execute x = x + 5 $\Rightarrow$ 4 + 5 = 9

If it executes, x = x + 6 $\Rightarrow$ 4 + 6 = 10

If it executes x + 5 then x + 3 then x = 15.

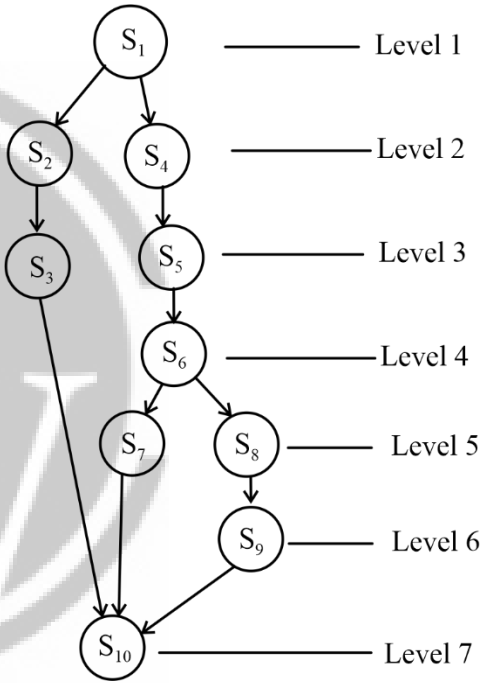
So, possible values are 9, 10 and 15.

6. (d)

Busy waiting means that the process continuously checks for the required value in a variables or an event, thereby waiting CPU cycle, as no useful work is being done.

So, option (d) is correct.

7. (7)



So, there are total 7 levels.

Branch : CSE & IT Batch : Hinglish

Operating Systems

Process Synchronization/Coordination

DPP 09

[MCQ]

1. Which of the following can prevent deadlock in dining philosopher's problem?

(a) Allow at most four philosophers' to be sitting simultaneously at the table.

(b) Allow a philosopher to pick up chopsticks only if both chopsticks are available (to do this, they must pick them up in a critical section).

(c) An odd-numbered philosopher picks up first left chopstick and then right chopstick, whereas an even-numbered philosopher picks up right chopstick and then left chopstick.

(d) All of the above.

[MSQ]

2. Dining philosopher problem is ____.

(a) Solution to producer-consumer problem.

(b) Example of concurrency-control problem.

(c) Application of semaphore.

(d) Classical IPC problem.

[MSQ]

3. Which of the following statements is/are correct?

(a) Concurrency is about doing multiple things at once.

(b) Concurrency is about dealing with multiple things at once.

(c) Parallelism is about doing multiple things at once.

(d) Parallelism is about dealing with multiple things at once.

[MSQ]

4. Which of the following condition must be satisfy if two statements are concurrent/independent?

(a) Atleast one shared variable should be present.

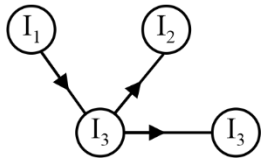
(b) No shared variable should present.

(c) Output of one statement should serve as input to other statement.

(d) Output of one statement should not serve as input to other statement.

[NAT]

5. Consider the following precedence graph:



How many nodes are dependent?

[MCQ]

6. Consider the following code by begin and end

```
a = 3
b = 4
begin:
    b = a * b
    a = a * b
end;
```

What could be the possible final values of a, b?

(a) a:{12}; b:{12}

(b) a:{12}; b:{12, 24}

(c) a:{12, 24}; b:{12}

(d) a:{12, 36}; b:{12}

[MCQ]

7. How many of the following statements is/are not independent?

I_1: a = b + c

I_2: c = d * e

I_3: e = a * g

I_4: g = e + m

I_5: m = m + f

[NAT]

8. Consider a counting semaphore variable 'X', following are the semaphore operations performed 20P, 6V, 8V, 7P, 12P, 16V. What will be the largest initial value of 'X' to keep 2 processes in suspended list?

Answer Key

1. **(d)**
2. **(b, c, d)**
3. **(b, c)**
4. **(b, d)**
5. **(3)**
6. **(d)**
7. **(3)**
8. **(7)**

Hints & Solutions

1. (d)

(a) Allow at most four philosophers' to be sitting simultaneously at the table.

(b) Allow a philosopher to pick up chopsticks only if both chopsticks are available (to do this, they must pick them up in a critical section).

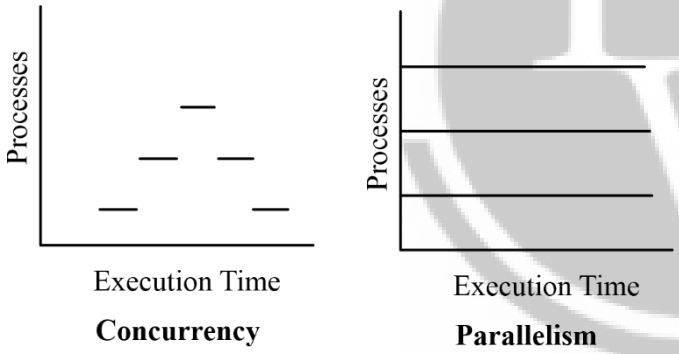
(c) An odd-numbered philosopher picks up first left chopstick and the right chopstick, whereas an even-numbered philosopher picks up right chopstick and then left chopstick.

2. (b, c, d)

Dining philosopher's problem is a classical IPC problem. Its solution can be implemented by using semaphore. It is an example of large class of concurrency control problems.

3. (b, c)

Concurrency is about dealing with lots of things at once. Parallelism is about doing lots of things at once.



4. (b, d)

If two statements are concurrent or independent then

(1) There should be no shared variable between then

(2) Output of one statement should not serve as input to other statement

5. (3)

$I_1 \rightarrow I_3$; I_3 depends on I_1

$I_3 \rightarrow I_2$ and $I_3 \rightarrow I_4$; I_2 and I_4 depends on I_3

So, 3 nodes are dependent.

6. (d)

If first it execute b = a * b $\Rightarrow$ b = 3 * 4 = 12

If first it executes a = a * b $\Rightarrow$ a = 3 * 4 = 12

If first it executes b = a * b and then a = a * b, then a = 3 * 12 = 36

So, final possible values are $\left\{\frac{12, 36}{a}, \frac{12}{b}\right\}$

7. (3)

I_1 = a = b + c

I_2 = c = d * e

Only these two statements are independent

I_3: e = a * g

I_4: g = e + m

I_5: g = m + f

These three statements are not independent

Therefore, 3 is correct.

8. (7)

–2 = X – 20 + 6 + 8 – 7 – 12 + 16

–2 = X – 20 + 6 + 1 + 4

–2 = X – 20 + 11

–2 = X – 9

–2 + 9 = X

X = 7

Branch : CSE & IT Batch : Hinglish

Operating Systems

Process Synchronization/Coordination

DPP 10

[MSQ]

1. Consider the following concurrent program:

```
begin
    S1;
    S2;
     Parbegin
       S3;
       S4;
       S5;
     Parend;
  S6;
end;
```

Which of the following statement is correct about above program?

(a) S_1, S_2, S_3, S_4, S_5 run sequentially.
(b) S_3, S_4, S_5 run concurrently.
(c) S_1, S_2, S_6 run concurrently.
(d) S_1, S_2, S_6 run sequentially.

[NAT]

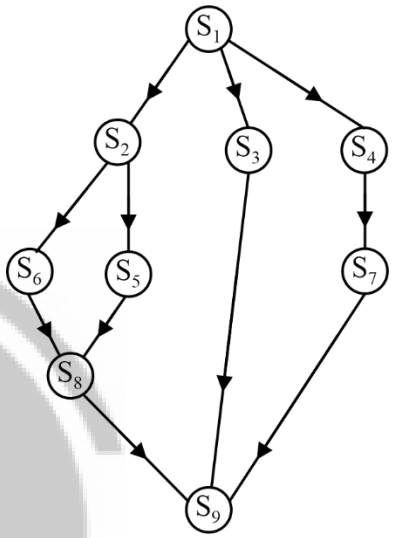
2. Consider the following concurrent program

```
S1;
Parbegin
    begin
        S2;
        S3;
        S4;
    end;
    S5;
    S6;
    begin
        S7;
        S8;
        S9;
    end;
    S10;
Parend;
S11;
```

After S_1 how many statements will start concurrently?

[MSQ]

3. Consider the given precedence graph:



Which of the following represent correct concurrent program of above precedence graph?

(a)
```
S1;
Parbegin;
  S3;
  begin
    S7;
    S4;
     Parbegin
      S2;
        Parbegin
          S6;
          S5;
        Parend;
      S8;
      Parend;
  end;
Parend;
S9;
```

(b)
```
S1;
Parbegin
    begin
      S2;
      Parbegin
        S6;
        S5;
      Parend;
    end;
      S8;
        begin
          S4;
          S7;
        end;
    S3;
Parend;
S9;
```

(c)
```
S1;
Parbegin
    begin
      S2;
       Parbegin
         S6;
         S5;
       Parend;
       S8;
    end;
    S3;
    begin;
      S4;
      S7;
    end;
Parend;
S9;
```

(d)
```
S1;
Parbegin
        begin
          S4;
          S7;
        S2;
          Parbegin
            S6;
            S5;
          Parend;
        S8;
        S3;
Parend;
S9;
```

[MSQ]

4. Consider the following program

```
main()
    {
      Parbegin
            P();
            Q();
            R();
      Parend;
    }
```

```
void P(void)
{
  Parbegin
        1();
        2();
        3();
  Parend;
}
```

```
void Q(void)
{
  Parbegin
        4();
        5();
        6();
  Parend;
}
```

```
void R(void)
{
  Parbegin
        7();
        8();
        9();
  Parend;
}
```

Which of the following output sequences are possible after the successful completion of P( ) and Q( )?

(a) 1 2 3 4 5 6 7 8 9 (b) 4 1 2 5 3 6 7 9 8

(c) 4 1 5 6 2 4 7 8 9 (d) 9 7 6 1 2 5 6 2 3

[MCQ]

5. Deadlock is _____

(a) Starvation

(b) Blocking a process for defined time

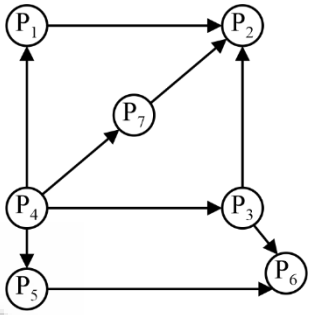
(c) Infinite waiting

(d) Utilization of CPU

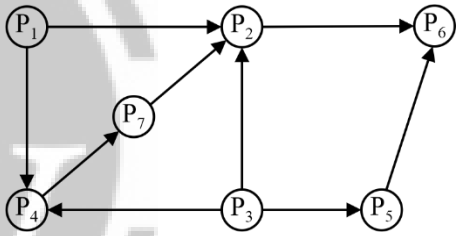
[MCQ]

6. Which of the following graph represents deadlock?

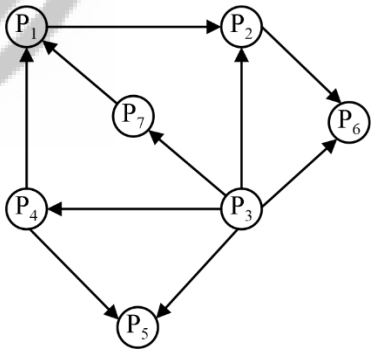
(a)



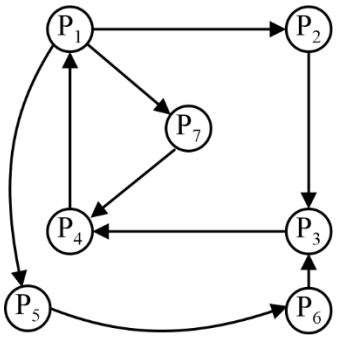
(b)



(c)



(d)



[MSQ]

7. Which of the following statement is/are correct?

(a) Deadlock is indefinite waiting.

(b) Deadlock is infinite waiting.

(c) Starvation is infinite waiting.

(d) Starvation is indefinite waiting.

[MCQ]

8. A problem encountered when a process is perpetually denied for indefinite time from necessary resources because that resource is currently used by another process. Such problem is known as _____

(a) Deadlock (b) Ageing

(c) Infinite blocking (d) Starvation

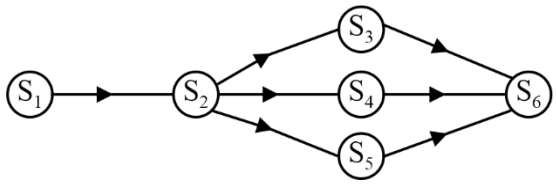
Answer Key

1. **(b, d)**
2. **(5)**
3. **(c)**
4. **(a, b)**
5. **(c)**
6. **(d)**
7. **(b, d)**
8. **(d)**

Hints & Solutions

1. (b, d)

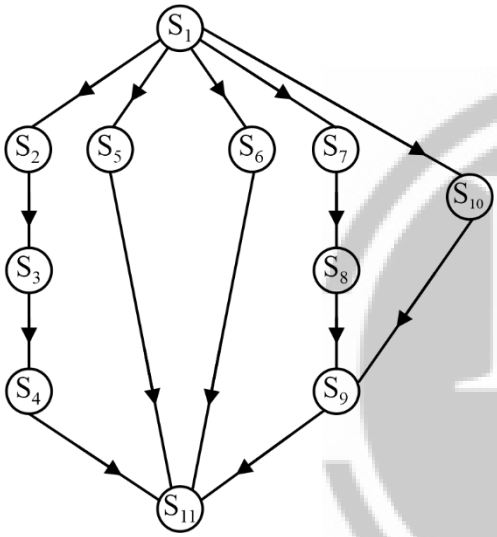
Precedence graph of above program is as follows



So, option b and d are correct.

2. (5)

Precedence graph for above program is as follows:



So, S_2, S_5, S_6, S_7, S_{10} will execute concurrently after S_1.

3. (c)

The concurrent program for given precedence graph is

S_1;
Parbegin
  begin
    S_2;
    Parbegin
      S_6;
      S_5;
    Parend;
    S_8;
  end;
  S_3;
  begin;
    S_4;
    S_7;
  end;
Parend;
S_9;

Therefore, option c is correct answer.

4. (a, b)

By using Parbegin and Parend all processes can run concurrently. So, every sequence is possible.

But in option 'c' and 'd' 4 and 2 are repeated twice respectively. Hence it is not possible.

5. (c)

Two more processes are said to be in deadlock if they wait for the happening of an event, which will never happen.

It is called infinite blocking or waiting.

6. (d)

In option d, P_1 is waiting for P_2, P_2 is waiting for P_3, P_3 is waiting for P_4, P_4 is waiting for P_1 and hence they all are waiting infinitely for each other.

Similarly, P_1, P_7, P_4 are in deadlock and P_1, P_5, P_6, P_3, P_4, P_1 are also in deadlock. Therefore, option d represents deadlock.

7. (b, d)

Deadlock is infinite waiting whereas starvation is indefinite blocking/waiting.

8. (d)

When a process is constantly denied for indefinite time from necessary resources is known as starvation.

Branch : CSE & IT Batch : Hinglish

Operating Systems

Deadlock

DPP 01

[MSQ]

1. Which of the following is not a hardware resource?

(a) Semaphore (b) Files

(c) Register (d) CPU

[MCQ]

2. If a process request is denied by OS and it is blocked forever, then the process is in _____.

(a) Starvation (b) Deadlock

(c) Ageing (d) Blocking

[MSQ]

3. What are the necessary conditions for deadlock?

(a) Mutual exclusion

(b) Hold and wait

(c) Circular wait

(d) Pre-emption

[MCQ]

4. Consider the following statements:

(i) Cycle in single instance resource is sufficient and necessary condition for deadlock.

(ii) Cycle in multi-instance resource is necessary and sufficient condition for deadlock.

Which of the following is correct?

(a) Only (i) is correct

(b) Only (ii) is correct

(c) Both (i) and (ii) are correct

(d) Both (i) and (ii) are incorrect

[MSQ]

5. In which deadlock handling strategy, deadlock can never occur?

(a) Deadlock avoidance

(b) Deadlock recovery

(c) Deadlock removal

(d) Deadlock prevention

[NAT]

6. Consider a system having 'n' resources. All these resources are shared between four processes P_0, P_1, P_2, P_3 and each process has a demand of 6, 9, 7 and 14 respectively.

What should be the maximum value of 'n' in order to lead the system to deadlock?

[MSQ]

7. Consider the following system state:

Process	Allocated	Maximum allocation
P_1	2	7
P_2	3	8
P_3	4	6

There are total 11 resources available, which of the following sequences will lead system to safe state.

(a) $P_1\ P_2\ P_3$

(b) $P_1\ P_3\ P_2$

(c) $P_3\ P_2\ P_1$

(d) $P_3\ P_1\ P_2$

[NAT]

8. Consider a system with five processes A, B, C, D and E. The requirements of resources to complete execution by A, B, C, D and E are 7, 6, 8, 12 and 11 respectively. Then, what is the minimum number of resources required to avoid deadlock in such a system?

[NAT]

9. If a system has 8 processes, each process needs maximum of 4 instances of a resources 'R', what is the maximum value of resources, so that the system is in deadlock?

Answer Key

1. **(a, b)**
2. **(b)**
3. **(a, b, c)**
4. **(a)**
5. **(a, d)**
6. **(32)**
7. **(c, d)**
8. **(40)**
9. **(24)**

Hints & Solutions

1. (a, b)

Semaphore and files are not hardware resource. Whereas, register and CPU are hardware resource.

2. (b)

If a process request is denied and the process is blocked forever, then the process is in deadlock.

3. (a, b, c)

There are four necessary conditions for deadlock:

(a) Mutual exclusion

(b) Hold and wait

(c) No-preemption

(d) Circular wait

4. (a)

Cycle in single instance resources is sufficient and necessary condition.

But cycle in multi-instance resource is necessary not sufficient condition to cause deadlock.

5. (a, d)

In deadlock prevention and deadlock avoidance strategies of deadlock handling, deadlock can never occur.

6. (32)

We know,

To avoid the deadlock the minimum resources required is:

$$\text{Number of resources} \geq \sum_{i=0}^{n} (\max - \text{need}(i) - 1) + 1$$

In order to lead to deadlock, we need to remove an resource provided to each process. So,

P_0 will have 5 resources

P_1 will have 8 resources

P_2 will have 6 resources

P_3 will have 13 resources

If any one process gets atleast one more resource then it can complete and then other process can also complete their execution subsequently.

Therefore, the maximum value of n will be:

5 + 8 + 6 + 13

n = 32

7. (c, d)

Process	Allocated	Maximum allocation	Need
P_1	2	7	5
P_2	3	8	5
P_3	4	6	2

Available = 11 – (2 + 3 + 4)

= 11 – 9

= 2

Process P_3 needs 2 resources, so remaining resources will be allocated to P_3.

Now available = 2 + 4 = 6

Any of the P_1 and P_2 can satisfy their need.

P_3 P_2 P_1 and P_3 P_1 P_2 both are safe sequences.

8. (40)

A requires 7 resources to complete its execution.

B requires 6 resources to complete its execution.

C requires 8 resources to complete its execution.

D requires 12 resources to complete its execution.

E requires 11 resources to complete its execution.

Then to avoid deadlock resources will be assigned as:

$A \rightarrow 6$

$B \rightarrow 5$

$C \rightarrow 7$

$D \rightarrow 11$

$E \rightarrow 10$

$+ 1$

40

9. (24)

Total 8 process and each process requires maximum of 4 instances of resources ‘R’.

So,

$R \leq (4 - 1) \times 8$

$R \leq 3 \times 8$

$R \leq 24$

Maximum value of R is 24, if R is 25 then there is no deadlock. So, 24 is the maximum value of resources that can cause deadlock.

Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 01

[MCQ]

1. According to abstract view of memory, memory is _____.

(a) Non-linear three-dimensional array.

(b) Linear three-dimensional array.

(c) Non-linear one-dimensional array of words.

(d) Linear one-dimensional array of words.

[MCQ]

2. What is smallest addressable unit in a memory?

(a) 1 Bit (b) 1 Byte

(c) 2^{10} byte (d) 2^2 bit

[NAT]

3. If there are total 16 words in memory and each word has a size of 8 bytes. How many bits of address is required to refer one word?

[NAT]

4. Consider the following statements:

(i) If there is a memory of size 32 KW, then number of bits required to address one word is x.

(ii) If number of bits required to address a memory are 18 bits, then the memory capacity is y KW.

Calculate x * y?

[NAT]

5. How many of the following are functions of memory manager?

(i) Memory allocation

(ii) Protection

(iii) Fragmentation

(iv) Address Translation

(v) Manage the execution of larger program in smaller memory area.

[MCQ]

6. An operating system uses the Banker's algorithm for deadlock avoidance. There are three types of resource A, B, and C allocated to three processes P_0, P_1, P_2. The below table represents the current system state.

	Allocation			Max		
	A	**B**	**C**	**A**	**B**	**C**
P_0	1	1	3	7	4	8
P_1	5	6	2	7	8	4
P_2	3	2	1	4	5	2

There are 2 units of each resource still available. The system is in safe state. Consider the following independent requests for additional resources in current state.

Request 1: P_0 request 2 units of A, 0 units of B, 1 units of C.

Request 2: P_1 request 2 units of A, 0 units of B, 2 units of C.

Which one of the following is TRUE?

(a) Request 1 can be granted, Request 2 cannot.

(b) Request 2 can be granted, Request 1 cannot.

(c) Both Request 1 and Request 2 can be granted.

(d) Neither of Request 1 and Request 2 can be granted.

[MCQ]

7. Request of any process should be granted iff the resulting state is safe otherwise it is denied, this is known as_____.

(a) Resource -Allocation Algorithm

(b) Resource -Access Algorithm

(c) Resource- Request Algorithm

(d) None of these

[MCQ]

8. Which of the following are deadlock prevention schemes?

(a) Each process request resources either in only increasing order or in only decreasing order.

(b) Whenever a process requests a resources, it does not hold any other resources.

(c) If a process is holding some resources and request another resources that cannot be immediately allocated to it, all resources being held are pre-empted.

(d) All of these

[MCQ]

9. Consider the following system.

	A	B	C	D
P_0	0	0	1	2
P_1	1	0	0	0
P_2	1	3	5	4
P_3	0	6	3	2

Allocation

	A	B	C	D
P_0	0	0	1	2
P_1	1	7	5	0
P_2	2	3	5	6
P_3	0	6	5	2

Max

A	B	C	D
1	5	2	0

Available

Which of the following Statement is/are correct.

(a) The system in unsafe state.

(b) The system in safe state.

(c) Data missing

(d) Deadlock will take place

[MSQ]

10. Consider which of the following statements is/are correct regarding deadlock?

(a) If a system is in unsafe state, the process may complete its execution without entering a deadlock state.

(b) If a process releases all its resources before requesting new resource, then deadlock and starvation both are possible.

(c) Deadlock avoidance is less restrictive than deadlock prevention.

(d) In deadlock avoidance, the request for resources is always granted if the resulting state is safe.

[MCQ]

11. For mutual exclusion to prevail in the system _____.

(a) The processor must be a uniprocessor rather than a multiprocessor.

(b) There must be at least one resource in a sharable mode.

(c) At least one resource must be held in a non-sharable mode.

(d) All of the these.

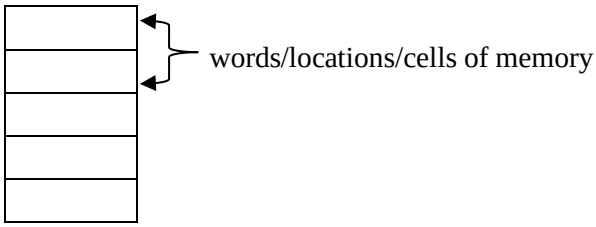
Answer Key

1. **(d)**
2. **(b)**
3. **(4)**
4. **(3840)**
5. **(3)**
6. **(b)**
7. **(c)**
8. **(d)**
9. **(b)**
10. **(a, c, d)**
11. **(c)**

Hints & Solutions

1. (d)

According to abstract view of memory or the memory from developer point of view is known as linear one-dimensional array of words.



1-D linear array

Abstract-view of memory.

2. (b)

Byte is the smallest addressable unit in memory and word-length is measured in the form of byte.

3. (4)

N = 16 {Total words in memory}

m = 8 byte = 64 bit {Size of each word in memory}

So, $n = \log_2 N$ bits

$n = \log_2 16$ bits

n = 4 bits

4. (3840)

(i) N = 32 KW

$\because n = \log_2 N$

$= \log_2(32\ KW)$

$= \log_2(2^5 \times 2^{10})$

n = 15

$\therefore$ **x = 15**

(ii) n = 18

$N = 2^n$

$= 2^{18}$

$= 2^8 \cdot 2^{10}$

= 256 KW

N = 256 KW

y = 256

$\therefore$ **y = 256**

x * y

15 * 256 = **3840**

5. (3)

Functions of memory manager includes:

(i) Memory allocation and deallocation

(ii) Memory protection

(iii) Free space management

(iv) Address translation

Goals of memory manager includes:

(i) Effective memory utilization (No wastage/Avoid fragmentation).

(ii) Manage the execution of larger programs in smaller memory area. Includes the concept of overlays and virtual memory.

6. (b)

Initially,

	Allocation			**Max**			**Need**		
	A	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**
$\mathbf{P_0}$	1	1	3	7	4	8	6	3	5
$\mathbf{P_1}$	5	6	2	7	8	4	2	2	2
$\mathbf{P_2}$	3	2	1	4	5	2	1	3	1

Available = A = 2; B = 2; C= 2.

Request 1 asks A = 2; B = 0; C = 1.

Now if Request 1 is permitted, then state would become:

	Allocation			**Max**			**Need**		
	A	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**
$\mathbf{P_0}$	3	1	4	7	4	8	4	3	4
$\mathbf{P_1}$	5	6	2	7	8	4	2	2	2
$\mathbf{P_2}$	3	2	1	4	5	2	1	3	1

Available = 0, 2, 1

None of the processes are able to satisfy their need. So, Request 1 can't be permitted.

Request 2 asks A = 2; B = 0; C = 2.

Now if Request 2 is permitted, then state would become:

	Allocation			Max			Need		
	A	B	C	A	B	C	A	B	C
P_0	1	1	3	7	4	8	6	3	5
P_1	7	6	4	7	8	4	0	2	0
P_2	3	2	1	4	5	2	1	3	1

Available = 0, 2, 0

P_1 will be executed.

After P_1, Available = 7, 8, 4

Now, P_2 and P_0 can be executed.

Hence Request 2 can be granted.

7. (c)

Resource- Request algorithm says " Request of any process should be granted iff the resulting state is safe otherwise the request will be denied".

8. (d)

Option A, Ensures circular wait condition.

Option C, Ensures that there will be no- preemption of resources that have been allocated.

Option A, Ensures hold and wait condition never occurs.

9. (b)

Available = 1, 5, 2, 0

Process P_0 can be serviced.

After P_0, Available = 1, 5, 3, 2

Now, Process P_2 can serviced.

After P_2, Available = 2, 8, 8, 6

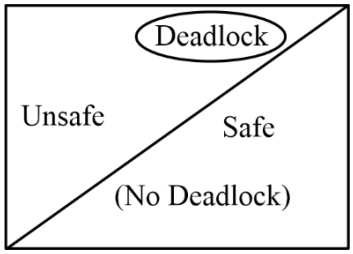
Next P_3 and P_1 can be serviced.

The system in safe state.

Therefore, option B is correct.

10. (a, c, d)

(a)



If a process is in unsafe state it can complete its execution without entering into deadlock. Correct

(b) If the process release all resources before requesting the new resource we are dissatisfying the hold and wait characteristic so deadlock not possible. Incorrect

(c) In deadlock prevention, request for a resource may not be granted even if the resulting state is safe. But in deadlock avoidance, request for a resource is granted if the resulting state is safe. Correct.

(d) In deadlock avoidance (Banker's algorithm), request for a resource is always granted if the resulting state is safe. Correct.

11. (c)

If another process request that resources (non-sharable resources), the requesting process must be delayed until the resources has been released.

Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 02

[MCQ]

1. Drawback of static loading is_____.

(a) Ineffective utilization of memory.
(b) Requires more time
(c) Need to load data before execution
(d) None of these

[MCQ]

2. Loading of modules of the program on demand is known as_____.

(a) Static loading (b) Access loading
(c) Dynamic loading (d) Page loading

[MSQ]

3. Dynamic link libraries are____ .

(a) Libraries that are linked with application during loading
(b) Libraries that are linked with application after compilation.
(c) Libraries that are linked with application before execution
(d) Libraries that are linked with the application at run time on demand

[MCQ]

4. Which of the following are address binding?

(a) Link time binding
(b) Compile time binding
(c) Load time binding
(d) Dynamic time binding

[NAT]

5. Consider a memory with the capacity of 32MB, how many bits are required to address this memory?

[MSQ]

6. Suppose 28 bits required to address a memory space, what would be the capacity of memory?

(a) 256 MB (b) 4096 Mb
(c) 512 MB (d) 2048 Mb

[MCQ]

7. Given are memory capacity (N) along with number of bits required to address that memory (n).
Match them correctly

N	n
(A) 16 MB	(i) 15 bits
(B) 32 KB	(ii) 16 bits
(C) 8 GB	(iii) 24 bits
(D) 64 KB	(iv) 33 bits

(a) A – (ii); B – (i); C – (iii); D – (iv)
(b) A – (iii); B – (i); C – (iv); D – (ii)
(c) A – (iii); B – (ii); C – (iv); D – (i)
(d) A – (ii); B – (iv); C – (iii); D – (i)

Answer Key

1. **(a)**
2. **(b)**
3. **(d)**
4. **(b, c, d)**
5. **(25)**
6. **(a, d)**
7. **(b)**

Hints & Solutions

1. (a)

Static loading leads to wastage of space for example consider the following program.

Main( )	f ()	g( )
{	{	{
if (Condition)		
f();	g()	
}	}	return 0;
		}
↓	↓	↓
Need 2KB	Need 10KB	Need 15KB

If the if (condition) gets fail in main () function, then it will waste the additional 10 + 15KB space for function f() and g().
So, static loading sometimes causes ineffective memory utilization.

2. (b)

Dynamic loading is done at run time and it is known as loading of the module of the program on demand.

3. (d)

Libraries that are linked with the application at run time on demand are known as Dynamic link libraries.

4. (b, c, d)

Addressing binding are of following types:

(i) Compile time binding

(ii) Load time binding

(iii) Dynamic time binding

5. (25)

N = 32MB

$n = \log_2 N$

$n = \log_2 (32\ MB)$

$= \log_2 (2^5 \cdot 2^{20})$

$= \log_2 (2^{25})$

$n = 25$ bits

6. (a, d)

$n = 28$

$N = 2^n$

$N = 2^{28}$

$\Rightarrow N = 2^8 . 2^{20}$

$\Rightarrow N = 256MB \approx 2048Mb$

7. (b)

A N = 16MB

n = 24 bits ….(iii)

B N = 32MB

n = 15 bits ….(i)

C N = 8GB

n = 33 bits ….(iv)

D N = 64 KB

n = 16 bits ….(ii)

Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 03

[MCQ]

1. Which of the following are contiguous allocation technique in memory management?

(a) Paging
(b) Overlays
(c) Segmentation
(d) Buddy system

[MCQ]

2. Consider the following statements:

(i) Overlaying is only possible when program can be divided into independent modules.

(ii) Overlaying is needed when process is larger than amount of memory allocated to it.

(iii) Overlaying requires special support from operating system and performed in kernel mode.

Which of the following is correct?

(a) All (i), (ii) and (iii) are correct.
(b) (ii) and (iii) are correct.
(c) (i) and (iii) are correct.
(d) (i) and (ii) are correct.

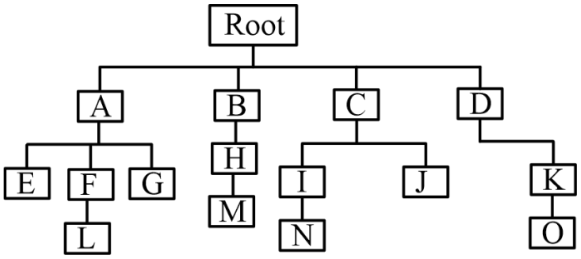
[MSQ]

3. Which of the following statements is/are TRUE?

(a) In fixed length partition, each memory partition should be of same size.
(b) Memory address protection is done with the help of registers.
(c) Free-space management is done using binary bits.
(d) In fixed-length partition, two program can reside in one partitions.

[NAT]

4. Consider the following diagram representing a program, blocks connected horizontally are independent modules and blocks connected vertically are dependent modules.



Memory requirement of each block is as follows:

Root: 10 KB

A: 5 KB	F: 9 KB	K: 7 KB
B: 6 KB	G: 6 KB	L: 4 KB
C: 4 KB	H: 2 KB	M: 5 KB
D: 9 KB	I: 7 KB	N: 8 KB
E: 8 KB	J: 3 KB	O: 6 KB

What is the minimum amount of memory (in KB) is sufficient to execute this program using overlay's when routine loading also needs 20 KB of space?

[MCQ]

5. Consider the following statements:

(i) Next fit may execute faster than first fit.
(ii) Worst fit suffers from internal fragmentation

Which of the following statements is CORRECT?

(a) Only (i) is correct.
(b) Only (ii) is correct.
(c) Both (i) and (ii) are correct.
(d) None of these

[MCQ]

6. Match the following:

(i) Fixed partition	(1) Suffers from external fragmentation.
	(2) suffers from internal fragmentation.
(ii) Variable partition	(3) Flexible degree of multiprogramming.
	(4) Preferred allocation policy is worst fit.
	(5) Preferred allocation policy is best fit.

(a) 1-(i), 2-(ii), 3-(i), 4-(ii), 5-(i)

(b) 1-(ii), 2-(i), 3-(ii), 4-(ii), 5-(i)

(c) 1-(i), 2-(i), 3-(ii), 4-(i), 5-(ii)

(d) 1-(ii), 2-(ii), 3-(i), 4-(i), 5-(ii)

[MCQ]

7. Given memory partitions in order as:

P_1: 200 KB; P_2: 400 KB; P_3: 150 KB; P_4: 500 KB. How would worst fit algorithm place processes (in order) requiring size 215 KB, 300 KB, 25 KB, 400 KB.

Note: The space left after filling a partition is not used by any process.

(a) P_4, P_1, P_3, 400 KB wait.

(b) P_1, P_2, P_3, 25 KB wait.

(c) P_4, P_2, P_3, 400 KB wait.

(d) P_3, P_2, P_4, 300 KB wait.

[NAT]

8. Consider a five memory partitions of size 100 KB, 200 KB, 300 KB, 450 KB, and 600 KB. The partitions are required to be allotted to six processes of size 180 KB, 50 KB, 210 KB, 30 KB, and 80 KB. Calculate the memory wastage using best-fit algorithm?

Answer Key

1. **(b, d)**
2. **(d)**
3. **(b, c)**
4. **(52 KB)**
5. **(c)**
6. **(b)**
7. **(c)**
8. **(1100)**

Hints & Solutions

1. (b, d)

Contiguous memory allocation technique includes:

(i) Overlays

(ii) Partitions (fixed length and variable length partition)

(iii) Buddy system

Non-contiguous memory allocation technique includes:

(i) Paging

(ii) Segmentation

(iii) Segmented-paging

2. (d)

(i) Overlaying is only possible when program can be divided into independent modules. **Correct**.

(ii) Overlaying is needed when process is larger then amount of memory allocated to it. **Correct**.

Overlays refer to a technique used to mange memory efficiently by overlaying (replacing) a portion of memory with another program. If size of process is smaller than allocated memory than it can run easily and in such case no need of overlaying is required.

(iii) Overlays are implemented by user, and no special support is needed from operating system. So, (iii) is **Incorrect**.

Therefore, option d is correct.

3. (b, c)

Memory partitions can be of different sizes. Option 'a' false.

Memory address protection is done with the help of base and limit registers. Option 'b' is True.

Free-space management is done using binary bits, if the bit is '0' the partition is free, if the bit value is '1' the partition is in use. Option 'c' is True.

In fixed-length, only one program can reside in one partition. Option 'd' is false.

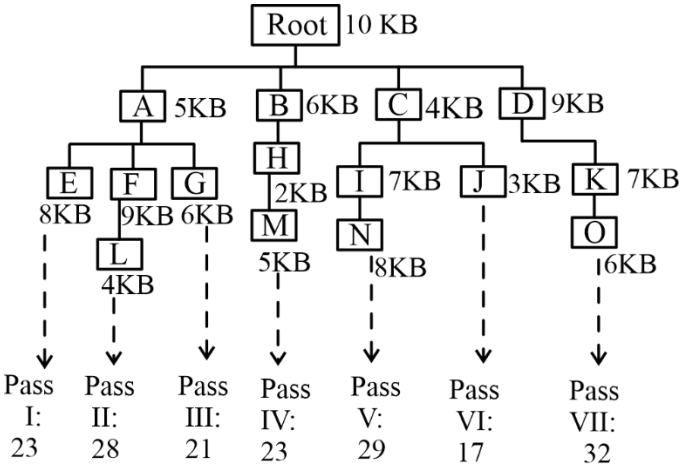
4. (52 KB)

Total space required to execute this program

= size of program + routine loading

= 99 KB + 20 KB

= 119 KB



So, pass VII requires 32 KB of space maximum of all passes, minimum amount of memory to execute this 119 KB program

= 32 KB + 20 KB (routine loading)

= 52 KB

5. (c)

(i) Next fit may execute faster than first fit.

(ii) Worst fit suffers from internal fragmentation

Both statements are correct. Therefore, c is correct option.

6. (b)

Fixed partition:

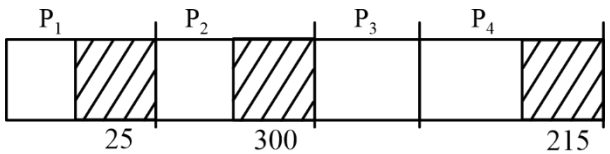
- Internal fragmentation.
- Has limited degree of multi-programing.
- Best fit is most preferred allocation policy.

Variable partition:

- Suffers from external fragmentation.
- Has flexible degree of multi-programming.
- Worst fit is most preferred allocation policy.

7. (c)

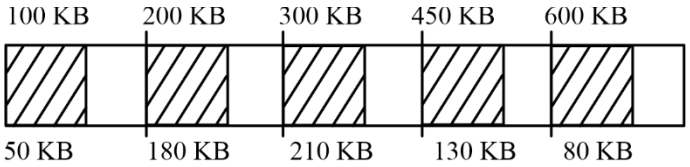
Worst fit: The largest partition is occupied first.



Order: P_4, P_2, P_1, 400 KB waits.

8. (1100)

The memory allocation is as follows:



Total space wastage = 50 + 20 + 90 + 420 + 520

= 1100

Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 04

[MSQ]

1. Problem of external fragmentation can be resolved by _____.

(a) Compaction
(b) Overlay
(c) Paging
(d) Non-contiguous allocation

[NAT]

2. Consider a memory system having six partitions 400 K, 200 K, 300 K, 250 K, 100 K and 500 K. There are four process of size 289 K, 458 K, 200 K, 300 K. What is the total size of unallocated partitions using best fit allocation policy (in K)?

[MCQ]

3. Compaction is possible with_____ address binding.

(a) Static address binding
(b) Dynamic address binding
(c) Load time address binding
(d) Compile time address binding

[MCQ]

4. Given five memory partitions of 200 K, 600 K, 300 K, 400 K, 700 K, (in order), how would the first-fit, best-fit and worst-fit algorithms place process of 312 K, 517 K, 212 K, and 526 K, (in order)? Which algorithm makes the most efficient use of memory?

Note: Space left after filling a partition is not used by any process.

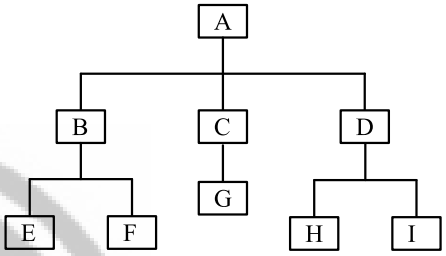
(a) First fit
(b) Best fit
(c) Worst fit
(d) All perform almost same.

[NAT]

5. Consider a memory system with five partitions: 200 K, 250 K, 500 K, 600 K and 300 K (in order). There are some process request of size 200 K, 350 K, 30 K, 25 K, 100 K (in order). How much memory space will remain unused because of internal fragmentation if first-fit allocation policy is used?

[MCQ]

6. Consider the following graphical representation of program here horizontally connected blocks are independent modules and blocks connected vertically are dependent modules.



Memory requirement of each block is as follows:

A : 10KB
B : 5KB
C : 3KB
D: 2KB
E: 5KB
F: 10KB
G: 3KB
H: 4KB
I : 6KB

Which of the following memory space is sufficient enough to execute the given program using overlays?

(a) 48 KB (b) 25KB
(c) 50KB (d) 24KB

[MCQ]

7. A computer has 1000K of main memory. The jobs arrive and finish the following sequence.

Job1 needs 100 K arrives
Job2 needs 250 K arrives
Job3 needs 400 K arrives
Job1 finishes
Job4 needs 40 K arrives
Job5 needs 250 K arrives
Job6 needs 60 K arrives

Which of the following allocation policy perform efficiently?

(a) First fit
(b) Next fit
(c) Both will perform same
(d) None of these

Answer Key

1. **(a, c, d)**
2. **(350K)**
3. **(b)**
4. **(b)**
5. **(1145K)**
6. **(c)**
7. **(a)**

Hints & Solutions

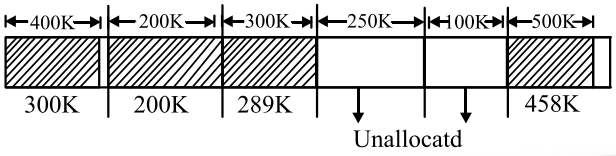
1. (a, c, d)

Compaction and non- contiguous allocation of memory are two solution for external fragmentation. Compaction is merging of non- contiguous free holes, by relocating all process to one end.

Paging is non- contiguous allocation technique used to solve problem of external fragmentation.

2. (350 K)

Using Best fit allocation policy.



So, 250+ 100 = 350K.

3. (b)

Compaction is possible with only Dynamic/run time address binding.

4. (b)

Using first fit:

312K is placed in 600K partition.

517K is placed in 700K partition.

212K is placed in 300K partition.

526K must wait.

Using Best fit:

312K is placed in 400K partition.

517K is placed in 600K partition.

212K is placed in 300K partition.

526K is placed in 700K partition.

Using Worst fit:

312K is placed in 700K partition.

517K is placed in 600K partition.

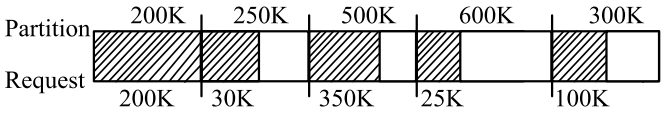
212K is placed in 400K partition.

526K must wait.

Therefore, best fit makes efficient use of memory option 'b' is correct.

5. (1145 K)

Using first fit:



Total internal fragmentation = [(200 – 200) + (250 – 30) + (500 – 350) + (600 –25) + (300 – 100)]

= [0 + 220 + 150 + 575 + 200]

= 1145

6. (c)

Total space required: 10 + 5 + 3 + 2 + 5 + 10 + 3 + 4 + 6

= 48 KB

Path I $\Rightarrow$ A $\rightarrow$ B $\rightarrow$ E $\Rightarrow$ 20 KB

Path II $\Rightarrow$ A $\rightarrow$ B $\rightarrow$ F $\Rightarrow$ 25 KB

Path III $\Rightarrow$ A $\rightarrow$ C $\rightarrow$ G $\Rightarrow$ 16 KB

Path IV $\Rightarrow$ A $\rightarrow$ D $\rightarrow$ H $\Rightarrow$ 16 KB

Path V $\Rightarrow$ A $\rightarrow$ D $\rightarrow$ I $\Rightarrow$ 18 KB

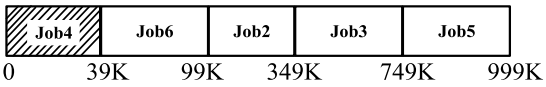
So, 25 KB will be sufficient enough to execute this program.

7. (a)

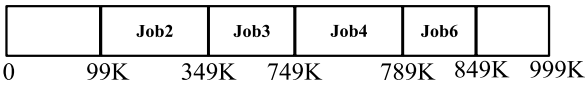
Sequence: Job1, Job2, Job3

Job 1 finishes

First fit:



Next fit:



Job 5 will wait.

Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 05

[NAT]

1. Consider a paging system with a 256MB logical address space. How many bits of logical address will be required corresponding to given LAS?

[NAT]

2. Consider a paging with 12 bits of logical address and each page is of size 2KB. Calculate the number of pages in this system?

[MSQ]

3. Consider a system with 22 bits of logical address and 12 bits of page offset. Calculate logical address space, the number of pages, and page size respectively. Which of the following is correct?

(a) 4MB, 4KB, 1K
(b) 8MB, 2KB, 2K
(c) 4096KB, 4KB, 1K
(d) 4096MB, 4MB, 2K

[MCQ]

4. Which among the following is not an operation of paging hardware?

(a) Page fault repair
(b) Memory protection
(c) Address translation
(d) none

[MCQ]

5. Consider the following statements:

S_1: In paging, logical address space is divided into fixed partitions called "pages".
S_2: In paging, physical address space is divided into fixed partition called "frames"

Which of the following is correct?

(a) Only S_1 is true
(b) Only S_2 is true
(c) Both S_1 as S_2 are true
(d) Both S_1 and S_2 are false

[MCQ]

6. Consider a system with 32-bit logical address, page size of 32KB and page table entry (PTE) size is 8 Bytes. How many pages are there in logical address space?

(a) 2^{17} (b) 2^{19}
(c) 2^{20} (d) none

[MCQ]

7. Which of the memory allocation scheme suffers from external fragmentation?

(a) Paging (b) Swapping
(c) Segmentation (d) none

[MCQ]

8. There are half as many holes as processes, the S be the average size of process and xS be the average size of holes. Then the total memory M is estimated using _____ if there are total n process in the system.

(a) $nS\left(\frac{x}{2}-1\right)$ (b) $nS\left(\frac{x}{2}+1\right)$
(c) $xS+\frac{n}{2}$ (d) $\frac{nS}{2}+x$

Answer Key

1. **(28)**
2. **(2)**
3. **(a, c)**
4. **(a)**
5. **(c)**
6. **(a)**
7. **(c)**
8. **(b)**

Hints & Solutions

1. (a)

LAS = 256MB

LA = $\log_2(256MB)$

= $\log2(2^8.2^{20})$

= 28 bits

2. (2)

LA = 12 bits

LAS = 2^{12}

LAS = $2^2.2^{10}$

LAS = 4KB

Page size = 2KB

$\therefore$ Number of pages = $\frac{\text{LAS}}{\text{Page size}}$

= $\frac{4KB}{2KB}$

= 2.

3. (a, c)

LA = 22 bits

LAS = 2^{22}

LAS = $2^2.2^{20}$

LAS = 4 MB $\approx$ 4096 KB

d = 12 bits,

So, page size = 4KB

Number of pages = $\frac{\text{LAS}}{\text{PS}} = \frac{2^{22}}{2^{12}} = 2^{10} = 1K$

4. (a)

Page fault repair: Page fault repair is not an operation

of paging hardware, It's an error which is occurred when operating system is unable to find the particular file in memory.

Hence a is correct option.

Memory protection: It's an operation of paging hardware where, memory protection prevents a process

from accessing unallocated memory in operating system.

Address translation: The addresses generated by the machine while executing in user mode are logical addresses. The paging hardware translates logical addresses to physical addresses.

5. (c)

- In paging, logical address space is divided into fixed partitions called **pages.**
- In paging, physical address space is divided into fixed partitions called 'frames'.

6. (a)

Given

Logical address (LA) = 32bits

Page size (PS) = 32KB

Page table entry size (PTES) = 8 Bytes

Number of pages in logical address space

= $\frac{2^{32}}{2^5 \times 2^{10}}$

= $\frac{2^{32}}{2^{15}}$

= 2^{17}

7. (c)

Segmentation does not Suffer from internal fragmentation but suffers from external fragmentation.

8. (b)

Number of space occupied by holes = total space – Number of space occupied by process.

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

$\Rightarrow M = \frac{n}{2} \times xS + nS$

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

Any issue with DPP, please report by clicking here:- https://forms.gle/t2SzQVvQcs638c4r5

For more questions, kindly visit the library section: Link for web: https://smart.link/sdfez8ejd80if

PW Mobile APP: https://smart.link/7wwosivoicgd4

Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 06

[MCQ]

1. A computer system has a physical address of 128 bits and a page size of 32 KB. Each page table entry contains 4 valid and 2 dirty bits along with the translation bits. If the maximum size of the page table of a process is 60MB. Calculate the size of logical address space supported by the system (in GB)?

(a) 512 (b) 32
(c) 128 (d) 64

[NAT]

2. Consider a system with 512 KB page size and each page table entry requires 8 bytes. The level of paging required to map a 30bit logical address if every page table fits into a single page are_____.

[NAT]

3. A computer system implements a 36 bits virtual address. Page size to 64KB and the size of physical memory is 34 bits. The approximate size of page table in the system is_____MB. [Assume memory is byte addressable]

[MCQ]

4. If page size is 4096 Bytes, in a paging system. A process needs 20 frames. What is the maximum possible internal fragmentation size_____.

(a) 4096 byts
(b) 4095 Bytes
(c) 2048 Bytes
(d) No internal fragmentation

[MCQ]

5. Consider a virtual address space of eight pages of 2048 words each mapped onto a physical memory of 32 frames. How many bits are there in the logical address.

(a) 13 (b) 14
(c) 10 (d) 11

[NAT]

6. Consider a virtual memory system with physical memory of 8GB. A page size of 8KB and 46-bit virtual address. Assume every page table exactly fits into a single page. If page table entry size is 4 bytes the how many levels of page tables would be required?

[NAT]

7. Consider the following statements:

I. Overlays are used to increase the size of physical memory.
II. The size of virtual memory depends on the size of main memory.
III. Aging is used to keep track of number of times a page is referenced.

How many of the above are correct statements?

[NAT]

8. A processor can support a maximum memory of 8 GB, where the memory is word addressable (each word consists 4 bytes). The size of address bus of the process is at least _____bits.

[MSQ]

9. Which of the following statement is/are correct regarding paging?

(a) Paging helps solve the issue external fragmentation
(b) Page size has no impact on internal fragmentation
(c) Paging incurs memory overheads.
(d) Multi-level paging is necessary to support pages of different sizes.

[MCQ]

10. What is basic objective of hierarchical paging?

(a) Reduce Internal fragmentation.
(b) Reduce External fragmentation.
(c) To reduce context -switch overhead.
(d) Reduce page table size overhead in memory.

Answer Key

1. **(c)**
2. **(1)**
3. **(3 to 3)**
4. **(b)**
5. **(b)**
6. **(3)**
7. **(0)**
8. **(31)**
9. **(a, c)**
10. **(d)**

Hints & Solutions

1. (c)

Given:

Page size = 32 KB

Physical address = 128 bits

$$\text{Number of frames} = \frac{\text{Physcial address sapce}}{\text{Page size}}$$

$$= \frac{2^{128}}{2^{15}} = 2^{113}$$

Frame bits = $\log_2 (2^{113}) = 113$ bits

Size of page table entry = Frame bits + Extra bits (Valid + Dirty)

= 113 + (4 + 2)

= 113 + 6

= 119 bits

Size of PTE = 15 bytes.

Now, Page table is of size = 60 MB.

$$\text{Page table entries} = \frac{\text{Page table size}}{\text{PTE size}} = \frac{60\text{MB}}{15\text{B}} = 4\text{ M}$$

Hence, Logical address space = No. of page table entry × page size

= 4 M × 32 KB

= 128 GB

Therefore, option 'C' is correct.

2. (1)

Given,

Page size = 512 KB

Page table entry size = 8 byte

First level page table size = (Number of pages in page table) × (PTE size)

$$= \frac{2^{30}\text{B}}{2^{19}\text{B}} \times 2^{8}\text{B}$$

$= 2^{19}$ B

= 512 KB

$\because$ 1st level page table size =Page size.

So, 1st level page table easily fits into a single page.

3. (3 to 3)

- Virtual address = 36 bits

  $\therefore$ Virtual address space = 2^{36} bytes

- Physical address = 34 bits

  $\therefore$ Physical address space = 2^{34} Bytes

- Number of frames = $\frac{2^{34}}{2^{16}} = 2^{18}$ Bytes

  Page table entry size = $\log_2 (2^{18}) = 18$ bits

$\therefore$ Page table size = (Number of pages in page table) × (PTE size)

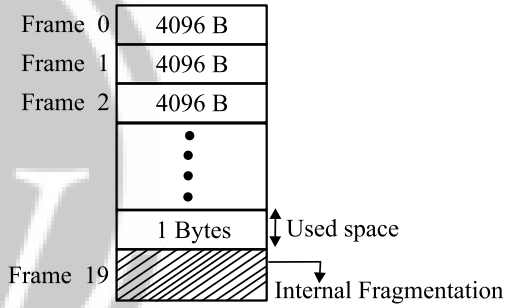
$$= \frac{2^{36}}{2^{16}} \times 18 \text{ bits}$$

$= 2^{20} \times 3$ bits

= 3 MB

4. (B)

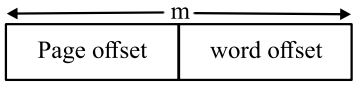
- When memory is divided into fixed sized partition then internal fragmentation may occur.
- Page size is equal to frame size so, in worst case process may store data in all allocated frame completely but still left with 1 byte of data.



$\therefore$ Maximum possible Internal fragmentation = 4096 –1 = 4095 bytes.

5. (b)

Virtual address space:



- Page size × word size = 2048 words

So, word offset = $\log_2 2048 = 11$ bits

- VAS = Number of page × page size
  = 8 × 2048
  = 2^{14} words

$\therefore$ Virtual Address = $\log_2 2^{14} = 14$ bits

6. (3)

Each page size = 8KB = 2^{13} bytes

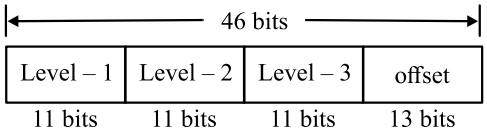
Page table entry size = 4 bytes

If page table exactly fits into a page, page table size = Page size

So, Number of page table entries

$$= \frac{\text{Page size}}{\text{Page table entry size}} = \frac{2^{13}}{4} = 2^{11}$$

So,



Therefore, there level are required.

7. (0)

Overlays are not used to increase the size of main memory. Size of virtual memory does not depends on size of main memory.

Aging is used to solve problem of saturation.

So, no statement is correct.

8. (31)

Size of memory = Number of words × Number of bits per word.

$\Rightarrow 2^{33}$ B = Number of words (address)× 4B

Number of words = $\frac{2^{33}\text{B}}{2^{2}\text{B}} = 2^{31}$ B.

So, 31 bits are required.

9. (A, C)

Page are divided into fixed size slots, so no external fragmentation, but application smaller than page size can cause internal fragmentation.

Page tables requires extra pages in memory. Therefore it is overhead to memory.

Therefore, (a) and (c) are correct.

10. (d)

Hierarchical paging or inverted paging is primarily designed to associate smaller page table with processes. In inverted paging, only one page table will be maintained for all the process avoiding the overhead of maintaining the page table for every process and reducing page table size overhead in memory.

Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 07

[MCQ]

1. The page table register should be mode of _____?

(a) Very high-speed logic
(b) Secondary memory.
(c) Large memory space.
(d) Very low speed logic.

[MCQ]

2. Consider a system implementing paging hardware with a TLB. Assume the entire page table and all the pages are in the physical memory. The effective memory access time is 400 msec. with TLB hit ratio 0.6 and search time of TLB is 20 msec, the main memory access time is _____(in msec.)

(a) 275 msec. (b) 271.43 msec.
(c) 120.5 msec. (d) 240 msec.

[MCQ]

3. Consider a 2 level paging system with TLB support. The page table has divided into 4 K pages each of size 8K words. If the physical address space has 64 M words which divided into 8 K frames. TLB access time is 20 nsec. and main memory access time is 300nsec. The CPU finds 126 page reference in the TLB out of total reference of 180. Then what is the effective memory access time?

(a) 545 nsec. (b) 440 nsec.
(c) 420 nsec. (d) 500 nsec.

[NAT]

4. Consider on operating system having 32 bits virtual address space and 32 MB physical memory. If page size is 8KB. What is the approximate size of the page table in MB?

[NAT]

5. Consider a system using paging hardware with a TLB to reduce EMAT. Assume that the entire page table and all the pages are in the physical memory. It take 30 millisec to search the TLB and 110 millisec to access the physical memory. If the TLB hit ratio is 0.6, the EMAT (in millisec) is_____.

[NAT]

6. A demand paging system takes 240 time units to service a page fault and 300 time units to replace a dirty page. Main memory access time is 20 time units. The probability of page fault is 0.4. In case of a page fault the probability of page being dirty is 0.06. The average access time _____time units.

[MCQ]

7. Consider a 32 bit machine where four-level paging scheme is used. If the TLB hit ratio is 90% and it takes 30 nsec to search the TLB and main memory access time is 100 nsec so, what is the effective memory access time in nanoseconds?

(a) 115 nsec. (b) 160 nsec.
(c) 170 nsec. (d) 180 nsec.

[MCQ]

8. A processor uses 36 bit physical address and 32 bit virtual addresses, with a page frame size of 4 Kbytes. Each page table entry is of size 4 bytes. A three-level page table is used for virtual to physical address translation, where the virtual address is used as follows:

- Bits 30 –31 are used to index into the first level page table.
- Bits 21 – 29 are used to index into the 2nd level page table.
- Bits 12 – 20 are used to index into the 3rd level page table.
- Bits 0 — 11 are used as offset within the page.

The number of bits required for addressing the next level page table(or page frame) in the page table entry of the first, second and third level page tables are respectively

(a) 20,20,20 (b) 24,24,24
(c) 24,24,20 (d) 25,25,24

Answer Key

1. **(a)**
2. **(b)**
3. **(d)**
4. **(1 to 1)**
5. **(184 to 184)**
6. **(103.44 to 103.44)**
7. **(c)**
8. **(d)**

Hints & Solutions

1. (a)

Page table base register holds the base address for the page table of the current process. Every process is associated with its own page table.

While context switching the process table PTBR has to switch to the next base address of the process immediately.

2. (b)

- Effective memory Access time = TLB_{hit} (TLB access time + Main memory access time)

  + TLB_{miss} (TLBaccess time + Page table access time + mm access time)

  EMAt = 0.6 (20 + x) + (1 – 0.6) (20 + x + x)

  400 = 0.6 (20 + x) + 0.4 (20 + 2x)

  400 = 12 + 0.6x + 8 + 0.8x

  0.6x + 0.8x = 400 – (12 + 8)

  1.4x = 380

  $x = \frac{380}{1.4}$ msec.

  x = 271. 43

3. (d)

- 126-page reference in the TLB out of total reference of 180:

  TLB hit ratio = $\frac{126}{80} = 0.7$

- Effective memory access time = TLB_{hit} (TLB_{time} + mm_{time}) + TLB_{miss} (TLB_{time} + 3*mm_{time})
  = 0.7 (20 + 300) + (1 – 0.7) (20 + 3 × 300)
  = 0.7 × 320 + 0.3 × 920
  EMAT = 500 nsec.

4. (1 to 1)

- Page Table size = (Number of entries in page table) × (Page Table Entry Size)
- Number of entries in Page Table

  $= \frac{\text{Virtual address space}}{\text{Page size}}$

  $= \frac{2^{32}}{2^{13}} = 2^{19}$ entries

- Page size is equal to frame size

∴ Number of frames in physical

$= \frac{\text{Physical memory}}{\text{Frame size}} = \frac{2^{25}}{2^{13}} = 2^{12}$ Frames

- Now, Frame bits locate page in physical memory = $\log_2 2^{12}$ = 12 bits

∴ Page Table Entry Size = Frame Bits

≈ 12 bits ≈ 2 bytes

- Page Table Size = $2^{19} \times 2^1$
  = 2^{20} Bytes
  = 1 MB.

5. (184 to 184)

EMAT = TLB hit × (TLB access time + MM access time) + TLB miss (TLB access time + Page Table access time + MM access time)

= 0.6 × (30 + 110) + (1 – 0.6) (30 + 110 + 110)

= 84 + 100

= 184 millisec.

6. (103. 44 to 103.44)

- Average access time = Page fault (% dirty × (Time to replace dirty page) + (1 – % dirty) × (Page fault services time) + (1 – Page fault) × MM access time

  = 0. 4 (0.06 × 300 + 0.94 × 240 + 0.6 ×10)

  = 0.4 (18 + 225.6) + 6

  = 97.44 + 6

  = 103.44-time units

7. (c)

- 4 level paging is used in the given system.

∴ EmAt = TLB_{hit} (TLB_{time} + MM_{time}) + (1– TLB_{hit}) (TLB_{time} + (n + 1) MM_{time})

EmAt = 0.9 (30 + 100) + (1– 0.9) (30 + 5 × 100)

= 0.9 × 130 + 0.1 × 530

= 117 + 53

= 170 nsec.

8. (d)

Physical address is 36 bits. So, number of bits to represent a page frame = 36 – 12 = 24 bits (12 offset bits as given in question to address 4 KB assuming byte addressing). So, each entry in a third level page table must have 24 bits for addressing the page frames.

A page in logical address space corresponds to a page frame in physical address space. So, in logical address

space also we need 12 bits as offset bits. From the logical address which is of 32 bits, we are now left with 32 — 12 = 20 bits', these 20 bits will be divided into three partitions (as given in the question) so that each partition represents which entry' in the i^{th} level page table we are referring to.

- An entry in level *i* page table determines 'which page table' at $(i-1)^{th}$ level is being referred.

Now, there is only 1 first level page table. But there can be many second level and third level page tables and "how many" of these exist depends on the physical memory capacity. (In actual the no. of such page tables depend on the memory usage of a given process, but for addressing we need to consider the worst case scenario). The simple formula for getting the number of page tables possible at a level is to divide the available physical memory size by the size of a given level page table.

Number of third level pages tables possible

$$= \frac{\text{Physical memory size}}{\text{Size of a third level page table}}$$

$$= \frac{2^{36}}{\text{Number of entries in a singal third level page table} \times \text{Size of an entry}}$$

$$= \frac{2^{36}}{2^9 \times 4} \therefore (\text{bits 12-20 gives 9 bits})$$

$$= \frac{2^{36}}{2^{11}} = 2^{25}$$

So, we need 25 bits in second level page table for addressing the third level page tables.

Similarly, we need to find the no. of possible second level page tables and we need to address each of them in first level page table.

Now,

Number of second level page tables possible

$$= \frac{\text{Physical memory size}}{\text{Size of a second level page table}}$$

$$= \frac{2^{36}}{\text{Number of entries in a single second level page table} \times \text{size of an entry}}$$

$$= \frac{2^{36}}{2^9 \times 4} Q\,(\text{bits 21– 29 gives 9 bits})$$

$$= \frac{2^{36}}{2^{11}}$$

$$= 2^{25}$$

So, we need 25 bits for addressing the second level tables as well.

So, answer is (D).

Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 08

[NAT]

1. Consider the following statements:

(i) Paging never suffers from internal fragmentation.

(ii) Segmentation never suffers from external fragmentation.

(iii) Segmentation may suffer from internal fragmentation.

How many of the above statements are correct?

[MSQ]

2. Consider the following segment table:

Segment No.	Base	Length
0	5519	400
1	2500	15
2	100	500
3	1257	600
4	1859	500

Which of the following logical address will generate trap addressing error?

(a) 0000110011010

(b) 0010000001111

(c) 0100100101100

(d) 1000111110011

[MCQ]

3. Virtual memory can be implemented using _____.

(a) Thrashing (b) Segmentation

(c) Demand paging (d) None of these

[MCQ]

4. If x is TLB or cache access time, y is main memory access time, and z is page fault service time or disk access time. Then, which of the fallowing relation are correct regarding access time.

(a) $x > y = z$

(b) $x < y = z$

(c) $x < y < z$

(d) $x > y > z$

[MCQ]

5. The size of virtual memory is limited by _____.

(a) Main memory size.

(b) Capacity of page table registers.

(c) Logical address space.

(d) Disk size.

[MCQ]

6. How many levels of paging is required if it is desired to limit page table to 1 page by assuming page size = 4KB, e = 4B, LA = 64 bits.

(a) 1 (b) 2

(c) 4 (d) 6

[NAT]

7. Consider a system using segmented paging architecture, where segment is divided into 16K pages each of size 4KB and segment table is divided into 8k pages of each size of 2KB (Byte addressable memory) PAS is 64MB, then calculate page table size of segment (in MB).

Answer Key

1. **(0)**
2. **(a, b)**
3. **(c)**
4. **(c)**
5. **(d)**
6. **(d)**
7. **(16)**

Hints & Solutions

1. (0)

Paging may have internal fragmentation but never suffers from external fragmentation.

Segmentation never have internal fragmentation but may have external fragmentation.

2. (a, b)

(a) 000 0110011010
↓ ↓
0 410
This will generate trap.

(b) 001 0000001111
↓ ↓
1 15
This will also generate trap.

(c) 010 0100101100
↓ ↓
2 300
No trap will be generated.

(d) 100 0111110011
↓ ↓
4 499
No trap will be generated.

Therefore, a and b are correct options.

3. (c)

Virtual memory is implemented through demand paging. Loading the pages on demand during run time is demand paging.

4. (c)

TLB or cache access time is less than main memory access time and main memory access time is less than disk access time.

$x < y < z$.

5. (d)

Virtual memory is stored in disk and so its size is limited by disk size.

6. (d)

Number of pages = $\frac{PTS}{PS}$ or $\frac{LAS}{PS}$

$= \frac{2^{64}}{2^{12}} = 2^{52}$

Level 1:

$PTS = 2^{52} * 4$

$= 2^{54} > 2^{12}$

Level 2:

Number of pages = $\frac{PTS}{PS} = \frac{2^{54}}{2^{12}} = 2^{42}$

$PTS = 2^{42} * 4$

$= 2^{44} > 2^{12}$

Level 3:

Number of pages = $\frac{PTS}{PS} = \frac{2^{44}}{2^{12}} = 2^{32}$

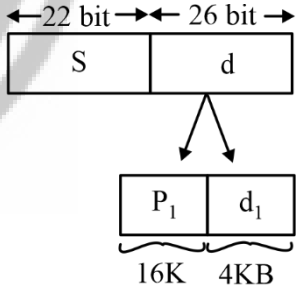
$= PTS = 2^{32} * 4$

$= 2^{34} > 2^{12} = 2^{34}$

and so on...

by solving we get 6 levels of paging and the page table size at level 6 is 2^2 which is less than 2^{12}. Hence 6 is correct answer. So, option d is correct.

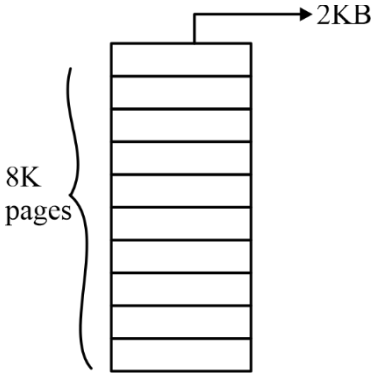
7. (16)



2^{14} 2^{12}

$14 + 12 \Rightarrow 26$ bits

Segment Table



Segment Table Size $\Rightarrow 8K \times 2KB$

$\Rightarrow 2^{13} \times 2^{11} \Rightarrow 2^{24}$

$= \quad 16MB$

Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 09

[NAT]

1. Given 3 page frames and page references are in the order:
2, 3, 4, 5, 2, 3, 6, 2, 3, 4, 5, 6. By using LRU page replacement algorithm. The number of page faults will occur.

[MSQ]

2. Which of the following are virtual memory policies?
(a) Page replacement
(b) Page reduction
(c) Page selection
(d) Page fault.

[MCQ]

3. Suppose there are 4 frames in memory and consider the following reference string:
A, B, E, D, C, E, F, A, G, E, D, C, A, C, B.
Which of the following is correct?
(a) FIFO has less page faults then LRU.
(b) LRU has less page faults then FIFO.
(c) Both FIFO and LRU has equal page faults.
(d) FIFO has 12 page fault.

[NAT]

4. Consider a system with page fault service time of 158ns and page fault hit ratio is 75%. If memory dues time is 10ns then effective memory access time (EMAT) is?

[MCQ]

5. Consider a main memory with five page frames and the following sequence of page references are 4, 9, 3, 4, 10, 2, 7, 4, 9, 10, 4, 7, 3, 2, 4. Which of the statement is true with respect to page replacement policies, first in first out (FIFO) and least recently used (LRU)?
(a) Page faults in FIFO is more than LRU.
(b) Page faults in LRU is more than FIFO.
(c) Both LRU and FIFO has some number of page faults .
(d) Page faults in FIFO has 2 more than LRU.

[MSQ]

6. The Belady's phenomenon is commonly experienced in
(a) First in first out
(b) Second chance algorithm
(c) Random page replacement algorithm
(d) Least recently used algorithm

[NAT]

7. Given reference to the following page by a program :
1, 10, 1, 2, 9, 2, 9, 8, 9, 8, 2, 3, 9, 3, 8, 9, 3, 4, 9, 3
If the program contains 4 page frames. How many page fault will occur in optimal page replacement policy?

[NAT]

8. Given 3 pages frames and page references in the order: 2, 3, 4, 5, 2, 3, 6, 2, 3, 4, 5, 6. By using optimal page replacement algorithm, the number of pages faults will be?

Answer Key

1. **(9)**
2. **(a, c)**
3. **(b)**
4. **(47)**
5. **(c)**
6. **(a, b, c)**
7. **(7)**
8. **(7)**

Hints & Solutions

1. (9)

Page reference: 2 3 4 5 2 3 6 2 3 4 5 6

Frame 1 →	2	2	2	5	5	5	6	6	6	6	6	6
Frame 2 →		3	3	3	2	2	2	2	2	4	4	4
Frame 3 →			4	4	4	3	3	3	3	3	5	5
	PF	PF	PF	PF	PF	PF	PF	PH	PH	PF	PF	PH

Total number of page faults = 9.

2. (a, c)

OS need to decide on policies on page faults concerning:

→ page selection (when to bring in)

→ page replacement (what to replace)

So, 'a' and 'c' are correct answer.

3. (b)

Using FIFO:

	A	B	E	D	C	E	F	A	G	E	D	C	A	C	B
F_1	A	A	A	A	C	C	C	C	C	E	E	E	E	E	B
F_2		B	B	B	B	B	F	F	F	F	D	D	D	D	D
F_3			E	E	E	E	E	A	A	A	A	C	C	C	C
F_4				D	D	D	D	D	G	G	G	G	A	A	A
	PF	PF	PF	PF	PF	PH	PF	PF	PF	PF	PF	PF	PF	PH	PF

Total page fault = 13

Using LRU:

	A	B	E	D	C	E	F	A	G	E	D	C	A	C	B
F_1	A	A	A	A	C	C	C	C	G	G	G	G	A	A	A
F_2		B	B	B	B	B	F	F	F	F	D	D	D	D	D
F_3			E	E	E	E	E	E	E	E	E	E	E	E	B
F_4				D	D	D	D	A	A	A	A	C	C	C	C
	PF	PF	PF	PF	PF	PH	PF	PF	PF	PH	PF	PF	PF	PH	PF

Total page fault = 12

LRU has less page fault than FIFO.

4. (47)

EMAT = (1 – P)M + PS

= (0.75)10 + 0.25(158)

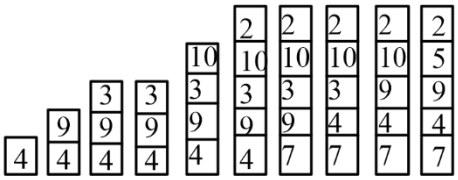
= 47 ns

5. (c)

FIFO

× × × ✓ × × × × × ✓ ✓ ✓ × ✓ ✓

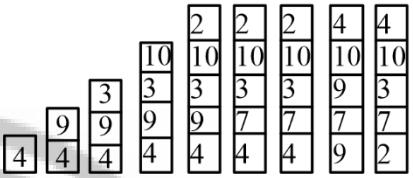
4 9 3 4 10 2 7 4 9 10 4 7 3 2 4



LRU:

× × × ✓ × × × ✓ × ✓ × ✓ ✓ × ✓

4 9 3 4 10 2 7 4 9 10 4 7 3 2 4



Both FIFO and LRU has same number of page faults

6. (a, b, c)

Belady's anomaly can never occur in LRU and optimal page replacement algorithm because these two algorithm belong to a class of stack-based page replacement algorithm.

7. (7)

1 10 1 2 9 2 9 8 9 8 2 3 9 3 8 9 3 4 9 3

× × ✓ × × ✓ ✓ × ✓ ✓ ✓ × ✓ ✓ ✓ ✓ ✓ × ✓ ✓

2 2 2 3 3

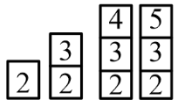
10 10 10 8 8 4

1 1 1 9 9 9

8. (7)

× × × × ✓ ✓ × ✓ ✓ × × ✓

2 3 4 5 2 3 6 2 3 4 5 6



Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 10

[MCQ]

1. A 32-bit address system, used a paged virtual memory; the page size is 2KBytes. What is the virtual page and the offset in the page (in decimal) for the virtual address 0x00030f40 respectively?

(a) 95, 2008 (b) 97, 1856

(c) 94, 1732 (d) 98, 2112

[MCQ]

2. Suppose, we have a page-reference string for a process with x frames (initially all empty). The page-reference string has length s, and y distinct page number occur on it, then,

(i) What is the maximum number of page fault?

(ii) What is the minimum number of page fault?

(a) (i) – x (ii) – y

(b) (i) – y (ii) – x

(c) (i) – s (ii) – y

(d) (i) – s (ii) – x

[NAT]

3. If MRU is used as page replacement policy in the system, and there are initially four-page frames for the following string of page reference.

4 2 5 1 3 5 6 2 3 2 3 4 5 2 3

What will be the value of page fault by page hits.

[NAT]

4. Consider a system with 3 frames and using LRU page replacement policy for the following page reference string.

3 2 3 4 1 4 2 4 1 2

What will be effective memory access time if time for accessing TLB is 30 ns and for accessing main memory is 90 ns.

[MCQ]

5. When will a page fault occur?

(a) When process tries to access a page which was not in CPU.

(b) When process tries to access a page which was not in disk.

(c) When process tries to access a page which was not in main memory.

(d) None of these.

Answer Key

1. **(b)**
2. **(c)**
3. **(4)**
4. **(156)**
5. **(c)**

Hints & Solutions

1. (b)

For a page size of N bytes the number of bits in the offset field is $\log_2 N$. In case of a 2KBytes page, there are:

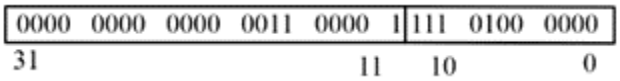
$\log_2 2^{11} = 11$ bits

Therefore, the number of bits for the page number is:

$32 - 11 = 21$

this means a total of $2^{21} = 2M$ pages.

The binary representation of address is:



The given virtual address identifies the virtual page number 0x61 = $(97)_{10}$.

The offset inside the page is 0x740 = $(1856)_{10}$.

2. (c)

Maximum number of page fault could occur equals length of page reference string (s) (No page Hit).

Minimum number of page fault occur equals to distinct page number appeared (y).

So, (c) is correct option.

3. (4)

Page Reference	4	2	5	1	3	5	6	2	3	2	3	4	5	2	3
Frame 1	4	4	4	4	4	4	4	4	4	4	4	4	5	2	2
Frame 2		2	2	2	2	2	2	2	3	2	3	3	3	3	3
Frame 3			5	1	3	5	6	6	6	6	6	6	6	6	6
	PF	PF	PF	PF	PF	PF	PF	PH	PF	PF	PF	PH	PF	PF	PH

Total Page Fault = 12.

Total Page hit = 3.

So, $\frac{\text{page fault}}{\text{page hit}} = \frac{12}{3} = 4$

4. (156)

LRU is used.

Page Reference	3	2	3	4	1	4	2	4	1	2
Frame 1	3	3	3	3	1	1	1	1	1	1
Frame 2		2	2	2	2	2	2	2	2	2
Frame 3				4	4	4	4	4	4	4
	PF	PF	PH	PF	PF	PH	PH	PH	PH	PH

Total page hits = 6

Total page miss/fault = 4

Hit ratio = 60%

So,

p = 0.6

Hit memory time = 30 + 90 = 120 ns.

Miss memory Time = 30 + 90 + 90 = 210 ns

Therefore, EAT = P × Hit + (1 - P) × miss

= 0.6 × 120 + (0.4) × 210

= 72 + 84

= 156 ns.

5. (c)

The page fault will occur when the process tries to access a page which was not in main memory.

Branch : CSE & IT Batch : Hinglish

Operating Systems

Memory Management

DPP 11

[MCQ]

1. Consider the following virtual addresses:
784, 224, 634, 734, 546, 978, 444, 299, 712, 526
If the page size is 150 bytes, then the reference string will be.
(a) 7, 2, 6, 7, 5, 9, 4, 2, 7, 5
(b) 7, 2, 6, 5, 9, 4, 6, 5
(c) 6, 1, 3, 4, 5, 9, 7, 6
(d) 5, 1, 4, 3, 6, 2, 1, 4, 3

[NAT]

2. Consider the following virtual addresses:
709, 540, 612, 311, 456, 908, 806, 708, 580, 250, 412,
If the page size of 200B, then what be the sum of length of corresponding reference string and number of unique pages in the reference string.

[MCQ]

3. Excessive paging activity results in _____.
(a) Compaction (b) Thrashing
(c) Segmentation (d) Paging

[MSQ]

4. Which of the following statements are correct?
(a) High thrashing can lead to deadlock.
(b) Lack of frames in main memory causes thrashing.
(c) Increasing memory frames can reduce page faults.
(d) High degree of multi programming can cause poor throughput.

[MCQ]

5. If the size of working set window (Δ) for a set of pages is 4. The following is the page reference string:
5, 4, 3, 1, 5, 4, 5, 3, 4, 5
What could be the minimum page demand of process at any time?
(a) 4 (b) 3
(c) 2 (d) 1

[MSQ]

6. Which of the following technique is/are used to handle thrashing and make system thrashing free?
(a) Decreasing page fault frequency
(b) working set model
(c) Compaction
(d) Page replacement policy

[MCQ]

7. Working set (t, k) at an instant of time t is
(a) The set of k reference with high frequency
(b) The set of k future reference that the OS will make in next t unit of terms.
(c) The k set of page that have been reference in the last t time units.
(d) None of these

❑❑❑

Answer Key

1. **(d)**
2. **(14)**
3. **(b)**
4. **(a, b, c, d)**
5. **(b)**
6. **(a, b)**
7. **(c)**

❑❑❑

Hints & Solutions

1. (d)

Page size is 150 bytes.

So,

0 – 149 : P_0

150 – 299 : P_1

300 – 449 : P_2

450 – 599 : P_3

600 – 749 : P_4

800 – 899 : P_5

900 – 1049 : P_6

So, the corresponding reference string will be;

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

Therefore option 'D' is correct.

2. (14)

Page size is 200 B

So, 0 – 199 : P_0

200 – 399 : P_1

400 – 599 : P_2

600 – 799 : P_3

800 – 999 : P_4

So, the corresponding reference string will be

3, 2, 3, 1, 2, 4, 3, 2, 1, 2

Length of reference string = 10

Number of unique pages in reference string

= < 1, 2, 3, 4 > = 4

So, 10 + 4 = 14

3. (b)

Excessive/ high paging activity result in thrashing. In multiprogramming, there can be a scenario when the system spends most of its time shuffling pages between the main memory and the secondary memory due to frequent page faults. This phenomenon is knowns as thrashing.

4. (a, b, c, d)

A → High thrashing can cause deadlock. Option 'A' correct.

B → Lack of frames/or less main memory space causes thrashing. Option 'B' 'C' correct.

D → High degree of multi programming cause thrashing and it leads to poor throughput. Option 'D' correct.

5. (b)

At time 0,

$\underbrace{5\ 4\ 3\ 1}_{\Delta=4}\ 5\ 4\ 5\ 3\ 4\ 5\ 5\ 5\ 2\ 3\ 4\ 5$

Demand of process = 4 page

At time 1,

$5\ \underbrace{4\ 3\ 1\ 5}_{\Delta=4}\ 4\ 5\ 3\ 4\ 5$

Demand of process = 4 page

At time 2,

$5\ 4\ \underbrace{3\ 1\ 5\ 4}_{\Delta=4}\ 5\ 3\ 4\ 5$

Demand of process = 4 page

At time 3,

$5\ 4\ 3\ \underbrace{1\ 5\ 4\ 5}_{\Delta=4}\ 3\ 4\ 5$

Demand of process = 3 page

At time 4,

$5\ 4\ 3\ 1\ \underbrace{5\ 4\ 5\ 3}_{\Delta=4}\ 4\ 5$

Demand of process = 3 page

At time 5,

$5\ 4\ 3\ 1\ 5\ \underbrace{4\ 5\ 3\ 4}_{\Delta=4}\ 5$

Demand of process = 3 page

At time 6,

$5\ 4\ 3\ 1\ 5\ 4\ \underbrace{5\ 3\ 4\ 5}_{\Delta=4}$

Demand of process = 3 page

So, minimum demand = 3

6. (a, b)

Thrashing is a situation in which the system is spending a major portion of its time servicing page faults rather than actually processing the request. This impacts system 's performance extensively.

Working set model and decreased page fault frequency can handle thrashing and make system thrashing free.

7. (c)

Working set (t, k) at an instant of time t is the k set of pages that have been referenced in the last t time units.

❑❑❑

Branch : CSE & IT Batch : Hinglish

Operating Systems

File System & Device Management

DPP 01

[MCQ]

1. What is cluster in a disk architecture?

(a) It is a group of one or more sectors on same track.
(b) It is a collection tracks on same cylinder.
(c) It is a group of one or more track one same platter.
(d) It is a collection of platter on the disk.

[NAT]

2. Consider the following disk specifications.
Number of platters = 8
Number of tracks per surface = 512
Number of sectors per track = 4096
Sector offset = 15
Calculates the unformatted capacity (in GB)?

[NAT]

3. In a computer system, a disk track can hold 32 sectors each having size of 256 bytes. The relation rate 2000 rpm. The data rate of disk is_____KB.

[MCQ]

4. A disk rotates at 7200 RPM. It has 500 sectors of 512 bytes each around the outer cylinder. How long does it take to read a sector?

(a) 14.87 μ sec. (b) 15.67 μ sec.
(c) 13.87 μ sec. (d) 16.67 μ sec.

[MCQ]

5. Consider the following disk specifications:

- A 10 GB disk rotates at 10,000 rpm
- Data transfer rates is 10^7 bytes/sec.
- Average seek time 8 ms.
- Blocks size is 32 KB.

What is the average services time to access a single disk blocks form a random location on the disk?

(a) 15.38 ms (b) 14.28 ms
(c) 16. 48 ms (d) 13.18 ms

[NAT]

6. A hard disk has 20 surface with 100 tracks for surface and 16 sectors per track. The disk transfer rate is 10^6 bytes/sec, page size is 512 bytes and the process size is 2048 bytes, and seek time is 25 msec. The disk is rotating at 300 rpm. What is the total swap time (in ms) ? [upto three decimal places]

[NAT]

7. Consider a typical disk that rotates at 20000 RPM and has a transfer rate of 32kBps. If the average seek time is 6 msec. and the controller's transfer time is twice of the disk transfer time, the average time need to read or write a 1024 bytes sector of disk is_____msec. (upto 1 decimal places)

[MSQ]

8. Two steps performed by operating system to uses a disk to hold disk its files are_____.

(a) partitioning
(b) Logical formatting
(c) Caching
(d) Swapping

Answer Key

1. **(a)**
2. **(512)**
3. **(273 KB)**
4. **(d)**
5. **(b)**
6. **(70.096)**
7. **(103.5 msec.)**
8. **(a, b)**

Hints & Solutions

1. (a)

Cluster is a group of one or more sectors on same tracks.

2. (512)

Disk capacity $= 8 \times 512 \times 4096 \times 32$ KB

$= 2^3 \times 2^9 \times 2^{12} \times 2^{15}$

$= 2^{39}$

$= 512$ GB

3. (273 KB)

Here Rotation rat is 2000 RPM means 2000 relations in 60 sec.

So, in 1 sec. $= \frac{2000}{60}$ rotations,

And time taken by 1 rotation $= \frac{60}{2000} = 0.3$ sec.

Track capacity $= 32 \times 256$ bytes

Data rate $= 32 \times 256 \times \frac{2000}{60} = 273$ KB (approx.)

4. (d)

Given, Rotation speed = 7200 RPM

60 second $\rightarrow$ 7200 rotation

1 rotation $\rightarrow \frac{60}{7200} = 0.00833$ sec.

$= 8.33$ m sec.

It 1 rotation, we can read 1 track data, so we can read 500 sectors in on rotation.

Time to read a sector = 8.33/500

$= 16.67$ m sec.

5. (b)

$T_{seek} = 8$ ms.

Rotational time $= \frac{\text{Rotatinal Latency}}{2}$

$= \frac{1}{2}\left(\frac{60}{10,000}\right)$ sec. $= 3$ msec.

Transfer time $= \frac{32 \times 1024}{10^7} = 3.2768$ ms.

≈ 3.28 ms

Strict time = 8ms + 3ms + 3.28 ms

$= 14.285$ms

6. (70.096 msec.)

10^6 bytes are transferred is second

So, 2048 bytes are transferred in $\frac{2048}{10^6} = 2.048$ m sec.

Seak time = 25 ms

RPM = 3000

60 sec. $\rightarrow$ 3000 rotation

1 rotation $\rightarrow \frac{60}{3000} = 20$ msec.

Thus, average rotational latency $= \frac{20}{2} = 10$

Swap time = 2× (Seek time + Average rotational latency + transfer time)

$= 2 \times (25\text{ms} + 10\text{ms} + 2.048\text{ms})$

$= 2 \times 37.048$ msec.

$= 74.\ 0.496$ msec.

7. (105. 5 msec.)

Average seek time = 6 msec.

Q There are 20000 rotations in 60 sec.

So, 1 rotation in $\frac{60}{20000} = 3$ msec.

Average rotational delay $= \frac{1}{2} \times 3$ msec. $= 1.5$ msec.

Sector size = 1024 byte

Transfer rate $= 32 \times 2^{10}$ bytes per second.

1024 bytes $\rightarrow \frac{1024}{32} \rightarrow 32$ sec.

Controllers transfer time $= 2 \times 32 = 64$ msec.

Average time to read/ write = 6 + 1.5 + 32 + 64

$= 103.\ 5$ msec.

8. (a, b)

Partitioning of logical formatting are two steps performed by operating system to use a disk to hold its files.

Any issue with DPP, please report by clicking here:- https://forms.gle/t2SzQVvQcs638c4r5

For more questions, kindly visit the library section: Link for web: https://smart.link/sdfez8ejd80if

PW Mobile APP: https://smart.link/7wwosivoicgd4

Branch : CSE & IT Batch : Hinglish

Operating Systems

File System & Device Management

DPP 02

[NAT]

1. Consider a disk queue with request for I/O to block on cylinders 53, 98, 140, 59, 115, 185, 197, 86. The SCAN algorithm as used. The head is initially at 100 moving towards left. The cylinders are numbered from 0 to 199. The total head movement incurred while servicing these request is_____.

[MCQ]

2. Consider the following statements:

S_1: Loop instructions cannot be interrupted till they complete.

S_2: Nearest cylinder next disk scheduling strategy gives the best through put in comparison to first come first serve scheduling strategy.

S_3: Using large file block size in a fixed block size file system leads poor disk through put.

Which of the above statements are incorrect?

(a) Only S_1 (b) Only S_1, S_2 and S_3

(c) Only S_1 and S_3 (d) Only S_2

[NAT]

3. Consider a disk has 100 numbered from 0 to 99. At some time the disk arm is at cylinder 40 and there is a queue of disk access request for cylinder 16, 35, 25, 60, 80, 90 and 20. If Shortest seeks time first (SSTT) is being used for scheduling disk access, the request for cylinder 60 is serviced after servicing the number of request _____.

[NAT]

4. An unix style I-node has 10 direct pointers and two single 5 double and 2 triple Indirect pointers. Disk block size is of 8 KB, disk block address is 32 bits. The maximum possible file size is _____TB.

[NAT]

5. Consider a disk queue with request sequence 125, 85, 46, 74, 80, 112, 21, 53, 140, the initial position of head is at 50 and it is moving towards right. The cylinder are number 0 to 150. Calculate the total time required to serve this request. When 4 ms time is needed to move head from one cylinder to another. The C-lock algorithm is used.

[MSQ]

6. Choose the correct statements from following

(a) A record is a collection of related fields that can be treated as a unit by some application program.

(b) A file is a collection of similar records.

(c) A database is a collection of non-related data.

(d) A field is the basic element of data

[MCQ]

7. Each Inode in a file system has 6 direct pointers to disk blocks, 4 single-indirect pointers to disk blocks, 3 double-indirect pointers to disk blocks and nothing else. A disk block is 500 bytes, and a pointer to disk block is of 10 bytes. The entire disk consists 17,000,000 bytes at most. Calculate the maximum size [in byte] of a file in this file system.

(a) 3850000 (b) 3853400

(c) 3853200 (d) 3853000

[MCQ]

8. In a UNIX OS, each data block is of 512 bits, each node has 5 direct data block addresses and three additional addresses. One for single indirect block, one is for double indirect block and one is for triple indirect block. Each block is addressed with 128-bit. Calculate the total size of a file possible in the file system (in k-bits).

(a) 81.91 to 81.92 (b) 82.91 to 82.92

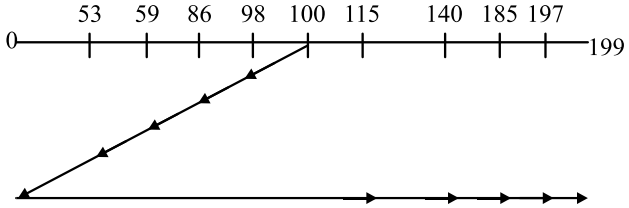
(c) 80.91 to 80.92 (d) None of these

Answer Key

1. **(299)**
2. **(c)**
3. **(4 to 4)**
4. **(128 to 128)**
5. **(936)**
6. **(a, b, d)**
7. **(d)**
8. **(d)**

Hints & Solutions

1. (299)



Total head movement = (100 – 98) + (98 – 86) + (86 – 59) + (59 – 53) + (153 – 0) + (115 – 0) + (140 – 115) + (185 – 140) + (197 –185) + (199 – 197)

= 2 + 12 + 27 + 6 + 53 + 115 + 25 + 45 + 12 + 2

= 299

2. (c)

S_1: CPU checks for interrupt after completion of current instruction execution. So, If loop certain more than 1 instruction, it can interrupt in between.

Given statements is false.

S_2: Nearest cylinder next disk scheduling gives best through but may lead to starvation.

S_3: Vising large file block size in a fixed block size file system.

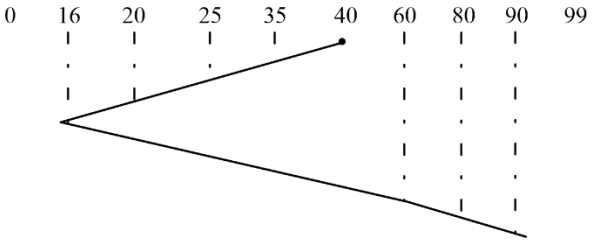
Leads to better disk through put poor disk space utilization.

- To store small data, we have to allocate complete block to it.

So, Gives statement is false.

3. (4 to 4)

- Shortest seeks time first (SSTF) is used for request 16, 35, 25, 60, 80, 90 and 20.
- Disk access request diagram.



∴ 60 is serviced after servicing 4 request.

4. (128 to 128)

$$\text{Number to disk block address} = \frac{\text{Disk block size}}{\text{Disk block address size}}$$

$$= \frac{8KB}{4B} = 2\text{ K} = 2^{11}$$

Now, tiple pointer will indicate maximum possible file size.

Hence, maximum possible file size,

$= 2 \times (2^{11}) \times (2^{11}) \times 2^{13}$ Bytes

$= 2^7 \times 2^{40}$ Bytes

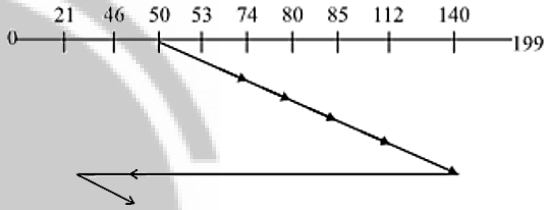
= 128 TB

5. (936)

Total head movement = (53 – 50) + (74 – 53) + (80 – 74) + (85 – 80) + (112 – 85) + (125 – 112) + (140 – 125) + (140 – 21) + (46 – 21)

= 3 + 21 + 6 + 5 + 27 + 13 + 15 + 119 + 25.

= 234



Time required for 1 head movement = 4 ms

Total time required = 234 × 4

= 936 ms

6. (a, b, d)

A data base is a collection of related data.

7. (d)

Each pointer is 10 bytes, so each disk block can contain

$\frac{500}{10} = 50$ pointers

An Inode can reference 5 blocks directly and each single indirect pointer references a block which have 50 pointers.

So, a total of 200 blocks can be referenced by 4 single indirect pointers.

similarly, each double-indirect pointer references a block of 50 pointers, so each of them again reference a block of 50 pointers, means 2500 blocks in total. So, for 3 double-indirect pointers, a total of 7500 blocks referenced by them.

Total: 6 + 200 + 7500 = 7706

which mean, 7706 × 500 = 3853000

8. (d)

Total file size =

$$\left[\text{Direct DBA} + \text{No. of single indirect pointers}\left(\frac{\text{Data Block size}}{\text{DBA}}\right) + \text{No. of double indirect pointer}\left(\frac{\text{Data block size}}{\text{DBA}}\right)^2 + \text{No. of triple indirect pointer}\left(\frac{\text{Data Block Size}}{\text{DBA}}\right)^3 \times\right] \text{Data Block Size}$$

Data block Size = 512 bits

$\left(\frac{\text{Data Block Size}}{\text{DBA}}\right)$ = No. of disk block address stored inside one block.

Maximum file size

$= [5 + 1(4) + 1(4)^2 + 1(4)^3] \times 512$

$= [5 + 4 + 16 + 64] \times 512$

$= 45.568$ K bits

Branch : CSE & IT Batch : Hinglish

Operating Systems

System Calls and Threads

DPP 01

[NAT]

1. Consider the below code segment. Total new process created is _____

```
#include<stdio.h>

int main ()

{

  int i;

  for (i=0; i<5, i+=2)

{

  switch(i)

  {

     Case 0: fork();

     Case 1: fork(); fork();

     Case 2: fork(); fork(); break;

     Case 3: fork(); fork(); fork(); break;

     Case 4: for (j = i, j >= 1, j --) fork();

  }

 }

return 0;

}
```

[NAT]

2. Consider the following code:

```
main()

{

  for(int i = 1, i<= 5; i++)

  fork ()

}
```

What will be the number of child processes/newly created processes for the above code?

[MCQ]

3. Match List – I and List – II and select the correct answer using the code given below:

List – 1	List – II
A. Context switching	1. Process creation
B. Degree of multi programming	2. Dispatcher
C. Message passing	3. Long term scheduler
D. Fork()	4. Inter process communication

Codes:

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

[MCQ]

4. To access the services of operating system, the interface is provided by the

(a) System call
(b) API
(c) Library
(d) Assembly Instructions.

[MCQ]

5. The following program fragment prints the strings "GATE2023" how many number of times?

```
main ()
  {
      fork ();
      fork ();
      printf("GATE2023"\n);
  }
```

(a) 1 (b) 2
(c) 4 (d) 8

[MCQ]

6. Which of the following scheduling can be done by thread library?

(a) User thread scheduling
(b) Process Scheduling
(c) Kernel thread scheduling
(d) None of the above.

[MCQ]

7. Consider the following code:

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

How many child processes created by the above code:

(a) 15 (b) 4
(c) 32 (d) 16

[MCQ]

8. Consider the following statements with respect to user-level threads and kernel – level threads.

S_1: Context switching in kernel level threads is faster than user level threads.
S_2: If one thread of user – level gets blocked, entire process also gets blocked.
S_3: kernel – level threads can be scheduled independently.

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

[MSQ]

9. What are the types of system calls in an operating system?

(a) Process control, file management
(b) Device management.
(c) Communication.
(d) Information maintenance.

[MCQ]

10. Which of the following operations require the executing code to be operating with Kernal mode?

(a) Performing semaphore 'P' operation
(b) Making system call
(c) Disabling interrupt
(d) Both (a) and (c)

Answer Key

1. **(2047)**
2. **(31)**
3. **(b)**
4. **(a)**
5. **(c)**
6. **(a)**
7. **(a)**
8. **(d)**
9. **(a, b, c, d)**
10. **(c)**

Hints & Solutions

1. (2047)

i = 0; 0 < 5 true switch(0)

Case 0: fork();

Case 1: fork(); fork();

Case 2: fork(); fork(); break; i = 2

i = 2; 2 < 5 true switch(2)

Case 2: fork(); fork(); break; i = 4

i = 4; 4 < 5 true switch(4)

Case 4: for j = 4 to 1 fork (); fork (); fork (); fork (); i = 6.

i = 5; 6 < 5 false

Total fork() calls made = 11

Total new processes = $2^{11} - 1 = 2047$

2. (31)

for n fork calls, number of child processes created = $2^n - 1$

$\therefore 2^5 - 1$

$= 32 - 1$

$= 31.$

3. (b)

- Context switching: Whenever process dispatch to running state by dispatcher, we need to perform context switching (loading PCB)
- Degree of multiprogramming: long term scheduler is responsible for creating new process to main memory so, it controls degree of multiprogramming.
- Message passing: Inter process communication (IPC) uses message passing method to communicate each other.
- Fork() : System call used to create new child process.

4. (a)

Through system call a process requests a service from the kernel of the operating system on which it is executing:

```
main ()
    {
        printf("Hello world");
    }
```

Here "printf()" is a system call used to request for monitor for printing "Hello world".

5. (c)

- If fork() calls "n" number of times then "2^n" number of process created.

  So, Total process created = $2^2 = 4$ processes
- Above program print 4 times

  "GATE2023".

6. (a)

Programmer can create and manage threads using API provided by thread library.

7. (a)

- Fork() is a system call used to create the child process.
- If fork() called "n" time then total number of child process will be "$2^n - 1$".
- Total "2^n"; 1 process is parent process and "$2^n - 1$" are child processes.
- Above program can be re-written as

```
void main()
{
    fork ();
    fork ();
    fork ();
    fork ();
}
```

- fork () called "4" times so, total child process will be "$2^4 - 1$" = 15.

8. (d)

S_1: Kernel – level threads have more context than user-level so, context switching in kernel – level threads is slower than user – level.

S_2: User – level threads are created using software so, operating system can not differentiate between user – level threads. So, blocking one user-level thread block entire process. True.

S_3: Operating system can easily differentiate kernel – level thread so, kernel – level threads can be scheduled independently. True.

9. (a, b, c, d)

There are five types of system calls – process control, file management
Device management, information management, communication.

10. (c)

System calls are executed in kernel mode but not made in kernel mode i.e. made in user mode.
Disabling interrupt is done in kernel mode.
So, option (c) is answer.