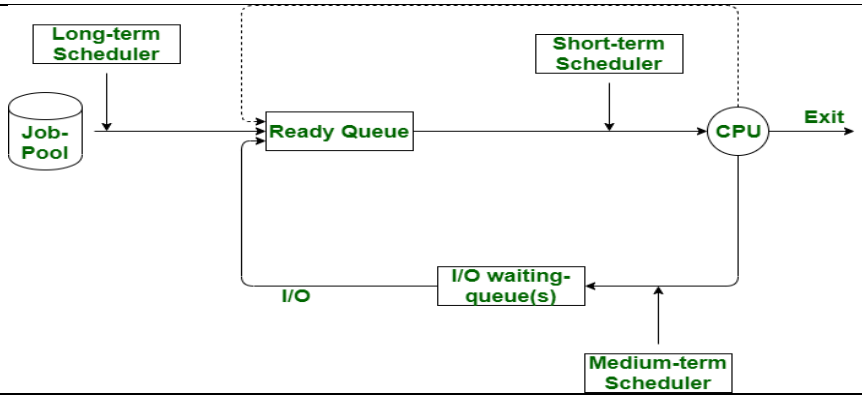
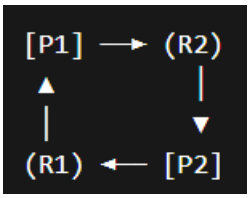
SIES Graduate School of Technology (Autonomous) NAAC A+ (Affiliated to University of Mumbai)		***End Semester Examination (R-24) FH 2026***		
Branch: CSE(IOT&CSIBCT)		Course: Operating System		
Year/ Semester: SE/IV		Course code: CSEC402		
Time: 03 hours		Marks: 80		
Note: 1. All questions are compulsory. **2. Figures to right indicate full marks.** **3. Assume suitable data wherever necessary.**		Marks	CO	BL
Q.1	**Attempt any FOUR. (All questions carry equal marks)**	**20**		
A.	**Compare Monolithic Kernel architecture and layered kernel architecture.** **Solution:** **Definition: 1 Mark** A Monolithic Kernel is an OS architecture where all core services such as process management, memory management, file system, and device drivers run in a single kernel space. The Linux uses a monolithic kernel. **Explanation: 2 Marks** Features of Monolithic Kernel: i All services execute in kernel mode ii High performance due to direct communication iii Large and complex structure Layered Kernel Architecture: i OS is divided into multiple layers ii Each layer performs a specific function iii Higher layers use services of lower layers **Difference: 2 Marks** i Monolithic is faster, layered is more modular ii Monolithic is less secure, layered provides better isolation iii Monolithic is harder to maintain, layered is easier to debug	05	CO1	L4
B.	**Explain long-term, medium-term, and short-term schedulers with neat diagram.** **Solution:** **Explanation: 3 Marks** Schedulers decide which process executes in the system. Types: i Long-term scheduler selects processes from job pool and loads into memory ii Medium-term scheduler suspends/resumes processes to control multiprogramming iii Short-term scheduler selects process from ready queue for CPU execution. **Diagram: 2 Marks**	05	CO2	L3

				
C.	**Explain the four necessary conditions for deadlock to occur. Give an example.** **Solution:** Deadlock occurs when processes are unable to proceed due to resource blocking. **Four Conditions: 4 Marks** i Mutual exclusion only one process can use resource at a time ii Hold and wait process holds resources while waiting for others iii No preemption resources cannot be forcibly taken iv Circular wait processes form a circular chain **Example: 1 Mark** Two processes P1 and P2 where P1 holds R1 and waits for R2, and P2 holds R2 and waits for R1  Explanation Process P1 holds R1 and is waiting for R2 Process P2 holds R2 and is waiting for R1 This creates a circular dependency, which satisfies one of the four necessary conditions of deadlock (Circular Wait) Since neither process can proceed, the system is in deadlock	05	CO3	L2
D.	**Explain the Copy-on-Write (COW) optimization. Describe how it is used in the fork system call.** **Solution:** **Definition (1 Mark)** Copy-on-Write (COW) is a memory optimization technique where multiple processes share the same memory pages and a copy is created only when modification is required. **Working (3 Marks)** When fork() is executed: The operating system creates a child process that is almost an exact copy of the parent process. Instead of immediately copying all memory, **copy-on-write** is used. Both parent and child initially share the same memory pages. i. These shared pages are marked as read-only.	05	CO4	L2

	ii. When either process modifies a page, a separate copy of that page is created for that process. iii. This reduces memory usage and improves performance, especially when the child process soon calls exec (). **Advantages (1 Mark)** Reduces memory usage Improves performance Faster process creation			
E.	**Explain in detail file access methods with advantages and disadvantages.** **Solution:** **Types (5 Marks)** Sequential Access a) Data accessed in order b) Operations read next write next Advantages i) Simple to implement ii) Efficient for sequential processing Disadvantages i) Slow for random access ii) Cannot jump directly Direct Access a) Records accessed using address b) Any location can be accessed Advantages i) Fast access ii) Suitable for databases Disadvantages i) Complex ii) Needs fixed size records Indexed Access a) Uses index to locate records b) Key maps to location Advantages i) Fast searching ii) Efficient for large data Disadvantages i) Extra storage needed ii) Maintenance overhead	05	CO5	L2
F.	**Explain the concept of granularity of protection.** **Solution:** **1) Definition (1 Mark)** Protection domain is a set of access rights assigned to a process It specifies which resources a process can access and what operations it can perform **2) Concept of Protection Domain (1.5 Marks)** i. Each process is executed within a domain ii. Domain contains objects and access rights iii. Objects include files, memory devices, printers	05	CO6	L2

iv. Access rights include read, write, execute, append
v. Ensure controlled and secure resource usage

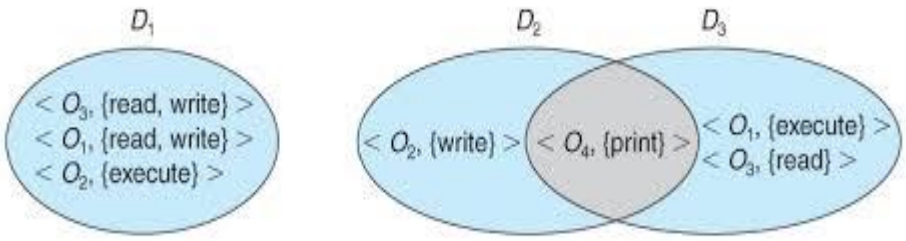
3) Structure of Protection Domain (1.5 Mark)

Domain is represented as set of object rights pairs

Example D = {(File1 read write) (File2 read) (Printer print)}

Defines permitted operations on each object

Diagram: 1 Mark



Q.2	**Attempt any FOUR. (All questions carry equal marks)**	**40**		
A.	**Construct the Gantt chart using Non-Preemptive Priority scheduling and First Come First Serve scheduling. Compute the average TAT and WT. Given four processes with Arrival Times (AT) and Burst Times (BT) as shown below**	10	CO2	L3

Process	**AT**	**BT**	**Priority (Lowest number higher priority)**
P2	0	10	1
P4	2	3	3
P3	3	6	2
P1	1	7	4

Solution:

Non-Preemptive Priority Scheduling (5 Marks)

Gantt Chart (2 Mark)

```
|    P2    |   P3   |  P4  |   P1   |
0          10       16     19       26
```

Calculations (2 Marks)

Averages (1 Mark)

Average Turnaround Time (TAT): 65 / 4 = 16.25

Average Waiting Time (WT): 39 / 4 = 9.75

First Come First Serve (FCFS): (5 Marks)

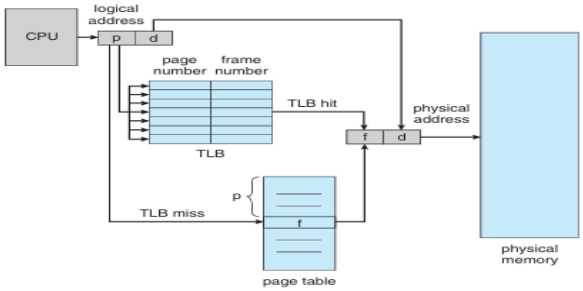
Gantt Chart (2 Mark)

```
|    P2    |   P1   |  P4  |   P3   |
0          10       17     20       26
```

	Calculations (2 Marks) Averages (1 Mark) • **Average Turnaround Time (TAT):** 67 / 4 = 16.75 • **Average Waiting Time (WT):** 41 / 4 = 10.25			
B.	**Apply the Banker's Algorithm on following data:** **A system has 4 processes (P0–P3) and 3 resource types (A, B, C). Resource type A has 12 instances, B has 8 instances, and C has 10 instances.** **Available Resources:** **A = 6, B = 3, C = 4** **Allocation Matrix** **Max Matrix** **Process A B C** **Process A B C** **P0 3 2 2** **P0 7 5 3** **P1 1 0 3** **P1 3 2 5** **P2 2 1 1** **P2 9 3 4** **P3 0 2 0** **P3 2 2 2** **Determine the system is in a safe state or not. If safe, provide the safe sequence of processes. If not safe, explain why deadlock may occur.** **Solution:** **1) Given Data (1 Mark)** **2) Calculate Need Matrix (2 Marks)** **Need = Max – Allocation** **Process A B C** P0 4 3 1 P1 2 2 2 P2 7 2 3 P3 2 0 2 **3) Apply Banker's Algorithm (Stepwise) (5 Marks)** Initial Available = (6, 3, 4) P1 can execute (Need ≤ Available) New Available = (6,3,4) + (1,0,3) = (7,3,7) P3 can execute New Available = (7,3,7) + (0,2,0) = (7,5,7) P0 can execute New Available = (7,5,7) + (3,2,2) = (10,7,9) P2 can execute New Available = (10,7,9) + (2,1,1) = (12,8,10) **4) Safe Sequence (1 Mark)** Safe Sequence = ⟨P1, P3, P0, P2⟩ **5) Conclusion (1 Mark)** All processes can complete successfully System is in safe state No deadlock will occur	10	CO3	L4
C.	**Design a synchronization algorithm using semaphores for the Producer–Consumer problem with a buffer of size N.**	10	CO3	L3

	Solution: **1) Problem Definition (1 Mark)** Producer–Consumer problem involves two processes: Producer produces items and adds them to buffer Consumer removes items from buffer Buffer has fixed size N **2) Requirements (1 Mark)** Producer should not add items when buffer is full Consumer should not remove items when buffer is empty Proper synchronization must be maintained **3) Semaphores Used (2 Marks)** mutex = 1 → ensures mutual exclusion empty = N → counts empty slots full = 0 → counts filled slots **4) Algorithm (Producer) (3 Marks)** wait(empty) wait(mutex) add item to buffer signal(mutex) signal(full) **5) Algorithm (Consumer) (3 Marks)** wait(full) wait(mutex) remove item from buffer signal(mutex) signal(empty) Conclusion Synchronization achieved using semaphores Ensures no overflow and no underflow Maintains data consistency and mutual exclusion			
D.	**Determine which block each process is assigned to and calculate the Total Internal Fragmentation for Best-Fit, First Fit, Worst Fit algorithms.** **The Scenario: A system has 5 memory blocks (frames) available in the following physical order:** **Block 1: 150 KB** **Block 2: 500 KB** **Block 3: 200 KB** **Block 4: 350 KB** **Block 5: 600 KB** **Four processes arrive in the sequence: P1 (212 KB), P2 (417 KB), P3 (112 KB), and P4 (326 KB).** **Solution:** **Given (1 Mark)** Memory Blocks: B1 = 150 KB, B2 = 500 KB, B3 = 200 KB, B4 = 350 KB, B5 = 600 KB Processes:	10	CO4	L3

	P1 = 212 KB, P2 = 417 KB, P3 = 112 KB, P4 = 326 KB **1. First Fit Allocation (3 Marks)** (Allocate first block that is large enough) Process Size Allocated Block Block Size Internal Fragmentation P1 212 B2 500 288 P2 417 B5 600 183 P3 112 B1 150 38 P4 326 B4 350 24 Total Internal Fragmentation = 288 + 183 + 38 + 24 = 533 KB **2. Best Fit Allocation (3 Marks)** (Allocate smallest block that fits) Process Size Allocated Block Block Size Internal Fragmentation P1 212 B4 350 138 P2 417 B2 500 83 P3 112 B1 150 38 P4 326 B5 600 274 Total Internal Fragmentation = 138 + 83 + 38 + 274 = 533 KB **3. Worst Fit Allocation (3 Marks)** (Allocate largest available block) Process Size Allocated Block Block Size Internal Fragmentation P1 212 B5 600 388 P2 417 B2 500 83 P3 112 B5 388 276 P4 326 B4 350 24 Total Internal Fragmentation = 388 + 83 + 276+24 = 771 KB **Conclusion (1 Mark)** First Fit and Best Fit both result in 533 KB fragmentation. Worst Fit gives the highest fragmentation (771 KB). Hence, First Fit / Best Fit are more efficient than Worst Fit in this case.			
E.	**Illustrate how the Translation Lookaside Buffer (TLB) improves efficiency with a neat diagram.** **Solution:** **1. Introduction (1 Mark)** 1 The Translation Lookaside Buffer (TLB) is a small, fast hardware cache used in paging systems. 2 It stores recently used page table entries for faster address translation. **2. Paging Without TLB (2 Marks)** 1 CPU generates a virtual address 2 Every memory reference needs two accesses 3 First access page table in main memory 4 Second access actual data in main memory 5 This increases memory access time and reduces performance **3. Paging With TLB (3 Marks)** 1 TLB stores page number to frame number mappings	10	CO4	L2

	2 CPU first checks TLB for required entry 3 If TLB hit then frame number is directly obtained 4 If TLB miss then page table is accessed in main memory 5 Miss result is updated into TLB for future use **4. Diagram (3 Marks)**  **5. Efficiency Improvement (1 Mark)** 1 Reduces number of memory accesses 2 Improves address translation speed 3 Increases CPU performance due to high hit ratio			
F.	**Define the following: Seek Time, Rotational Latency, Transfer Time. Calculate total disk capacity for following data:** **Suppose you have a hard disk with the following specifications:** **Platters: 5, Tracks per surface: 10000, Sectors per track: 500, Sector size: 512 bytes.** **Solution:** **1. Disk Access Time Components (3 Marks)** **1. Seek Time** 1 Time taken by the read/write head to move to the required track 2 Depends on distance between current and target track **2. Rotational Latency** 1 Time taken for the required sector to rotate under the read/write head 2 Depends on disk rotation speed **3. Transfer Time** 1 Time taken to actually read or write data once the head is positioned 2 Depends on data size and transfer rate **3. Disk Capacity Calculation (7 Marks)** Given Data: (1 M) 1 Platters = 5 2 Tracks per surface = 10,000 3 Sectors per track = 500 4 Sector size = 512 bytes Disk Capacity = Number of Platters × Surfaces per Platter × Tracks per Surface × Sectors per Track × Sector Size (2 M) Step 1: Surfaces (1 M) 1 Each platter has 2 surfaces 2 Total surfaces = 5 × 2 = 10 surfaces Step 2: Total Tracks (1 M) 1 Total tracks = 10 surfaces × 10,000 tracks = 100,000 tracks Step 3: Total Sectors (1 M)	10	CO5	L3

	1 Total sectors = 100,000 × 500 = 50,000,000 sectors Step 4: Total Capacity (1 M) 1 Capacity = 50,000,000 × 512 bytes = 25,600,000,000 bytes Final Answer: 1 Total Disk Capacity = 25.6 GB (approx.)			
Q.3	**Attempt any FOUR**	**20**		
A.	**Define Operating System. Discuss Objectives and functions of an OS** **Solution:** **Definition: 1 Mark** An Operating System (OS) is system software that acts as an interface between the user and computer hardware and manages all system resources efficiently. It is a core part of Operating Systems. **Objectives of OS: 2 Marks** i Efficient resource utilization ii Convenience to users (easy interface) iii Fair allocation of resources iv Improve system performance **Functions of OS: 2 Marks** i Process management (creation, scheduling, termination) ii Memory management (allocation, deallocation) iii File system management (files and directories) iv Device management (I/O control) v Security and protection	05	CO1	L2
B.	**Explain context switching with example.** **Solution:** Context Switching (5 Marks) Definition: Context switching is the process in which the operating system saves the state (context) of a currently running process and loads the state of another process so that execution can continue smoothly. Context includes: 1. Program Counter (PC) 2. CPU registers 3. Process state (ready, running, waiting) 4. Memory management information Need for Context Switching: 1. Enables multitasking in the system 2. Allows multiple processes to share the CPU 3. Handles interrupts and scheduling efficiently Example: Consider two processes P1 and P2. 1. P1 is currently executing on the CPU 2. The time slice expires or an interrupt occurs 3. The operating system saves the context of P1 4. The scheduler selects P2 from the ready queue	05	CO2	L2

	5. The operating system loads the context of P2 6. P2 starts executing from where it left off			
C.	**Explain the basic concepts of inter-process communication and synchronization. Discuss the necessity of synchronization in a multiprogramming environment.** **Solution:** **Inter-process Communication: 1.5 Marks** Inter-Process Communication (IPC) is a mechanism that allows processes to communicate and share data with each other. Basic Concepts of IPC: i Message passing (send/receive messages) ii Shared memory (common memory space) **Synchronization: 1.5 Marks** Synchronization ensures that multiple processes access shared resources in a controlled manner to avoid conflicts. **Why Synchronization is necessary: 2 Marks** i Prevents race conditions ii Maintains data consistency iii Avoids process interference iv Ensures correct execution order	05	CO3	L3
D.	**Explain paging hardware and give examples to convert Logical address to physical address.** **Solution:** **1. Definition (1 Mark)** Paging hardware is a memory management mechanism used by the operating system to map logical addresses generated by the CPU to physical addresses in main memory using a page table. **2. Components of Paging Hardware (2 Marks)** 1. Logical Address (LA) Divided into: o Page Number (p) o Offset (d) 2. Page Table o Stores the mapping of page numbers to frame numbers 3. Page Table Base Register (PTBR) o Holds the starting address of the page table 4. Frame Number (f) o Obtained from the page table 5. Physical Address (PA) o Formed by combining frame number and offset **3. Address Translation Process (1 Mark)** 1. CPU generates logical address (p, d) 2. Page number (p) is used to access the page table 3. Frame number (f) is obtained 4. Physical address is formed as: PA = (Frame Number × Page Size) + Offset **4. Example (1 Mark)**	05	CO4	L2
E.	**Compare file allocation methods with an example.**	05	CO5	L4

Solution:

File allocation methods (5 Marks)

Method	Description	Example	Advantage	Limitation
Contiguous	Continuous blocks	File in blocks 10–15	Fast access	External fragmentation
Linked	Blocks linked via pointers	5 → 9 → 13	No fragmentation	Slow access
Indexed	Index block stores pointers	Index → blocks	Direct access	Extra memory

F. **A system consists of four subjects (Users): U1, U2, U3, U4 and three objects (Files): F1, F2, F3. The access rights are Read (R), Write (W), and Execute (X). The access permissions are defined as follows:** 05 CO6 L3

U1 has R, W access on F1 and R access on F2
U2 has R access on F1 and W, X access on F3
U3 has R, X access on F2 and W access on F3
U4 has only Execute (X) access on F1 and F2
Answer the following questions for given scenario
a) Construct the Access Control Matrix for the given system.
b) Identify which users have write access to F3.
c) Convert the above matrix into an Access Control List (ACL) for object F1.

Solution:

a) Access Control Matrix (3 Marks)

User/Object	F1	F2	F3
U1	R, W	R	–
U2	R	–	W, X
U3	–	R, X	W
U4	X	X	–

b) Users with Write Access to F3 (1 Mark)

From the matrix:
U2 → W, X on F3
U3 → W on F3

c) Access Control List (ACL) for F1 (1 Mark)

ACL lists permissions object-wise.
F1 ACL:
U1 → R, W
U2 → R
U4 → X

*********** All the Best***********