# Scheme of Valuation/Answer Key

(Scheme of evaluation (marks in brackets) and answers of problems/key)

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**

B.Tech Degree S4 (R,S) / S2 (PT) (S,FE) / S4 (WP) (R) Examination May 2024 (2019 Scheme)

**Course Code: CST206**

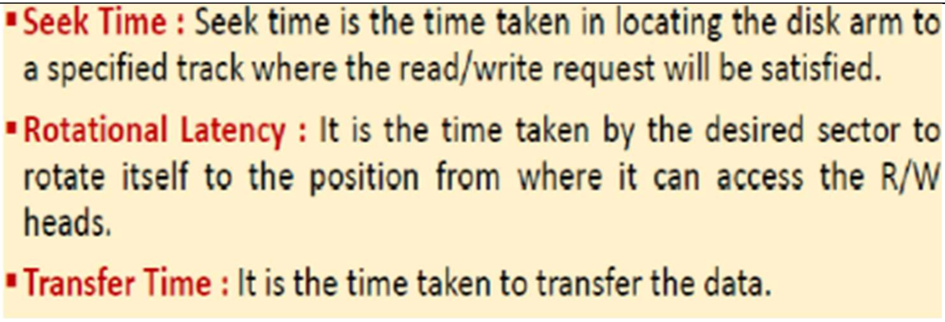
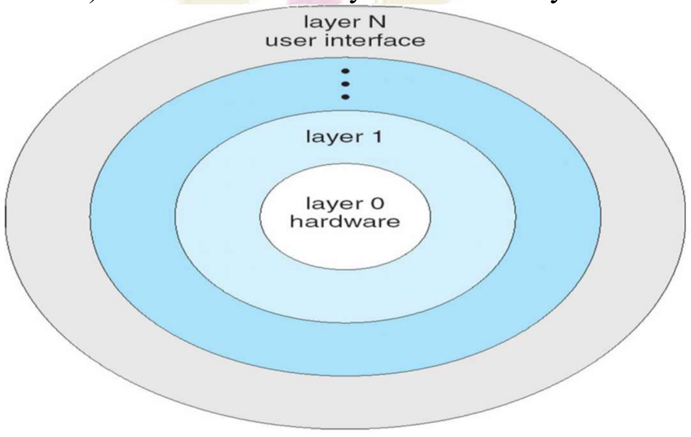
**Course Name: OPERATING SYSTEMS**

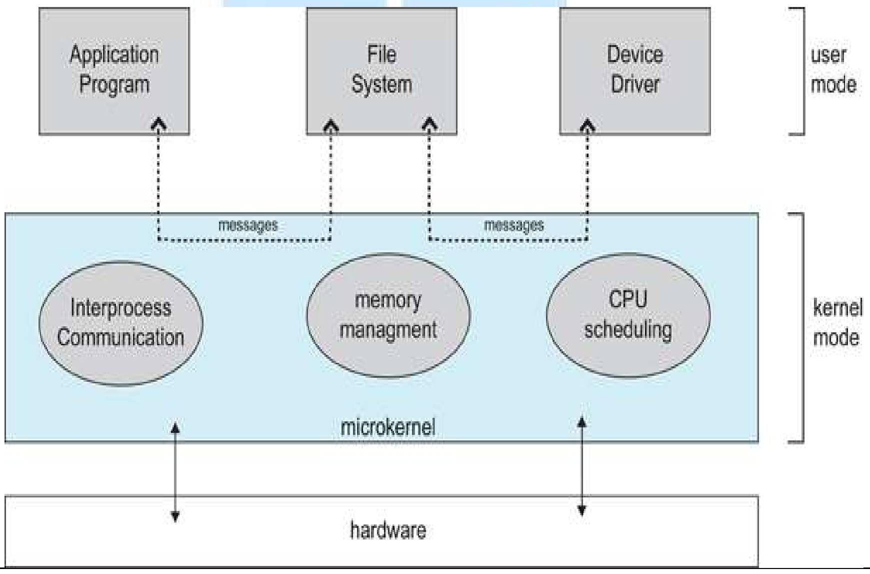
Max. Marks: 100 | Duration: 3 Hours

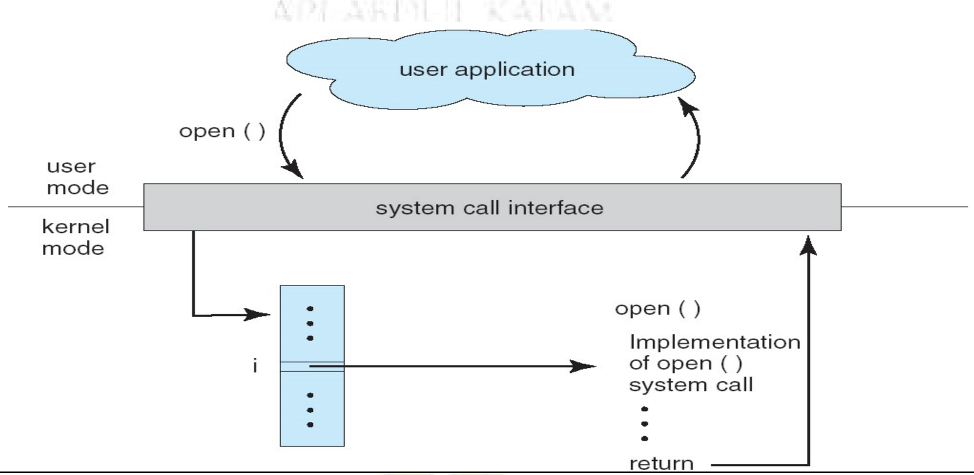
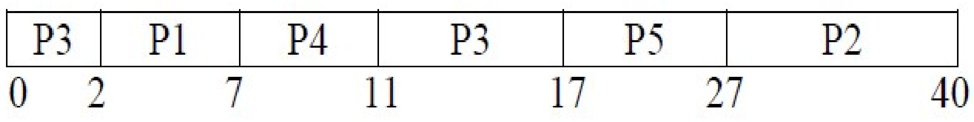
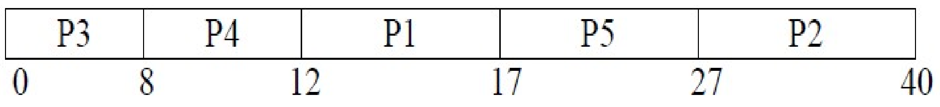
**PART A**

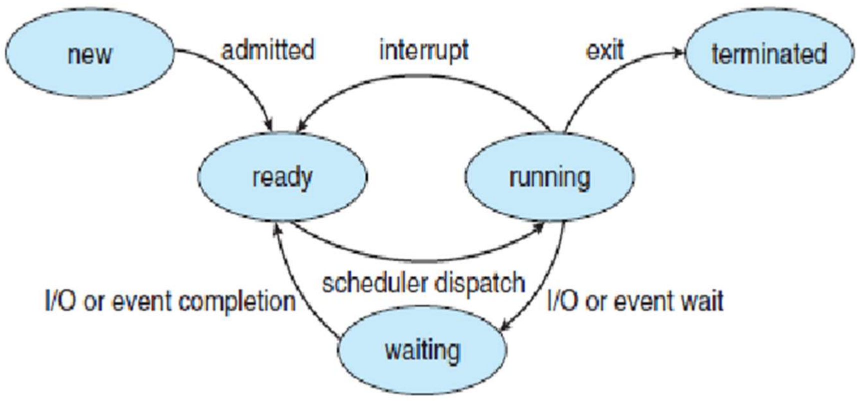
| | | *(Answer all questions; each question carries 3 marks)* | Marks |
|---|---|---|---|
| 1 | | **Symmetric multiprocessing (SMP) (1.5 marks)**<br>• Each processor performs all tasks within the operating system.<br>• All processors are peers; no boss–worker relationship exists between processors<br>**Asymmetric multiprocessing: (1.5 marks)**<br>• Each processor is assigned a specific task.<br>• A boss processor controls the system; the other processors either look to the boss for instruction or have predefined tasks.<br>• This scheme defines a boss–worker relationship.<br>• The boss processor schedules and allocates work to the worker processors.<br>. | 3 |
| 2 | | User mode (1.5 marks)<br>Kernel mode (1.5 marks)<br>Kernel model also called supervisor mode, system mode, or privileged mode). A bit, called the mode bit, is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1). With the mode bit, we can distinguish between a task that is executed on behalf of the operating system and one that is executed on behalf of the user. When the computer system is executing on behalf of a user application, the system is in user mode. However, when a user application requests a service from the operating system (via a system call), the system must transition from user to kernel mode to fulfil the request. | 3 |
| 3 | | Process control block- definition (1.5 marks)<br>Contents of PCB- (1.5 marks)<br>Each process is represented in the operating system by a process control block (PCB) also called a task control block.<br>Process state: The state may be new, ready, running, waiting, terminated, and so on.<br>• Program counter: The counter indicates the address of the next instruction to be executed for this process.<br>• CPU registers: The registers vary in number and type, depending on the computer architecture.<br>• They include accumulators, index registers, stack pointers, and general-purpose registers, any condition-code information.<br>• CPU-scheduling information: This information includes a process priority, pointers | 3 |

| | | | |
|---|---|---|---|
| | | to scheduling queues, and any other scheduling parameters.<br>• Memory-management information: This information may include such items as the<br>value of the base and limit registers and the page tables, or the segment tables, depending on the memory system used by the operating system.<br>• Accounting information: This information includes the amount of CPU and real time<br>used, time limits, job or process numbers, and so on.<br>• I/O status information: This information includes the list of I/O devices allocated to<br>the process, a list of open files, and so on. | |
| 4 | | 7 child processes and Good Luck printed 2 times and Do well printed 8 times. | 3 |
| 5 | | 3 requirements- Mutual exclusion, progress, bounded waiting 1 mark each | 3 |
| 6 | | Wait operation- 1.5 marks, signal operation- 1.5 marks | 3 |
| 7 | | External fragmentation- 1.5 marks, Internal fragmentation- 1.5 marks<br>External Fragmentation<br> As process are loaded and removed from memory, the free memory space is broken into little pieces. This is known as external fragmentation.<br> External fragmentation exists when there is enough total memory space to satisfy a request but the available spaces are not contiguous: storage is fragmented into a large number of small holes.<br> Both the first-fit and best-fit strategies for memory allocation suffer from external fragmentation.<br>Internal fragmentation<br> Unused memory that is internal to a partition.<br> Occurs when memory is partitioned on fixed size basis.<br>Consider multiple-partition allocation scheme with a hole of 18,464 bytes. Suppose that the next process requests 18,462 bytes. If we allocate exactly the requested block, we are left with a hole of 2 bytes. | 3 |
| 8 | | Memory management unit- Mapping from virtual address to physical address is done with Memory Management Unit (MMU)<br> Base register is also called relocation register | 3 |
| 9 | | Single directory structure<br>• The single-level directory is the simplest directory structure.<br>• All files are contained in the same directory<br>• Since all the files are in the same directory, they must have a unique name.<br>• Searching will become time taking if the directory is large.<br>• This can’t group the same type of files together. | 3 |

|  |  |  |  |
|---|---|---|---|
| 10 |  |  ▪Seek Time : Seek time is the time taken in locating the disk arm to a specified track where the read/write request will be satisfied.<br>▪Rotational Latency : It is the time taken by the desired sector to rotate itself to the position from where it can access the R/W heads.<br>▪Transfer Time : It is the time taken to transfer the data. | 3 |
| **PART B** |  |  |  |
| *(Answer one full question from each module, each question carries 14 marks)* |  |  |  |
| **Module -1** |  |  |  |
| 11 | a) | Any six functions- 1 mark each<br>• Process management<br>• Memory management<br>• File management<br>• I/O management<br>• Storage management<br>• Protection and security | 6 |
|  | b) | **Layered approach- 4 marks (Diagram+Explanation+Advantages and disadvantages)**<br>• The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.<br>• With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers<br> <br>• Less efficient<br>**Microkernel approach- 4 marks ((Diagram+Explanation+Advantages and disadvantages)**<br>• Moves as much from the kernel into user space | 8 |

| | | | |
|---|---|---|---|
| | | • Mach example of microkernel Mac OS X kernel (Darwin) partly based on Mach<br>• Communication takes place between user modules using message passing<br>• Benefits:<br>    • Easier to extend a microkernel<br>    • Easier to port the operating system to new architectures<br>    • More reliable (less code is running in kernel mode)<br>    • More secure<br> Application Program<br>File System<br>Device Driver<br>user mode<br>messages<br>messages<br>Interprocess Communication<br>memory managment<br>CPU scheduling<br>microkernel<br>kernel mode<br>hardware | |
| 12 | a) | Listing of 9 services and explanation -9 X 1mark =9 marks<br><br>1. User Interfaces<br>2. Program execution<br>3. I/O operations<br>4. File System manipulation<br>5. Communication<br>6. Error Detection<br>7. Resource Allocation<br>8. Accounting | 9 |

| | | | |
|---|---|---|---|
| | | 9. Protection and Security | |
| | b) | System call explanation – 2 marks<br>Explaining system calls using APIs and system call interface – 3 marks<br>• The mechanism used by an application program to request service from the operating system.<br>• System calls causes the processor to change mode (e.g. to "supervisor mode" or "protected mode").<br>• This allows the OS to perform restricted actions such as accessing hardware devices or the memory management unit.<br>• Typically,<br>    • a **number** associated with each system call<br>    • Number used as an index to a table: System Call **table**<br>    • Table keeps **addresses** of system calls (routines)<br>    • System call runs and returns<br>• Caller does not know system call implementation<br>    • Just knows interface<br> user application<br>open ( )<br>user mode<br>kernel mode<br>system call interface<br>i<br>open ( )<br>Implementation of open ( ) system call<br>return | 5 |
| **Module -2** | | | |
| 13 | a) | Gantt chart: SRTF – 2marks<br> <br>A.W.T. = ( 0+24+9+2+16)/5 = 51/5 = 10.2 – 1 mark<br>turnaround time. (5+37+17+6+26)/5 = 18.2 – 1 mark<br>Gantt chart: SJF – 2 marks<br> <br>A.W.T. = ( 10+24+0+3+16)/5 = 53/5 = 10.6 – 1 mark | 8 |

| | | | |
|---|---|---|---|
| | | turnaround time. (15+37+8+7+26)/5=18.6 – 1 mark | |
| | b) | Process state diagram- 2 marks<br> new<br>admitted<br>interrupt<br>exit<br>terminated<br>ready<br>running<br>I/O or event completion<br>scheduler dispatch<br>I/O or event wait<br>waiting<br>Process state description- 4 marks<br>1. Start/ New<br>This is the initial state when a process is first started/created.<br>2 Ready<br>The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run. Process may come into this state after Start state or while running it, but interrupted by the scheduler to assign CPU to some other process<br>3. Running<br>Once the process has been assigned to a processor by the OS scheduler, the process state is set to running and the processor executes its instructions.<br>4. Waiting<br>Process moves into the waiting state if it needs to wait for a resource, such as waiting for user input, or waiting for a file to become available.<br>5 Terminated or Exit<br>Once the process finishes its execution, or it is terminated by the operating system, it is moved to the terminated state. | 6 |
| 14 | a) | Shared memory- 4 marks<br>Message passing- 4 marks<br>• IPC using shared memory requires communicating processes to establish a region of shared memory.<br>• Processes can then exchange information by reading and writing data to the shared region.<br>• A shared-memory region resides in the address space of the process creating the shared- memory segment.<br>• Other processes that wish to communicate using this shared-memory segment must attach it to their address space.<br>• They can then exchange information by reading and writing data in the shared areas. | 8 |

| | | | |
|---|---|---|---|
| | | • The form of the data and the location are determined by these processes and are not under the operating system's control.<br>• The processes are also responsible for ensuring that they are not writing to the same location simultaneously.<br>Message passing model<br>Message passing provides a mechanism to allow processes to communicate and to<br>synchronize their actions without sharing the same address space.<br>• It is particularly useful in a distributed environment, where the communicating processes may reside on different computers connected by a network.<br>• Eg: Internet Chat Program<br>• A message-passing facility provides at least two operations:<br>1. send(message)<br>2. receive(message)<br>3. If processes P and Q wish to communicate, they need to:<br>1. Establish a communication link between them<br>2. Exchange messages via send/receive | |
| | b) | Short term scheduler- 2 marks<br>Long term scheduler- 2 marks<br>Medium term scheduler- 2 marks<br>The long-term scheduler, or job scheduler, selects processes from this pool and loads them into memory for execution.<br>The short-term scheduler, or CPU scheduler, selects from among the processes that are ready to execute and allocates the CPU to one of them.<br>Medium-term scheduler can remove a process from memory (and from active contention for the CPU) and later the process can be reintroduced into memory, and its execution can be continued where it left off.<br> This scheme is called swapping.<br> Swapping is necessary to improve the process mix (CPU bound I/O bound) | 6 |
| | | **Module -3** | |
| 15 | a) | i) A = 4+1+1+3 = 9, B = 1+2+6+2+2 = 13, C = 5+3+1+1 = 10,<br>D = 1+4+3+2+1 = 11 – 1 mark<br>ii) Need matrix : - 1 marks<br>    A B C D<br>P0 2 0 1 1<br>P1 1 6 5 0<br>P2 1 1 0 2<br>P3 1 0 2 0<br>P4 1 4 4 4<br>iii) Explanation of the algorithm with the steps – 4 Marks<br>YES , system is in safe state. – 1 mark<br>Safe Sequence: P0, P2, P3, P4, P1 – 1 mark | 8 |

| | | | |
|---|---|---|---|
| | b) | Any one two process solution with algorithm- petersons solution | 6 |
| 16 | a) | Reader's writers problem- 2 marks<br>Solution- reader process structure- 2 marks<br>Writer process structure- 2 marks | 8 |
| | b) | Recovery from deadlock<br>2 methods with sub-divisions and explanation 2 X 3 Marks<br>1-Resource pre-emption<br>2- Process termination | 6 |
| **Module -4** | | | |
| 17 | a) | FIFO- page faults- 6, Optimal- 5 page faults, LRU- 9 page faults | 9 |
| | b) | Segmentation concept- 1 mark, segment table- 2 marks. Address translation with diagram- 2 marks. | 5 |
| 18 | a) | (i) 3+10 = 13 bits<br>(ii) $2^{13}.2^{32} = 2^{45}$ bytes<br>(iii) int fragmentation –> 3K<br>(iv) No. 13 frames needed<br>2 marks each | 8 |
| | b) | Diagram – 2 Marks<br>Explanation with 6 steps involved in handling a page fault – 4 Marks | 6 |
| **Module -5** | | | |
| 19 | a) | Contiguous allocation- Explanation + Advantages and disadvantages- 5 marks<br>Linked file allocation- Explanation + Advantages and disadvantages -5 marks | 10 |
| | b) | Any four file attributes 1 mark each | 4 |
| 20 | a) | FCFS- 100->20->89->130->45->120->180<br>No: of head movements= 410 cylinders<br>SSTF- 100->89->120->130->180->45->20<br>Head movements- 262 cylinders<br>C-SCAN- 100->120->130->180->199->0->20->45->89<br>Head movements- 387 cylinders<br>OR<br>C-SCAN- 100->89->45->20->0->199->180->130->120<br>No: of head movements= **378 cylinders**<br>3 marks each | 9 |
| | b) | Sequential access- 2.5 marks<br>Direct access- 2.5 marks | 5 |
| | | ********* | |