Course code:20CA3104/2020

Gayatri vidya parishad college of Engineering(Autonomous)
Affiliated to JNTU-K, Kakinada
MCA I Semester Supplementary Examinations, December-2021
Operating Systems
(Schema & Key)

Date:30.12.21 **Max Marks:60**

UNIT-I(CO-1)

1. a) What is an Operating System? Explain functions of an Operating System.? 6M

An operating system (OS) is a program that acts as an interface between the system hardware and the user. Moreover, it handles all the interactions between the software and the hardware. All the working of a computer system depends on the OS at the base level. Further, it performs all the functions like handling memory, processes, the interaction between hardware and software, etc.

Functions of Operating System

- Memory Management
- Processor Management/Scheduling
- Device Management
- File Management
- Security
- Accounting

1. b) Give brief overview of different Computing Environments? 6M

Different types Computing Environments?

- Personal Computing Environment
- Time Sharing Computing Environment
- Client Server Computing Environment
- Distributed Computing Environment
- Cloud Computing Environment
- Cluster Computing Environment

2.a) Explain in brief Single Processor Systems and Multiprocessor Systems?

Single-Processor Systems **3M**

most computer systems used a single processor. On a singleprocessor system, there is one main CPU capable of executing a general-purpose instruction set, including instructions from user processes.

Multiprocessor Systems **3M**

Multiprocessor systems first appeared prominently appeared in servers and have since migrated to desktop and laptop systems. Recently, multiple processors have appeared on mobile devices such as smartphones and tablet computers. Multiprocessor systems have three main advantages

. Increased throughput

. Economy of scale.
. Increased reliability

2.b) Write short on Systems calls? 6M

System calls can be grouped roughly into six major categories:

- process control
- file manipulation
- device manipulation
- information maintenance
- communication
- protection

UNIT-II(CO-2)

3.a) Explain about Inter process Communication Briefly. 6M

Inter-process communication (IPC) is a mechanism that allows processes to communicate with each other and synchronize their actions.

Processes can communicate with each other through both:

i) Shared Memory Method

Ex:Producer-Consumerproblem

There are two processes: Producer and Consumer. The producer produces some items and the Consumer consumes that item. The two processes share a common space or memory location known as a buffer where the item produced by the Producer is stored and from which the Consumer consumes the item if needed.

ii) Messaging Passing Method

In this method, processes communicate with each other without using any kind of shared memory. If two processes p1 and p2 want to communicate with each other, they proceed as follows:

- Establish a communication link (if a link already exists, no need to establish it again.)
- Start exchanging messages using basic primitives.
  We need at least two primitives:
  – **send**(message, destination) or **send**(message)
  – **receive**(message, host) or **receive**(message)

TYPES:

- Message Passing through Communication Link:
  Direct Communication link

Indirect Communication link

- Message Passing through Exchanging the Messages:

Synchronous Message Passing

Asynchronous Message Passing

3.b) Present FCFS, SJF and Round Robin Process Scheduling Algorithms in detail?

First-Come, First-Served Scheduling: **3*2M=6M**

the process that requests the CPU first is allocated the CPU first. The implementation of the FCFS policy is easily managed with a FIFO queue. When a process enters the ready queue, its PCB is linked onto the tail of the queue. When the CPU is free, it is allocated to the process at the head of the queue. The running process is then removed from the queue. The code for FCFS scheduling is simple to write and understand.

Process	Burst Time -----
P1	24
p2	3
P3	3

If the processes ani ve in the order P1, P2, P3, and are served in FCFS order, we get the result shown in the following Gantt chart, which is a bar chart that illustrates a particular schedule, including the start and finish times of each of the participating processes: 0 24 27 30 The waiting time is 0 milliseconds for process P1, 24 milliseconds for process P2 , and 27 milliseconds for process P3 . Thus, the average waiting time is (0 + 24 + 27)/3 = 17 ncilliseconds. If the processes arrive in the order P2, P3 , P1, however, the results will be as shown in the following Gantt chart: 0 3 6 30 The average waiting time is now (6 + 0 + 3)/3 = 3 milliseconds. This reduction is substantial. Thus, the average waiting time under an FCFS policy is generally not minimal and d may vary substantially if the processes CPU burst times vary greatly.

SJF scheduling algorithm, schedules the processes according to their burst time.In SJF scheduling, the process with the lowest burst time, among the list of available processes in the ready queue, is going to be scheduled next.

Example

In the following example, there are five jobs named as P1, P2, P3, P4 and P5. Their arrival time and burst time are given in the table below.

PID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
1	1	7	8	7	0

2	3	3	13	10	7
3	6	2	10	4	2
4	7	10	31	24	14
5	9	8	21	12	4

Since, No Process arrives at time 0 hence; there will be an empty slot in the **Gantt chart** from time 0 to 1 (the time at which the first process arrives).

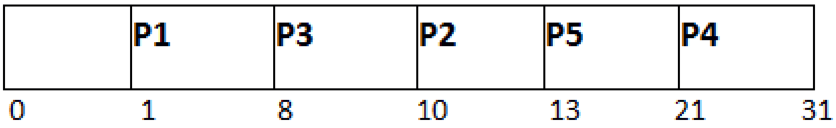
According to the algorithm, the OS schedules the process which is having the lowest burst time among the available processes in the ready queue.

Till now, we have only one process in the ready queue hence the scheduler will schedule this to the processor no matter what is its burst time.

This will be executed till 8 units of time. Till then we have three more processes arrived in the ready queue hence the scheduler will choose the process with the lowest burst time.

Among the processes given in the table, P3 will be executed next since it is having the lowest burst time among all the available processes.

So that's how the procedure will go on in **shortest job first (SJF)** scheduling algorithm.



Avg Waiting Time = 27/5

Round-robin scheduling:

In Round-robin scheduling, each ready task runs turn by turn only in a cyclic queue for a limited time slice. **Example** Consider this following three processes

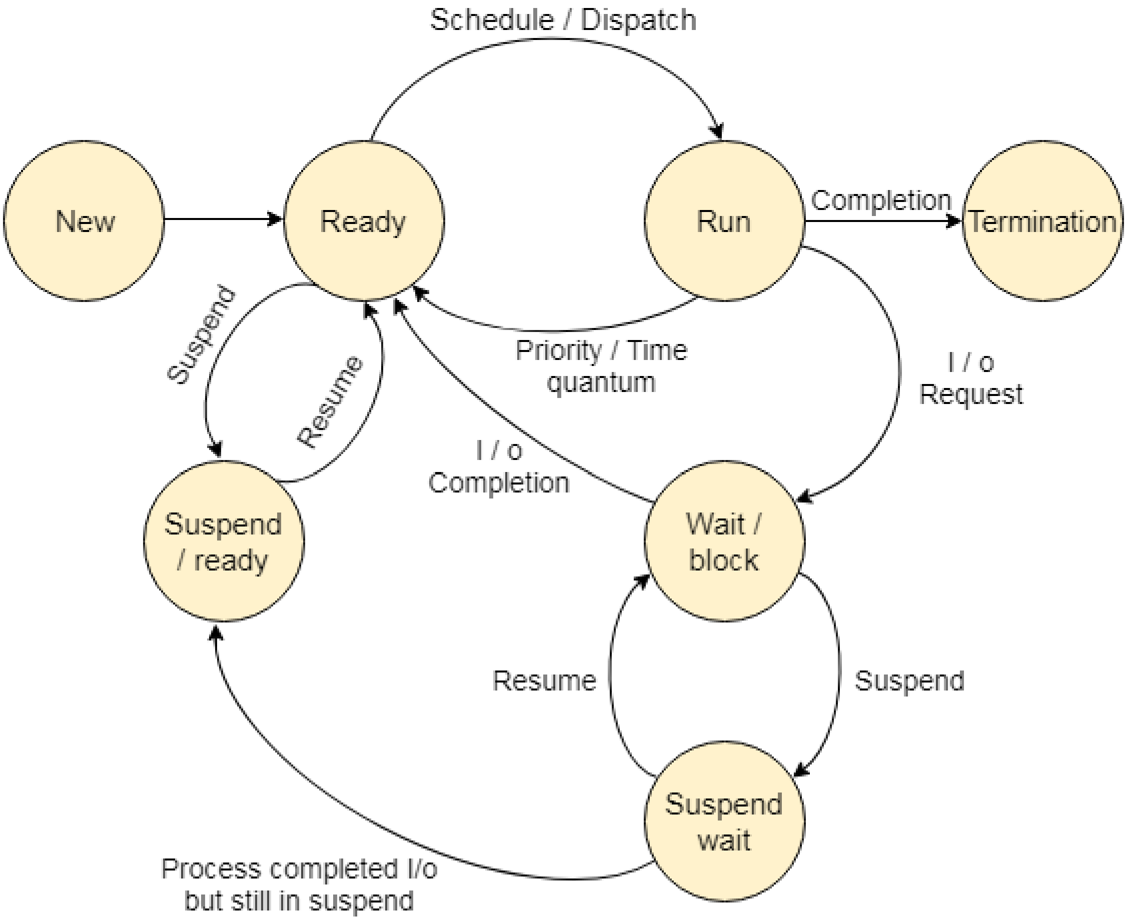
Process Queue	**Burst time**
P1	4
P2	3
P3	5

Step 1) The execution begins with process P1, which has burst time 4. Here, every process executes for 2 seconds. P2 and P3 are still in the waiting queue.
Step 2) At time =2, P1 is added to the end of the Queue and P2 starts executing
Step 3) At time=4 , P2 is pre-empted and add at the end of the queue. P3 starts executing.
Step 4) At time=6 , P3 is pre-empted and add at the end of the queue. P1 starts executing.
Step 5) At time=8 , P1 has a burst time of 4. It has completed execution. P2 starts execution
Step 6) P2 has a burst time of 3. It has already executed for 2 intervals. At time=9, P2 completes execution. Then, P3 starts execution till it completes.
Step 7) Let's calculate the average waiting time for above example.

Wait time
P1= 0+ 4= 4
P2= 2+4= 6
P3= 4+3= 7

4.a) Draw Process State Diagram and Explain? 6M

State Diagram



4.b) Write a Short note on Preemtive Scheduling and Explain Scheduling Criteria?

Preemptive Scheduling 3M

is a CPU scheduling technique that works by dividing time slots of CPU to a given process. The time slot given might be able to complete the whole process or might not be able to it. When the burst time of the process is greater than CPU cycle, it is placed back into the ready queue and will execute in the next chance. This scheduling is used when the process switch to ready state.

Algorithms that are backed by preemptive Scheduling are round-robin (RR), priority, SRTF (shortest remaining time first).

Scheduling Criteria: 3M

- Different CPU scheduling algorithms have different properties, and the choice of a particular algorithm may favour one class of processes over another.
- In choosing which algorithm to use in a particular situation, we must consider the properties of the various algorithms.
  - **CPU utilization**. We want to keep the CPU as busy as possible. Conceptually, CPU utilization can range from 0 to 100 percent. In a real system, it should range from 40 percent (for a lightly loaded system) to 90 percent (for a heavily used system).
  - **Throughput**. If the CPU is busy executing processes, then work is being done. One measure of work is the number of processes that are completed per time unit, called throughput. For long processes, this rate may be one process per hour; for short transactions, it may be 10 processes per second.
  - **Turnaround time**. From the point of view of a particular process, the important criterion is how long it takes to execute that process. The interval from the time of submission of a process to the time of completion is the turnaround time.
    - $T_r = T_s + T_w$
    - T_s :*Execution time.*
    - T_w :*Waiting time.*
  - **Waiting time**. The CPU scheduling algorithm does not affect the amount of time during which a process executes or does I/O; it affects only the amount of time that a process spends waiting in the ready queue.
  - **Response time**. In an interactive system, turnaround time may not be the best criterion. Often, a process can produce some output fairly early and can continue computing new results while previous results are being output to the user. Thus, another measure is the time from the submission of a request until the first response is produced.

UNIT-III(CO-3)

5.a) Review Dining –Philosphers Solution using monitors? 6M

Significance of this Problem

- Potential for deadlock and starvation
- Academic benchmark for evaluation and comparison of synchronization and mutual exclusion mechanisms
- An example for demonstrating various process and thread synchronization mechanisms
- A good solution has no deadlock or starvation

Review of Monitor Concept

- Encapsulated data objects and procedures (a.k.a. methods or functions)
- Per-monitor lock enforces mutual exclusion
- Only one thread may be executing in the monitor at a time
- Thread inside the monitor may reliquish the monitor lock to wait for a condition
- POSIX mutex and CVs designed to implement monitors

5.b) Analyze in detail Deadlock characterization? 6M

Deadlock characterization:

- **Mutual Exclusion:**The mutual-exclusion condition must hold for nonsharable resources.
- **Hold and Wait:**To ensure that the hold-and-wait condition never occurs in the system, we must guarantee that, whenever a process requests a resource, it does not hold any other resources
- **No Preemption**: The third necessary condition for deadlocks is that there be no preemption of resources that have already been allocated. To ensure that this condition does not hold, we can use the following protocol.
- **Circular Wait:**The fourth and final condition for deadlocks is the circular-wait condition. One way to ensure that this condition never holds is to impose a total ordering of all resource types and to require that each process requests resources in an increasing order of enumeration.

6.a) Discuss any two classic problems of Synchronization?

Bound-Buffer problem **3M**

Also known as the **Producer-Consumer problem**. In this problem, there is a buffer of n slots, and each buffer is capable of storing one unit of data. There are two processes that are operating on the buffer – Producer and Consumer. The producer tries to insert data and the consumer tries to remove data.

If the processes are run simultaneously they will not yield the expected output.

The solution to this problem is creating two semaphores, one full and the other empty to keep a track of the concurrent processes.

Dining Philosopher's problem **3M**

This problem states that there are K number of philosophers sitting around a circular table with one chopstick placed between each pair of philosophers. The philosopher will be able to eat if he can pick up two chopsticks that are adjacent to the philosopher.
This problem deals with the allocation of limited resources.

6.b) Explain the methods for handling Deadlock? 6M

- Deadlock Ignorance
- Deadlock prevention
- Deadlock avoidance
- Deadlock detection and recovery

UNIT-IV(CO-4)

7.a) Explain the concept of Segmentation? 6M

A process is divided into Segments. The chunks that a program is divided into which are not necessarily all of the same sizes are called segments.
There are types of segmentation:

1. **Virtualmemorysegmentation** –
   Each process is divided into a number of segments, not all of which are resident at any one point in time.
2. **Simplesegmentation** –
   Each process is divided into a number of segments, all of which are loaded into memory at run time, though not necessarily contiguously.

7.b) Present any two file allocation methods in disk space in detail?

Contiguous Allocation **3M**

If the blocks are allocated to the file in such a way that all the logical blocks of the file get the contiguous physical block in the hard disk then such allocation scheme is known as contiguous allocation.

Advantages

- It is simple to implement.
- We will get Excellent read performance.
- Supports Random Access into files.

Disadvantages

- he disk will become fragmented.
- It may be difficult to have a file grow.

Linked List Allocation **3M**

Linked List allocation solves all problems of contiguous allocation. In linked list allocation, each file is considered as the linked list of disk blocks. However, the disks blocks allocated to a particular file need not to be contiguous on the disk. Each disk block allocated to a file contains a pointer which points to the next disk block allocated to the same file.

Advantages

- There is no external fragmentation with linked allocation.
- Any free block can be utilized in order to satisfy the file block requests.
- File can continue to grow as long as the free blocks are available.
- Directory entry will only contain the starting block address.

Disadvantages

- Random Access is not provided.
- Pointers require some space in the disk blocks.
- Any of the pointers in the linked list must not be broken otherwise the file will get corrupted.
- Need to traverse each block.

8.a) Explain Different File access Methods 6M

File Access Methods

Sequential Access

Most of the operating systems access the file sequentially. In other words, we can say that most of the files need to be accessed sequentially by the operating system.In sequential access, the OS read the file word by word. A pointer is maintained which initially points to the base address of the file. If the user wants to read first word of the file then the pointer provides that word to the user and increases its value by 1 word. This process continues till the end of the file.

Modern word systems do provide the concept of direct access and indexed access but the most used method is sequential access due to the fact that most of the files such as text files, audio files, video files, etc need to be sequentially accessed.

Direct Access

The Direct Access is mostly required in the case of database systems. In most of the cases, we need filtered information from the database. The sequential access can be very slow and inefficient in such cases.

Suppose every block of the storage stores 4 records and we know that the record we needed is stored in 10th block. In that case, the sequential access will not be implemented because it will traverse all the blocks in order to access the needed record.

Direct access will give the required result despite of the fact that the operating system has to perform some complex tasks such as determining the desired block number. However, that is generally implemented in database applications.

Indexed Access If a file can be sorted on any of the filed then an index can be assigned to a group of certain records. However, A particular record can be accessed by its index. The index is nothing but the address of a record in the file.

In index accessing, searching in a large database became very quick and easy but we need to have some extra space in the memory to store the index value.

8.b) What is Demand Paging?Explain in brief. 6M

According to the concept of Virtual Memory, in order to execute some process, only a part of the process needs to be present in the main memory which means that only a few pages will only be present in the main memory at any time.

However, deciding, which pages need to be kept in the main memory and which need to be kept in the secondary memory, is going to be difficult because we cannot say in advance that a process will require a particular page at particular time.

Therefore, to overcome this problem, there is a concept called Demand Paging is introduced. It suggests keeping all pages of the frames in the secondary memory until they are required. In other words, it says that do not load any page in the main memory until it is required.

UNIT-V(CO-5)

9.a) Summarize the concept of Swap-Space Management? 6M

Swap space is a space on a hard disk that is a substitute for physical memory. It is used as virtual memory, which contains process memory images. Whenever our computer runs short of physical memory, it uses its virtual memory and stores information in memory on a disk. This interchange of data between virtual memory and real memory is called ***swapping*** and space on disk as swap space. Swap space helps the computer's operating system pretend that it has more RAM than it actually has. It is also called a ***swap file***.
Virtual memory is a combination of RAM and disk space that running processes can use. **Swap space** is the **portion of virtual memory** on the hard disk, used when RAM is full. Swap space can be useful to computers in the following various ways, such as:

- It can be used as a single contiguous memory which reduces I/O operations to read or write a file.
- Applications that are not used or less used can be kept in a swap file.
- Having sufficient swap files helps the system keep some physical memory free all the time.

- The space in physical memory that has been freed due to swap space can be used by OS for other important tasks.

9.b) What is Access Matrix? Explain in detail. 6M

Access matrix in Operating System. **Access Matrix** is a security model of protection state in computer system. It is represented as a **matrix**. **Access matrix** is used to define the rights of each process executing in the domain with respect to each object. The rows of **matrix** represent domains and columns represent objects.

10. Suppose that a disk drive has 5000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order is 86,1470,913,1774,948,1509,1022,1750,130.
Starting from the current head position, What is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests, for each of the following disk scheduling algorithms and Explain.? 12M

a.The FCFS schedule is 143, 86, 1470, 913, 1774, 948, 1509, 1022,1750, 130. The total seek distance is 7081.

b. The SSTF schedule is 143, 130, 86, 913, 948, 1022, 1470, 1509, 1750,1774. The total seek distance is 1745.

c. The SCAN schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774,4999, 130, 86. The total seek distance is 9769.

d. The LOOK schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774,130, 86. The total seek distance is 3319.

e. The C-SCAN schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774,4999, 86, 130. The total seek distance is 9813.

f.The C-LOOK schedule is 143, 913, 948, 1022, 1470, 1509,1750, 1774, 86, 130. The total seek distance is 3363