**Intro and Process and Threads:**

1. Write the difference between Process and Threads.

Process is a unit for resource allocation. Thread is a unit for execution.

Process has at least one thread or more. Thread is a part of process.

Process has code/data/heap stack and register. Thread does not have data

and heap.

Threads within a process share code/data/heap, share I/O, but each has its own stack & registers.

If a process dies, its resources are reclaimed, and all threads die. If a thread dies, its stack is reclaimed.

Inter-process communication via OS and data copying. Inter-thread communication via memory.

Each process can run on a different physical processor. Each thread can run on a different physical processor.

1. Show the Secondary Storage hierarchy according to its cost and speed.

1. Register

2. Cache

3. Primary Memory (RAM)

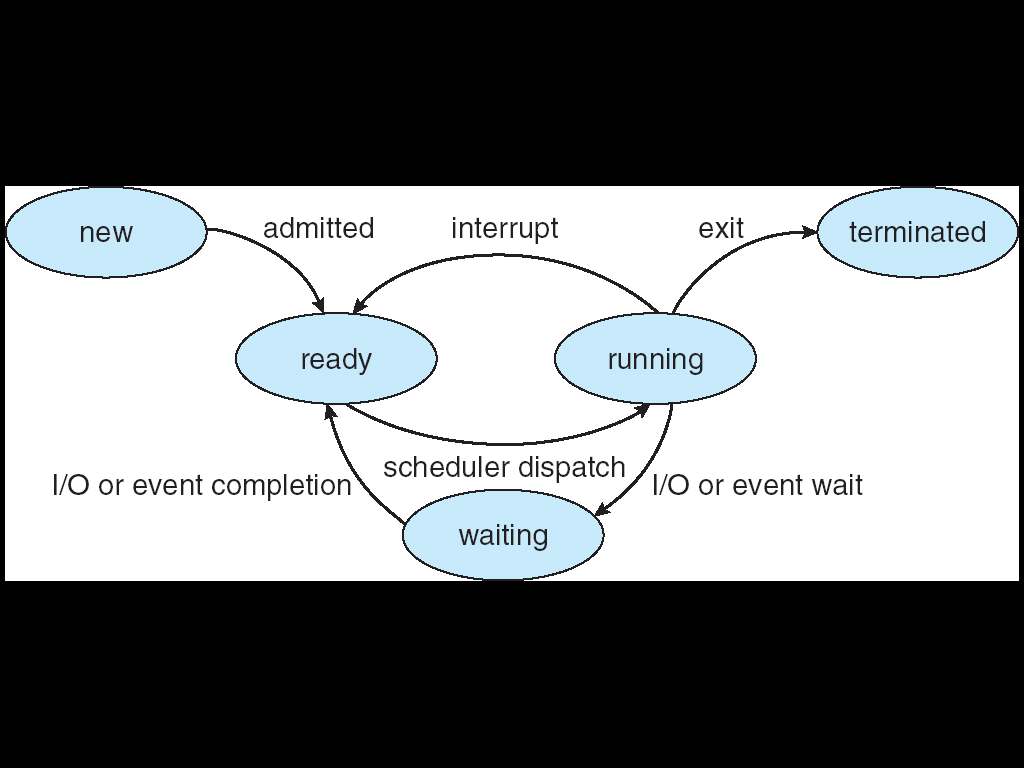
4. Electronic Disk

5. Magnetic Disk

6. Optical Disk

7. Magnetic Tape

1. Write the names of these states shown in figure 1



1. Write the basic comparisons between Linux and Windows OS.

Linux | Windows

1. Cost Free | Not free

2. Security More secure | Vulnerable to worms, attacks

3. Kernel Monolithic kernel | Mikrokernel

4. Open Source Yes | No

5. Custumizable Yes | No

6. Case sensitive Yes | Not case sensitive

7. Directory structure Backslash (\) | Forward slash (/)

8. Filesystems EXT2, EXT3, EXT4, | FAT, FAT32, NTFS, REFS

9. Booting primary and logical | Primary

10. Performance Good | Lower than Linux

**Distributed OS:**

1. Write down full form of these terms: WAN, DNS, ARP, DFS & FTP

WAN: Wide Area Network

DNS: Domain Name System

ARP: Address Resolution Protocol

DFS: Distributed File System

FTP: File Transfer Protocol

1. Write the difference between Circuit switching and Packet Switching.

Circuit Switching Packet Switching

Concept - Telephone system looks for Packets sent as soon as arrive

physical path

Terminal Telephone, modem computer

Information representation Analog or digital voice Any binary information

Transmission System Analog or digital over various media Digital over various media

Addressing Hierarchical numbering plan Hierarchical address space

Routing Route selected during call Each packet send independently

Multiplexing Circuit multiplexing Packet multiplexing shared   
 media across network

Packets arrive inn order Yes No

1. Show the ISO Network Model with its layers and describe each layer.

1. Application Layer: Deals with the user interface, logging etc.

2. Presentation Layer: Deals with the formats of the files.

3. Session Layer: Deals with creating and terminating sessions

4. Transport Layer: Used for transporting the data

5. Network Layer: Used for creating the network connection

6. DataLink Layer: Used for dealing with packages and fragments

7. Physical Layer: Used for converting the data to bits

**RTOS and Deadlock:**

1. Define RTOS, Preemption, Reentrancy, Concurrency and Deadlock.

A **real-time operating system** (RTOS) is any operating system (OS) which not only produces correct results but also the results should be within specific deadline period.

The act of taking control of the operating system from one task and giving it to another task is called preempting.

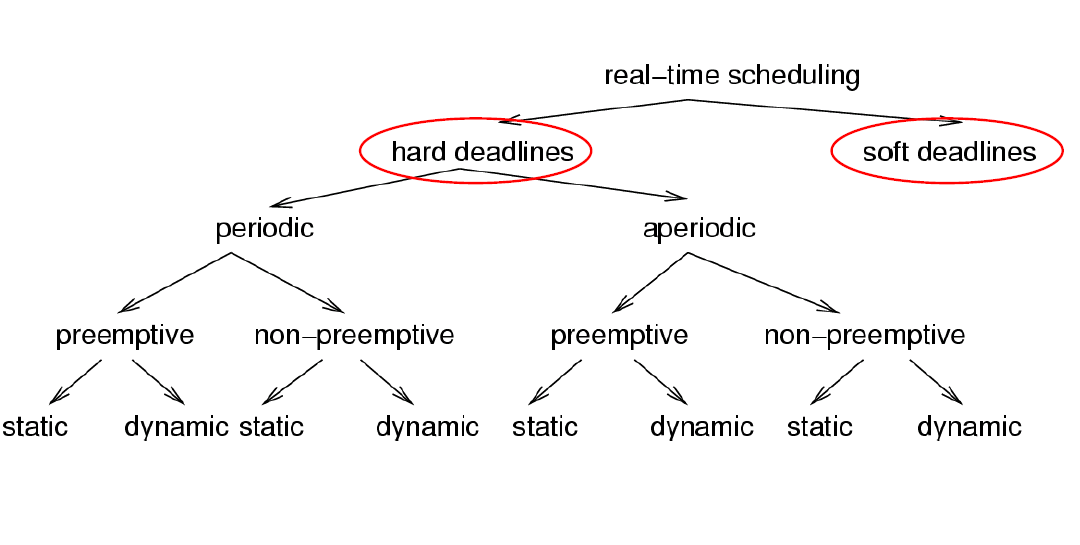
**preemption** is the act of temporarily interrupting a [task](https://en.wikipedia.org/wiki/Task_(computing)) being carried out by a [computer system](https://en.wikipedia.org/wiki/Computer), without requiring its cooperation, and with the intention of resuming the task at a later time.

a computer program or subroutine is called **reentrant** if it can be interrupted in the middle of its execution and then safely be called again ("re-entered") before its previous invocation's complete execution.

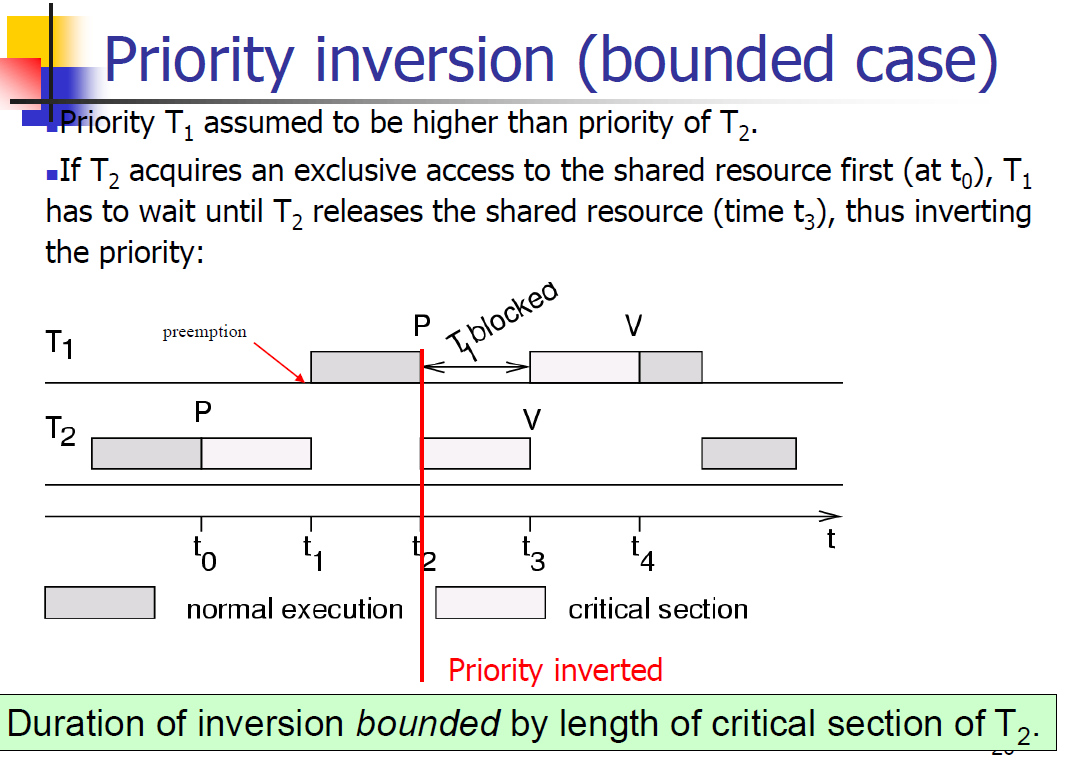
**Concurrency** means multiple computations are happening at the same time.  **concurrency** is the ability of different parts or units of a program, algorithm, or problem to be executed out-of-order or in partial order, without affecting the final outcome.

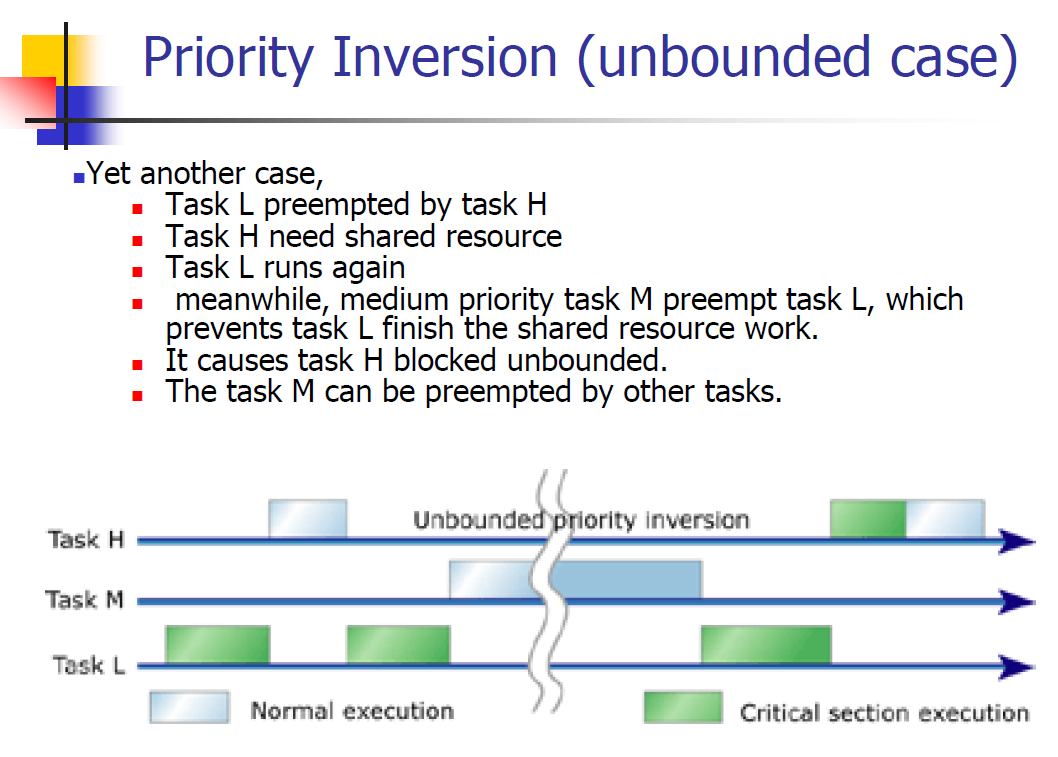
a **deadlock** is a state in which each member of a group is waiting for another member, including itself, to take action, such as sending a message or more commonly releasing a [lock](https://en.wikipedia.org/wiki/Lock_(computer_science)).

1. Classify Schedule algorithm.



1. Priority inversion Bounded vs Unbounded with example.

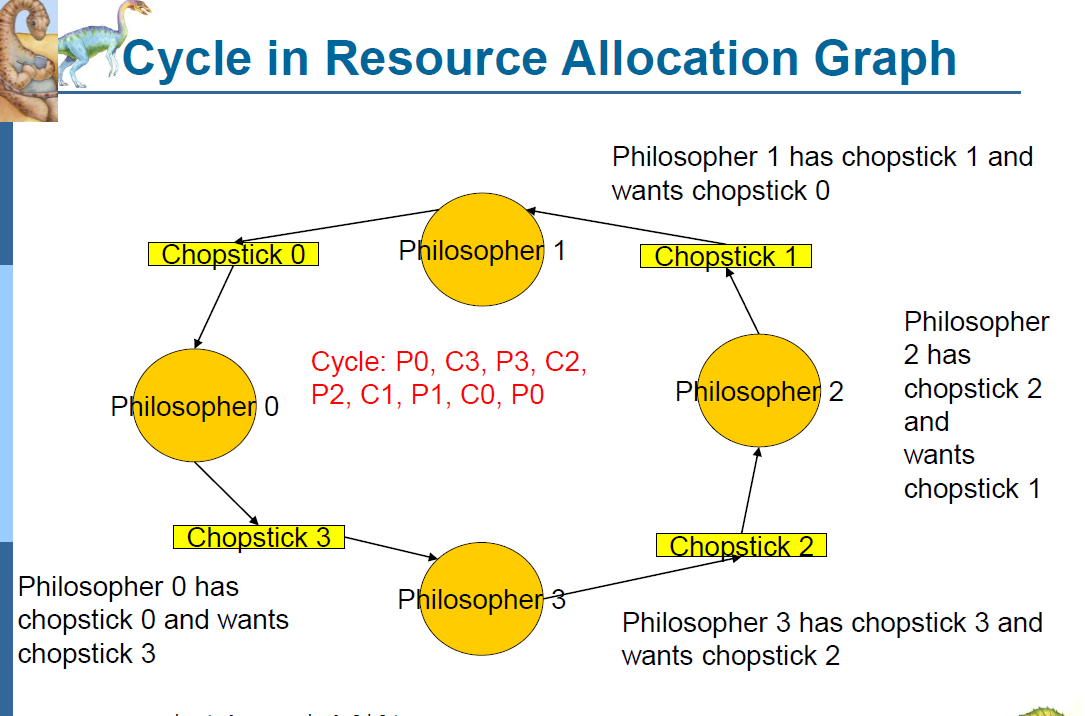




1. Difference between Priority Celling vs Priority Inheritance protocol.

***priority inheritance:***when a job blocks one or more high priority jobs, it ignores its original priority assignment and executes its critical section at the highest priority level of all the jobs it blocks.  
***priority ceiling:*** gives each shared resource a predefined priority ceiling. When a task acquires a shared resource, the task is hoisted (has its priority temporarily raised) to the priority ceiling of that resource. It will not see whether the job has been blocked or not, simply it raises to the priority of the shared resource.

1. Draw the cycle of resource allocation graph for Dining philosopher.



1. Difference between Deadlock Prevention, Avoidance and Detection.

Deadlock avoidance: means, whenever a request is made for a particular resource by a particular process, you look at the available resources, if the future resource needs for process's already in use resources, determine the possibility of a deadlock in case the resource is granted. If possible, don't grant the resource, if not possible then grant the resource

Deadlock prevention: make sure that at least one of the condition for deadlock to occur is not fulfilled at anytime. This can be achieved in the way resources are requested and granted in the system

1) Prevention: structure the system in such a way that one of the deadlock conditions is negated

2) Detection and recovery: let deadlocks occur, detect them and take action

3) Avoidance: -don’t start processes whose requests may cause a deadlock -don’t grant requests which may cause a deadlock

**Deadlock:**

A deadlock is a situation where two or more competing actions are each waiting for the other to finish, and thus neither ever does. It can also be defined as a set of blocked processes each holding a resource and waiting to acquire a resource held by another process in the set.

For example if there is a system with two disk drives. If there are two processes P1 and P2 each hold one disk drive and each needs the other one, then the situation of deadlock occurs. The following conditions will be held simultaneously in case of deadlock:

• **Mutual exclusion**: only one process at a time can use a resource

• **Hold and wait:** a process holding at least one resource is waiting to acquire additional resources held by other processes

• **No preemption**: a resource can be released only voluntarily by the process holding it, after that process has completed its task

• **Circular wait**: there exists a set {P0, P1, …, Pn} of waiting processes such that P0 is waiting for a resource that is held by P1, P1 is waiting for a resource that is held by P2, …, Pn–1 is waiting for a resource that is held by Pn, and Pn is waiting for a resource that is held by P0.

Differences between Deadlock prevention, avoidance and detection are as follows: