Operating System

LEARN IN 1 DAY

KRISHNA RUNGTA

Learn Operating System in 1 Day

By Krishna Rungta

Table Of Content

Chapter 1: What is Operating System? Explain Types of OS, Features and Examples

1. What is an Operating System?
2. History Of OS
3. Examples of Operating System with Market Share
4. Types of Operating System (OS)
5. Functions of Operating System
6. Features of Operating System (OS)
7. Advantage of using Operating System
8. Disadvantages of using Operating System
9. What is Kernel in Operating System?
10. Features of Kennel
11. Difference between Firmware and Operating System
12. Difference between 32-Bit vs. 64 Bit Operating System

Chapter 2: What is Semaphore? Binary, Counting Types with Example

1. What is Semaphore?
2. Characteristic of Semaphore
3. Types of Semaphores
4. Example of Semaphore
5. Wait and Signal Operations in Semaphores
6. Counting Semaphore vs. Binary Semaphore
7. Difference between Semaphore vs. Mutex
8. Advantages of Semaphores
9. Disadvantage of semaphores

Chapter 3: Components of Operating Systems

1. What are OS Components?
2. File Management
3. Process Management
4. I/O Device Management
5. Network Management
6. Main Memory management
7. Secondary-Storage Management
8. Security Management
9. Other Important Activities

Chapter 4: Microkernel in Operating System: Architecture, Advantages

1. What is Kernel?
2. What is Microkernel?
3. What is a Monolithic Kernel?
4. Microkernel Architecture
5. Components of Microkernel
6. Difference Between Microkernel and Monolithic Kernel
7. Advantages of Microkernel
8. Disadvantage of Microkernel

Chapter 5: System Call in OS (Operating System): What is, Types and Examples

1. What is System Call in Operating System?
2. Example of System Call
3. How System Call Works?
4. Why do you need System Calls in OS?
5. Types of System calls
6. Rules for passing Parameters for System Call
7. Important System Calls Used in OS

Chapter 6: File Systems in Operating System: Structure, Attributes, Type

1. What is File System?
2. Objective of File management System
3. Properties of a File System
4. File structure
5. File Attributes
6. File Type
7. Functions of File
8. Commonly used terms in File systems
9. File Access Methods
10. Space Allocation
11. File Directories
12. File types- name, extension

Chapter 7: Real-time operating system (RTOS): Components, Types, Examples

1. What is a Real-Time Operating System (RTOS)?
2. Why use an RTOS?
3. Components of RTOS
4. Types of RTOS
5. Terms used in RTOS
6. Features of RTOS
7. Factors for selecting an RTOS
8. Difference between in GPOS and RTOS
9. Applications of Real Time Operating System
10. Disadvantages of RTOS

Chapter 8: Remote Procedure Call (RPC) Protocol in Distributed System

1. What is RPC?
2. Types of RPC
3. RPC Architecture
4. How RPC Works?
5. Characteristics of RPC
6. Features of RPC
7. Advantages of RPC
8. Disadvantages of RPC

Chapter 9: CPU Scheduling Algorithms in Operating Systems

1. What is CPU Scheduling?
2. Types of CPU Scheduling
3. Important CPU scheduling Terminologies
4. CPU Scheduling Criteria
5. Interval Timer
6. What is Dispatcher?
7. Types of CPU scheduling Algorithm
8. First Come First Serve
9. Shortest Remaining Time
10. Priority Based Scheduling
11. Round-Robin Scheduling
12. Shortest Job First
13. Multiple-Level Queues Scheduling
14. The Purpose of a Scheduling algorithm

Chapter 10: Process Management in Operating System: PCB in OS

1. What is a Process?
2. What is Process Management?
3. Process Architecture

4. Process Control Blocks
5. Process States
6. Process Control Block(PCB)

Chapter 11: Introduction to DEADLOCK in Operating System

1. What is Deadlock?
2. Example of Deadlock
3. What is Circular wait?
4. Deadlock Detection
5. Deadlock Prevention:
6. Deadlock Avoidance
7. Difference Between Starvation and Deadlock
8. Advantages of Deadlock
9. Disadvantages of Deadlock method

Chapter 12: FCFS Scheduling Algorithm: What is, Example Program

1. What is First Come First Serve Method?
2. Characteristics of FCFS method
3. Example of FCFS scheduling
4. How FCFS Works? Calculating Average Waiting Time
5. Advantages of FCFS
6. Disadvantages of FCFS

Chapter 13: Paging in Operating System(OS)

1. What is Paging?
2. Example
3. What is Paging Protection?

4. Advantages of Paging
5. Disadvantages of Paging
6. What is Segmentation?
7. Advantages of a Segmentation method
8. Disadvantages of Segmentation

Chapter 14: Livelock: What is, Example, Difference with Deadlock

1. What is Livelock?
2. Examples of Livelock
3. What Leads to Livelock?
4. What is Deadlock?
5. Example of Deadlock
6. What is Starvation?
7. Difference Between Deadlock, Starvation, and Livelock

Chapter 15: Inter Process Communication (IPC)

1. What is Inter Process Communication?
2. Approaches for Inter-Process Communication
3. Why IPC?
4. Terms Used in IPC
5. What is Like FIFOS and Unlike FIFOS

Chapter 16: Round Robin Scheduling Algorithm with Example

1. What is Round-Robin Scheduling?
2. Characteristics of Round-Robin Scheduling
3. Example of Round-robin Scheduling
4. Advantage of Round-robin Scheduling

5. Disadvantages of Round-robin Scheduling
6. Worst Case Latency

Chapter 17: Process Synchronization: Critical Section Problem in OS

1. What is Process Synchronization?
2. How Process Synchronization Works?
3. Sections of a Program
4. What is Critical Section Problem?
5. Rules for Critical Section
6. Solutions To The Critical Section

Chapter 18: Process Scheduling: Long, Medium, Short Term Scheduler

1. What is Process Scheduling?
2. Process Scheduling Queues
3. Two State Process Model
4. Scheduling Objectives
5. Type of Process Schedulers
6. Long Term Scheduler
7. Medium Term Scheduler
8. Short Term Scheduler
9. Difference between Schedulers
10. What is Context switch?

Chapter 19: Priority Scheduling Algorithm: Preemptive, Non-Preemptive EXAMPLE

1. What is Priority Scheduling?
2. Types of Priority Scheduling
3. Characteristics of Priority Scheduling

4. Example of Priority Scheduling
5. Advantages of priority scheduling
6. Disadvantages of priority scheduling

Chapter 20: Memory Management in OS: Contiguous, Swapping, Fragmentation

1. What is Memory Management?
2. Why Use Memory Management?
3. Memory Management Techniques
4. What is Swapping?
5. What is Memory allocation?
6. Partition Allocation
7. What is Paging?
8. What is Fragmentation?
9. What is Segmentation?
10. What is Dynamic Loading?
11. What is Dynamic Linking?
12. Difference Between Static and Dynamic Loading
13. Difference Between Static and Dynamic Linking

Chapter 21: Shortest Job First (SJF): Preemptive, Non-Preemptive Example

1. What is Shortest Job First Scheduling?
2. Characteristics of SJF Scheduling
3. Non-Preemptive SJF
4. Preemptive SJF
5. Advantages of SJF
6. Disadvantages/Cons of SJF

Chapter 22: Virtual Memory in OS: What is, Demand Paging, Advantages

1. What is Virtual Memory?
2. Why Need Virtual Memory?
3. How Virtual Memory Works?
4. What is Demand Paging?
5. Types of Page Replacement Methods
6. FIFO Page Replacement
7. Optimal Algorithm
8. LRU Page Replacement
9. Advantages of Virtual Memory
10. Disadvantages of Virtual Memory

Chapter 23: Banker's Algorithm in Operating System [Example]

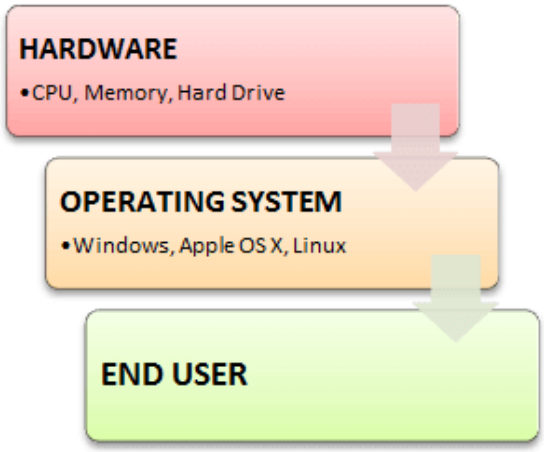
1. What is Banker's Algorithm?
2. Banker's Algorithm Notations
3. Example of Banker's algorithm
4. Characteristics of Banker's Algorithm
5. Disadvantage of Banker's algorithm

Chapter 1: What is Operating System? Explain Types of OS, Features and Examples

What is an Operating System?

An **Operating System (OS)** is a software that acts as an interface between computer hardware components and the user. Every computer system must have at least one operating system to run other programs. Applications like Browsers, MS Office, Notepad Games, etc., need some environment to run and perform its tasks.

The OS helps you to communicate with the computer without knowing how to speak the computer's language. It is not possible for the user to use any computer or mobile device without having an operating system.



Introduction to Operating System

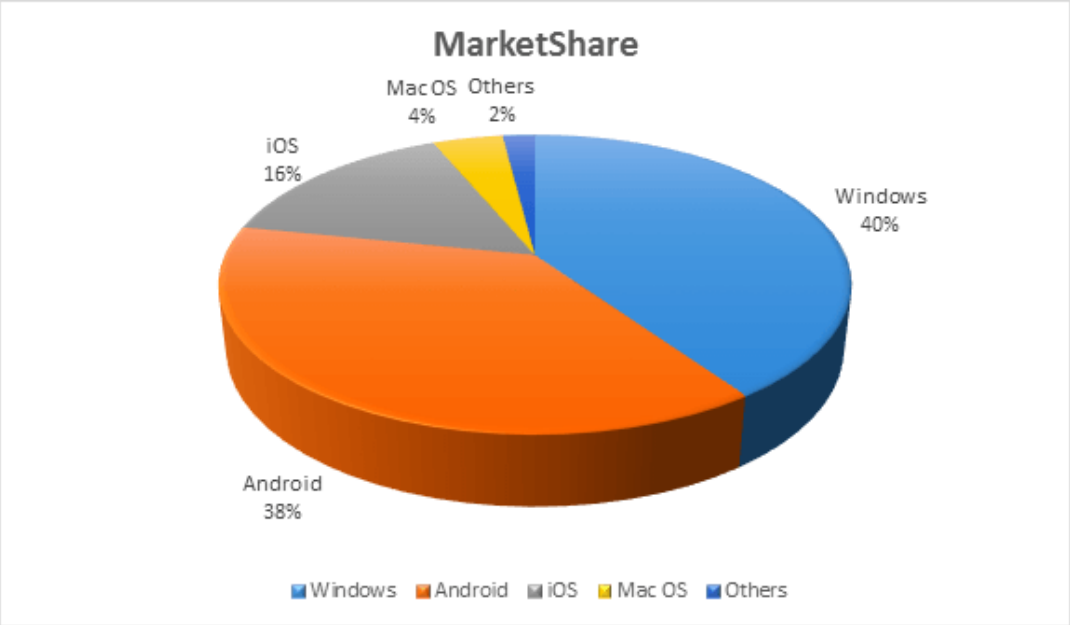
In this OS tutorial, you will learn:

- What is an Operating System?
- History Of OS
- Examples of Operating System with Market Share
- Types of Operating System (OS)
- Functions of Operating System
- Features of Operating System (OS)
- Advantage of using Operating System
- Disadvantages of using Operating System
- What is Kernel in Operating System?
- Difference between Firmware and Operating System
- Difference between 32-Bit vs. 64 Bit Operating System

History Of OS

- Operating systems were first developed in the late 1950s to manage tape storage
- The General Motors Research Lab implemented the first OS in the early 1950s for their IBM 701
- In the mid-1960s, operating systems started to use disks
- In the late 1960s, the first version of the Unix OS was developed
- The first OS built by Microsoft was DOS. It was built in 1981 by purchasing the 86-DOS software from a Seattle company
- The present-day popular OS Windows first came to existence in 1985 when a GUI was created and paired with MS-DOS.

Examples of Operating System with Market Share



Market Share of Operating Systems

Following are the Operating System examples with the latest Market Share

OS Name	Share
Windows	40.34
Android	37.95
iOS	15.44
Mac OS	4.34
Linux	0.95
Chrome OS	0.14
Windows Phone OS	0.06

Types of Operating System (OS)

Following are the popular types of OS (Operating System):

- Batch Operating System
- Multitasking/Time Sharing OS
- Multiprocessing OS
- Real Time OS
- Distributed OS
- Network OS

- Mobile OS

Batch Operating System

Some computer processes are very lengthy and time-consuming. To speed the same process, a job with a similar type of needs are batched together and run as a group. The user of a batch operating system never directly interacts with the computer. In this type of OS, every user prepares his or her job on an offline device like a punch card and submit it to the computer operator.

Multi-Tasking/Time-sharing Operating systems

Time-sharing operating system enables people located at a different terminal(shell) to use a single computer system at the same time. The processor time (CPU) which is shared among multiple users is termed as time sharing.

Real time OS

A real time operating system time interval to process and respond to inputs is very small. Examples: Military Software Systems, Space Software Systems are the Real time OS example.

Distributed Operating System

Distributed systems use many processors located in different machines to provide very fast computation to its users.

Network Operating System

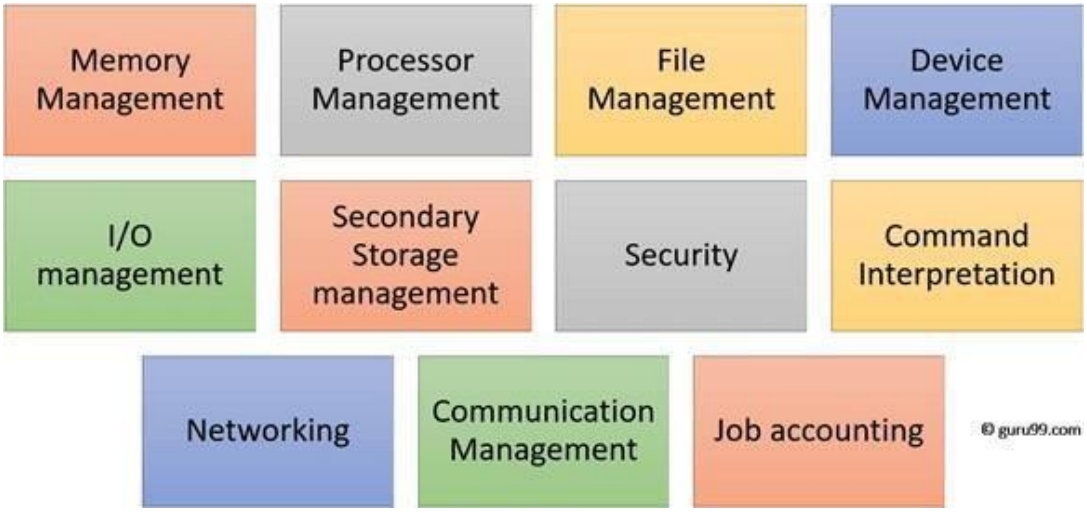
Network Operating System runs on a server. It provides the capability to serve to manage data, user, groups, security, application, and other networking functions.

Mobile OS

Mobile operating systems are those OS which is especially that are designed to power smartphones, tablets, and wearables devices. Some most famous mobile operating systems are Android and iOS, but others include BlackBerry, Web, and watchOS.

Functions of Operating System

Some typical operating system functions may include managing memory, files, processes, I/O system & devices, security, etc. Below are the main functions of Operating System:



Functions of Operating System

In an operating system software performs each of the function:

1. **Process management**:- Process management helps OS to create and delete processes. It also provides mechanisms for synchronization and communication among processes.

2. **Memory management:-** Memory management module performs the task of allocation and de-allocation of memory space to programs in need of this resources.

3. **File management**:- It manages all the file-related activities such as organization storage, retrieval, naming, sharing, and protection of files.

4. **Device Management**: Device management keeps tracks of all devices. This module also responsible for this task is known as the I/O controller. It also performs the task of allocation and de-allocation of the devices.

5. **I/O System Management:** One of the main objects of any OS is to hide the peculiarities of that hardware devices from the user.

6. **Secondary-Storage Management**: Systems have several levels of storage which includes primary storage, secondary storage, and cache storage. Instructions and data must be stored in primary storage or cache so that a running program can reference it.

7. **Security**:- Security module protects the data and information of a computer system against malware threat and authorized access.

8. **Command interpretation**: This module is interpreting commands given by the and acting system resources to process that commands.

9. **Networking:** A distributed system is a group of processors which do not share memory, hardware

devices, or a clock. The processors communicate with one another through the network.

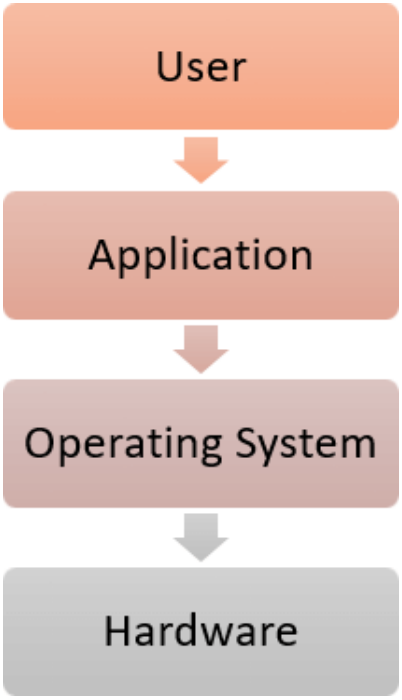
10. **Job accounting**: Keeping track of time & resource used by various job and users.

11. **Communication management**: Coordination and assignment of compilers, interpreters, and another software resource of the various users of the computer systems.

Features of Operating System (OS)

Here is a list important features of OS:

- Protected and supervisor mode
- Allows disk access and file systems Device drivers Networking Security
- Program Execution
- Memory management Virtual Memory Multitasking
- Handling I/O operations
- Manipulation of the file system
- Error Detection and handling
- Resource allocation
- Information and Resource Protection



Advantage of using Operating System

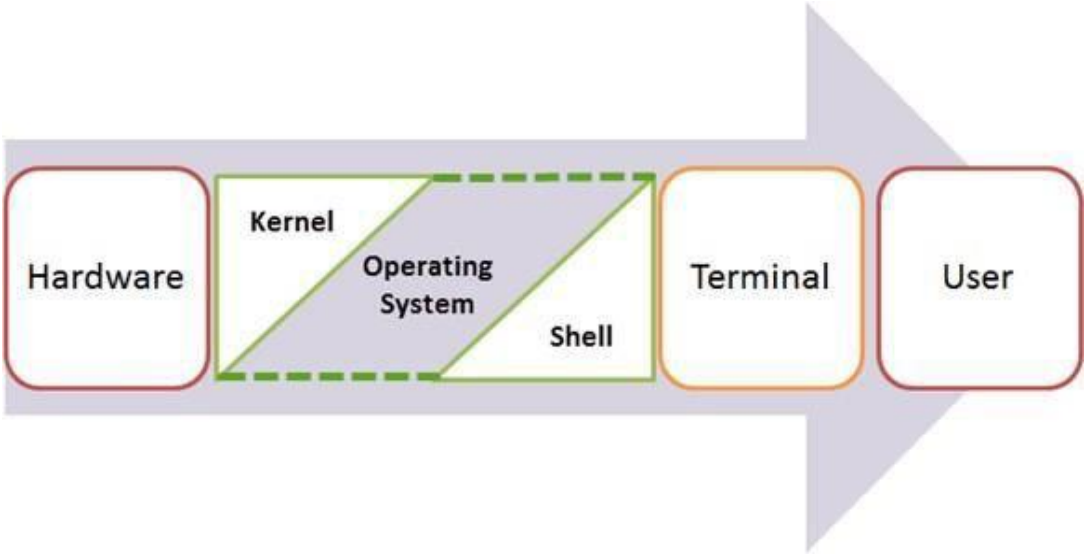
- Allows you to hide details of hardware by creating an abstraction
- Easy to use with a GUI
- Offers an environment in which a user may execute programs/applications
- The operating system must make sure that the computer system convenient to use
- Operating System acts as an intermediary among applications and the hardware components
- It provides the computer system resources with easy to use format
- Acts as an intermediator between all hardware's and software's of the system

Disadvantages of using Operating System

- If any issue occurs in OS, you may lose all the contents which have been stored in your system
- Operating system's software is quite expensive for small size organization which adds burden on them. Example Windows
- It is never entirely secure as a threat can occur at any time

What is Kernel in Operating System?

The kernel is the central component of a computer operating systems. The only job performed by the kernel is to the manage the communication between the software and the hardware. A Kernel is at the nucleus of a computer. It makes the communication between the hardware and software possible. While the Kernel is the innermost part of an operating system, a shell is the outermost one.



Introduction to Kernel

Features of Kennel

- Low-level scheduling of processes
- Inter-process communication

- Process synchronization
- Context switching

Types of Kernels

There are many types of kernels that exists, but among them, the two most popular kernels are:

1. **Monolithic** A monolithic kernel is a single code or block of the program. It provides all the required services offered by the operating system. It is a simplistic design which creates a distinct communication layer between the hardware and software.

2. **Microkernels** Microkernel manages all system resources. In this type of kernel, services are implemented in different address space. The user services are stored in user address space, and kernel services are stored under kernel address space. So, it helps to reduce the size of both the kernel and operating system.

Difference between Firmware and Operating System

Firmware	Operating System
Define Firmware: Firmware is one kind of programming that is embedded on a chip in the device which controls that specific device.	Define Operating System: OS provides functionality over and above that which is provided by the firmware.
Firmware is programs that been encoded by the manufacture of the IC or something and cannot be changed.	OS is a program that can be installed by the user and can be changed.
It is stored on non-volatile memory.	OS is stored on the hard drive.

Difference between 32-Bit vs. 64 Bit Operating System

Parameters	32. Bit	64. Bit
Architecture and Software	Allow 32 bit of data processing simultaneously	Allow 64 bit of data processing simultaneously
Compatibility	32-bit applications require 32-bit OS and CPUs.	64-bit applications require a 64-bit OS and CPU.
Systems Available	All versions of Windows 8, Windows 7, Windows Vista, and Windows XP, Linux, etc.	Windows XP Professional, Vista, 7, Mac OS X and Linux.
Memory Limits	32-bit systems are limited to 3.2 GB of RAM.	64-bit systems allow a maximum 17 Billion GB of RAM.

Summary

- What is OS (Operating System definition) and its Types: An operating system is a software which acts as an interface between the end user and computer hardware. Different categories of Operating System in computer and other devices are: Batch Operating System, Multitasking/Time Sharing OS, Multiprocessing OS, Real Time OS, Distributed OS, Network OS & Mobile OS
- Personal Computer Operating Systems were first developed in the late 1950s to manage tape storage
- Explain Operating System working: OS works as an intermediate between the user and computer. It helps the user to communicate with the computer without knowing how to speak the computer's language.
- The kernel is the central component of a computer operating systems. The only job performed by the kernel is to the manage the communication between the software and the hardware
- Two most popular kernels are Monolithic and MicroKernels
- Process, Device, File, I/O, Secondary-Storage, Memory management are various functions of an Operating System

Chapter 2: What is Semaphore? Binary, Counting Types with Example

What is Semaphore?

Semaphore is simply a variable that is non-negative and shared between threads. A semaphore is a signaling mechanism, and a thread that is waiting on a semaphore can be signaled by another thread. It uses two atomic operations, 1)wait, and 2) signal for the process synchronization. A semaphore either allows or disallows access to the resource, which depends on how it is set up.

In this Operating System(OS) tutorial, you will learn:

- Characteristic of Semaphore
- What is Semaphore?
- Types of Semaphores
- Example of Semaphore
- Wait and Signal Operations in Semaphores
- Counting Semaphore vs. Binary Semaphore
- Difference between Semaphore vs. Mutex
- Advantages of Semaphores
- Disadvantage of semaphores

Characteristic of Semaphore

Here, are characteristic of a semaphore:

- It is a mechanism that can be used to provide synchronization of tasks.
- It is a low-level synchronization mechanism.
- Semaphore will always hold a non-negative integer value.
- Semaphore can be implemented using test operations and interrupts, which should be executed using file descriptors.

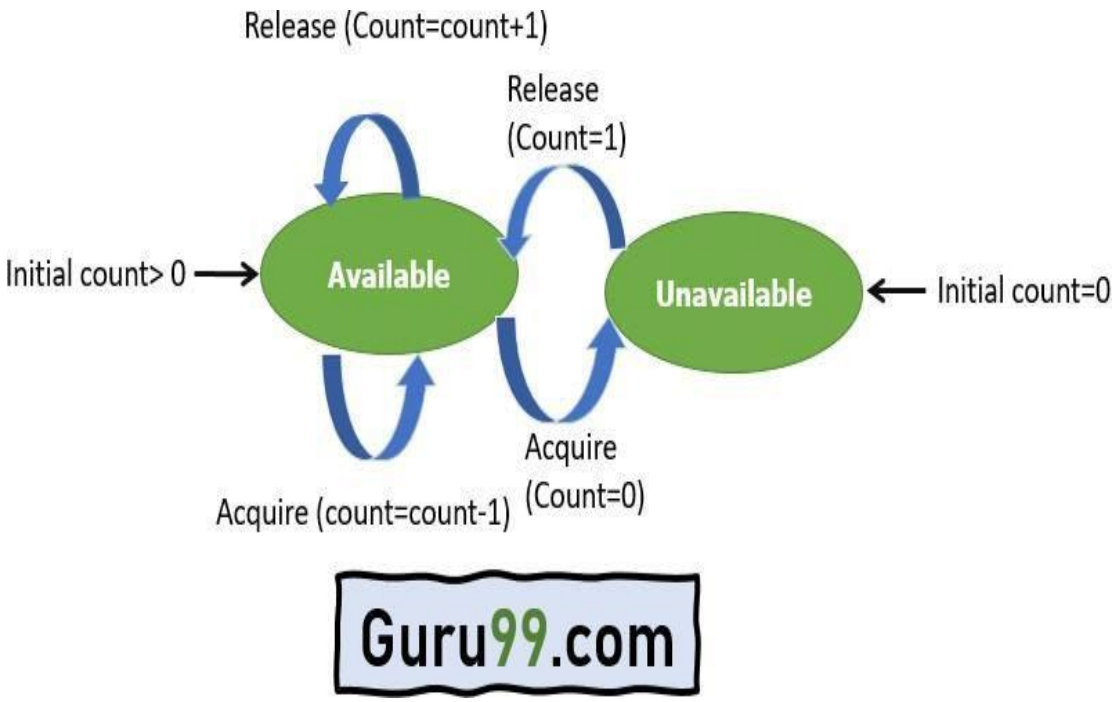
Types of Semaphores

The two common kinds of semaphores are

- Counting semaphores
- Binary semaphores.

Counting Semaphores

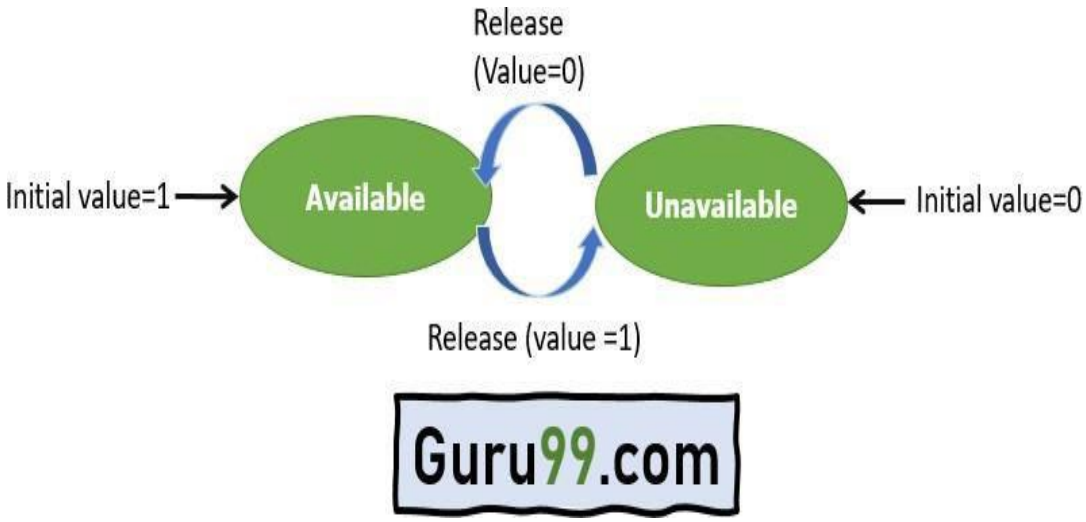
This type of Semaphore uses a count that helps task to be acquired or released numerous times. If the initial count = 0, the counting semaphore should be created in the unavailable state.



However, If the count is > 0, the semaphore is created in the available state, and the number of tokens it has equals to its count.

Binary Semaphores

The binary semaphores are quite similar to counting semaphores, but their value is restricted to 0 and 1. In this type of semaphore, the wait operation works only if semaphore = 1, and the signal operation succeeds when semaphore= 0. It is easy to implement than counting semaphores.



Example of Semaphore

The below-given program is a step by step implementation, which involves usage and declaration of semaphore.

```
Shared var mutex: semaphore = 1;
Process i
    begin
    .
    .
    P(mutex);
    execute CS;
    V(mutex);
    .
    .
    End;
```

Wait and Signal Operations in Semaphores

Both of these operations are used to implement process synchronization. The goal of this semaphore operation is to get mutual exclusion.

Wait for Operation

This type of semaphore operation helps you to control the entry of a task into the critical section. However, If the value of wait is positive, then the value of the wait argument X is decremented. In the case of negative or zero value, no operation is executed. It is also called P(S) operation. After the semaphore value is decreased, which becomes negative, the command is held up until the required conditions are satisfied.

```
Copy CodeP(S)
{
    while (S<=0);
    S--;
}
```

Signal operation

This type of Semaphore operation is used to control the exit of a task from a critical section. It helps to increase the value of the argument by 1, which is denoted as V(S).

```
Copy CodeP(S)
{
    while (S>=0);
    S++;
}
```

Counting Semaphore vs. Binary Semaphore

Here, are some major differences between counting and binary semaphore:

Counting Semaphore	Binary Semaphore
No mutual exclusion	Mutual exclusion
Any integer value	Value only 0 and 1
More than one slot	Only one slot
Provide a set of Processes	It has a mutual exclusion mechanism.

Difference between Semaphore vs. Mutex

Parameters	Semaphore	Mutex
Mechanism	It is a type of signaling mechanism.	It is a locking mechanism.
Data Type	Semaphore is an integer variable.	Mutex is just an object.
Modification	The wait and signal operations can modify a semaphore.	It is modified only by the process that may request or release a resource.
Resource management	If no resource is free, then the process requires a resource that should execute wait operation. It should wait until the count of the semaphore is greater than 0.	If it is locked, the process has to wait. The process should be kept in a queue. This needs to be accessed only when the mutex is unlocked.
Thread	You can have multiple program threads.	You can have multiple program threads in mutex but not simultaneously.
Ownership	Value can be changed by any process releasing or obtaining the resource.	Object lock is released only by the process, which has obtained the lock on it.
Types	Types of Semaphore are counting semaphore and binary semaphore and	Mutex has no subtypes.
Operation	Semaphore value is modified using wait () and signal () operation.	Mutex object is locked or unlocked.
Resources Occupancy	It is occupied if all resources are being used and the process requesting for resource performs wait () operation and blocks itself until semaphore count becomes >1.	In case if the object is already locked, the process requesting resources waits and is queued by the system before lock is released.

Advantages of Semaphores

Here, are pros/benefits of using Semaphore:

- It allows more than one thread to access the critical section
- Semaphores are machine-independent.
- Semaphores are implemented in the machine-independent code of the microkernel.
- They do not allow multiple processes to enter the critical section.

- As there is busy waiting in semaphore, there is never a wastage of process time and resources.
- They are machine-independent, which should be run in the machine-independent code of the microkernel.
- They allow flexible management of resources.

Disadvantage of semaphores

Here, are cons/drawback of semaphore

- One of the biggest limitations of a semaphore is priority inversion.
- The operating system has to keep track of all calls to wait and signal semaphore.
- Their use is never enforced, but it is by convention only.
- In order to avoid deadlocks in semaphore, the Wait and Signal operations require to be executed in the correct order.
- Semaphore programming is a complicated, so there are chances of not achieving mutual exclusion.
- It is also not a practical method for large scale use as their use leads to loss of modularity.
- Semaphore is more prone to programmer error.
- It may cause deadlock or violation of mutual exclusion due to programmer error.

Summary:

- Semaphore is defined as a variable that is non-negative and shared between threads.
- It is a mechanism that can be used to provide synchronization of tasks.
- Counting semaphore uses a count that helps task to be acquired or released numerous times.

- The binary semaphores are quite similar to counting semaphores, but their value is restricted to 0 and 1.
- Wait operation helps you to control the entry of a task into the critical section
- Signal semaphore operation is used to control the exit of a task from a critical section
- Counting Semaphore has no mutual exclusion whereas Binary Semaphore has Mutual exclusion
- Semaphore means a signaling mechanism whereas Mutex is a locking mechanism
- Semaphore allows more than one thread to access the critical section
- One of the biggest limitations of a semaphore is priority inversion.