Operating System

PLACEMENT PREPARATION [EXCLUSIVE NOTES]

MUST SAVE AND SHARE WITH YOUR CONNECTIONS

CURATED BY- HIMANSHU KUMAR (LINKEDIN) https://www.linkedin.com/in/himanshukumarmahuri

CREDIT-INTERNET

What is an Operating System?

An operating system is a software that manages the computer hardware. It acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner. It records that the hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.

Various relevant definitions of Operating System:

- An operating system is a program that controls the execution of application programs and acts as an interface between the user of a computer and the computer hardware.
- A more common definition is that the operating system is the one program running at all times on the computer (usually called the kernel), with all else being application programs.
- An operating system is concerned with the allocation of resources and services, such as memory, processors, devices, and information. The operating system correspondingly includes programs to manage these resources, such as a traffic controller, a scheduler, memory management module, I/O programs, and a file system.

Key functions of the Operating System: -

- Convenience: An OS makes a computer more convenient to use.
- Efficiency: An OS allows the computer system resources to be used in an efficient manner.
- Ability to Evolve: An OS should be constructed in such a way as to permit the effective development, testing and introduction of new system functions at the same time without interfering with service.

Types of Operating System: -

- Single Tasking Operating System - This type of Operating System can run only one program at a particular time.
- Multiprogramming Operating System - A computer running more than one program at a time (like running Excel and Firefox simultaneously).
- Multitasking Operating System – Tasks sharing a common resource (like 1 CPU).
- Multiprocessing Operating System– A computer using more than one CPU at a time.
- Multithreading Operating System - This is an extension of multitasking.
- Multiuser Operating System - This refers to the system where multiple users sitting on different computers can access a single OS resource.

Primary components in a Computer System with an Operating System: -

1. User
2. System and application programs
3. Operating system
4. Hardware

Every general-purpose computer consists of the hardware, operating system, system programs, and application programs. The hardware consists of memory, CPU, ALU, and I/O devices, peripheral device, and storage device. System program consists of compilers, loaders, editors, OS, etc. The application program consists of business programs, database programs. Every computer must have an operating system to run other programs. The

operating system is a set of special programs that coordinate the use of the hardware among the various system programs and application programs for various users. It simply provides an environment within which other programs can do useful work. It performs basic tasks such as recognizing input from the keyboard, keeping track of files and directories on the disk, sending output to the display screen and controlling peripheral devices. OS is designed to serve two basic purposes:

1. It controls the allocation and use of the computing System's resources among the various user and tasks.
2. It provides an interface between the computer hardware and the programmer that simplifies and makes feasible for coding, creation, debugging of application programs.

The Operating system must support the following tasks.
The task is:-

- Provides the facilities to create, modification of programs and data files using an editor.
- Access to the compiler for translating the user program from high-level language to machine language.
- Provide a loader program to move the compiled program code to the computer's memory for execution.
- Provide routines that handle the details of I/O programming.

Let's look at the I/O system and few drivers related to the hardware components that the OS needs to maintain its task:

I/O System Management - The module that keeps track of the status of devices is called the I/O traffic controller. Each I/O device has a device handler that resides in a separate process associated with that device.

The I/O subsystem consists of

- A memory Management component that includes buffering caching and spooling.
- A general device driver interface.

Assembler - The input to an assembler is an assembly language program. The output is an object program plus information that enables the loader to prepare

the object program for execution. At one time, the computer programmer had at his disposal a basic machine that interpreted, through hardware, certain fundamental instructions. He would program this computer by writing a series of ones and Zeros (Machine language), place them into the memory of the machine.

Compiler - The High-level languages- examples are FORTRAN, COBOL, ALGOL and PL/I are processed by compilers and interpreters. A compiler is a program that accepts a source program in a "high-level language "and produces a corresponding object program. An interpreter is a program that appears to execute a source program as if it was machine language. The same name (FORTRAN, COBOL, etc.) is often used to designate both a compiler and its associated language.

Loader - A Loader is a routine that loads an object program and prepares it for execution. There are various loading schemes: absolute, relocating and direct-linking. In general, the loader must load, relocate and link the object program. The loader is a program that places programs into memory and prepares them for execution. In a simple loading scheme, the assembler outputs the machine language translation of a program on a secondary device and a loader places it in the core. The loader places into memory the machine language version of the user's program and transfers control to it. Since the loader program is much smaller than the assembler, those make more core available to the user's program.

Examples of Operating System are -

- Windows (GUI based, PC)
- GNU/Linux (Personal, Workstations, ISP, File and print server, Three-tier client/Server)
- macOS (Macintosh), used for Apple's personal computers and work stations (MacBook, iMac).
- Android (Google's Operating System for smartphones/tablets/smartwatches)
- iOS (Apple's OS for iPhone, iPad and iPod Touch)

Advantages of Operating System:

1. GUI feature makes it easy to use.
2. Links the applications to the hardware, acting as an intermediary between the two.
3. Provides a user-friendly environment for the execution of programs and software.
4. Using abstraction, OS saves the details and complexity of the underlying hardware.

TYPES OF OPERATING SYSTEM

An Operating System performs all the basic tasks like managing files, processes, and memory. Thus operating system acts as the manager of all the resources, i.e. resource manager. Thus, the operating system becomes an interface between user and machine.
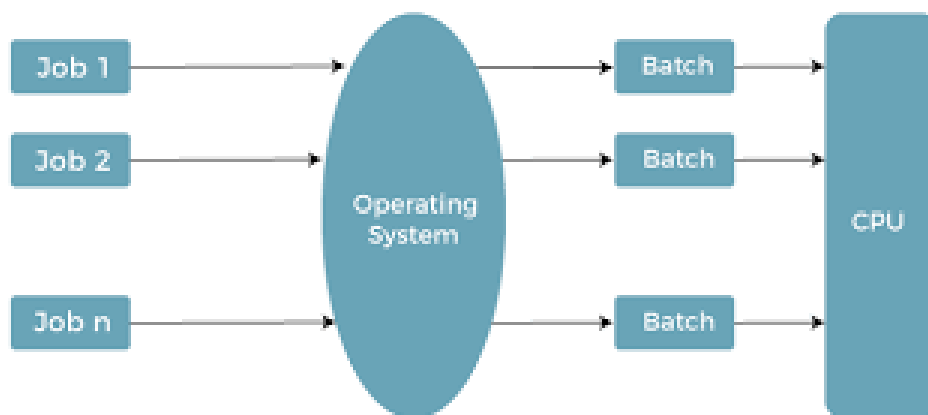
Types of Operating Systems:

Some widely used operating systems are as follows-

1. Batch Operating System

2. Time-Sharing Operating Systems

3. Distributed Operating System

4. Network Operating System

5. Real-Time Operating System

1. Batch Operating System -

This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirement and group them into batches. It is the responsibility of the operator to sort jobs with similar needs.

Advantages of Batch Operating System:

- It is very difficult to guess or know the time required for any job to complete. Processors of the batch systems know how long the job would be when it is in queue.
- Multiple users can share the batch systems
- The idle time for the batch system is very less
- It is easy to manage large work repeatedly in batch systems

Disadvantages of Batch Operating System:

- The computer operators should be well known with batch systems
- Batch systems are hard to debug
- It is sometimes costly
- The other jobs will have to wait for an unknown time if any job fails

Examples of Batch based Operating System: Payroll System, Bank Statements, etc.

2. Time-Sharing Operating Systems -

Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of CPU as they use a single system. These systems are also known as Multitasking Systems. The task can be from a single user or different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.
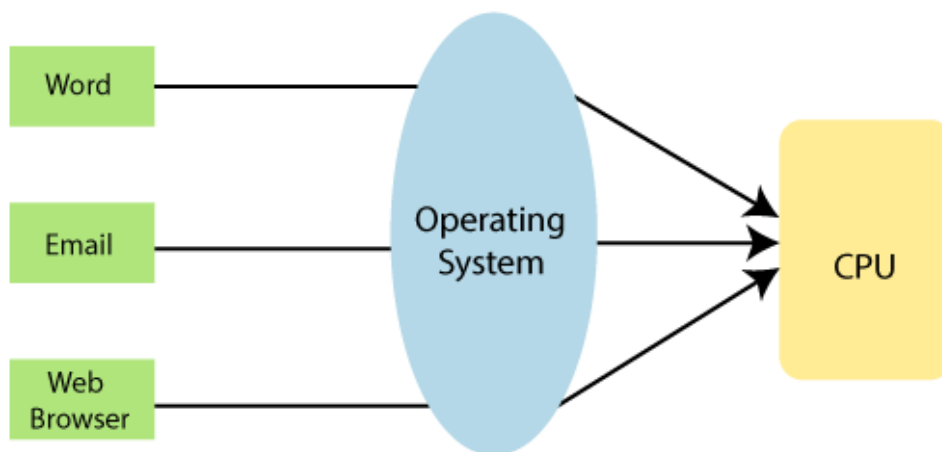
Advantages of Time-Sharing OS:

- Each task gets an equal opportunity
- Fewer chances of duplication of software
- CPU idle time can be reduced

Disadvantages of Time-Sharing OS:

- Reliability problem
- One must have to take care of the security and integrity of user programs and data
- Data communication problem.

Examples of Time-Sharing OSs are: Multics, Unix, etc.

3. Distributed Operating System -

These types of the operating system is a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, with a great pace. Various autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred to as loosely coupled systems or distributed systems. These system's processors differ in size and function. The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system, but some other system connected within this network i.e., remote access is enabled within the devices connected in that network.

Architecture of Distributed OS
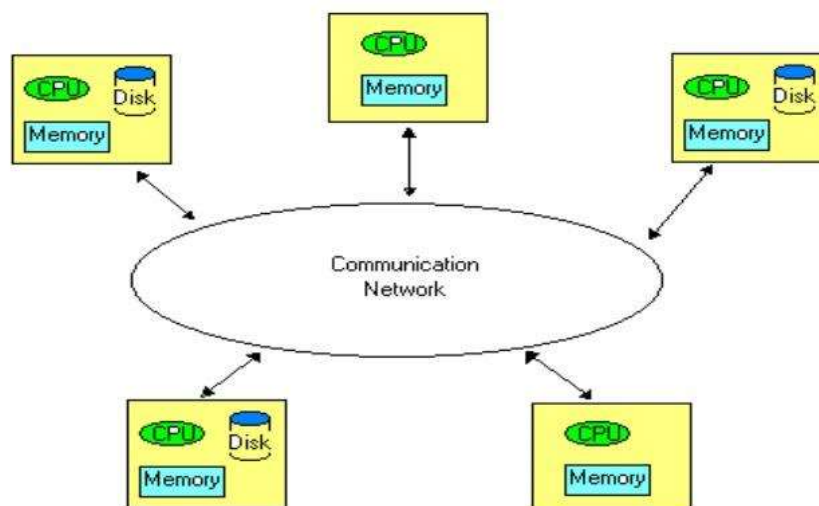
Advantages of Distributed Operating System:

- Failure of one will not affect the other network communication, as all systems are independent from each other
- Electronic mail increases the data exchange speed
- Since resources are being shared, computation is highly fast and durable
- Load on host computer reduces
- These systems are easily scalable as many systems can be easily added to the network
- Delay in data processing reduces

Disadvantages of Distributed Operating System:

- Failure of the main network will stop the entire communication
- To establish distributed systems the language which is used are not well defined yet
- These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet

Examples of Distributed Operating System are- LOCUS, etc.

4. Network Operating System -

These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions. These types of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network. One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections, etc. and that's why these computers are popularly known as tightly coupled systems.
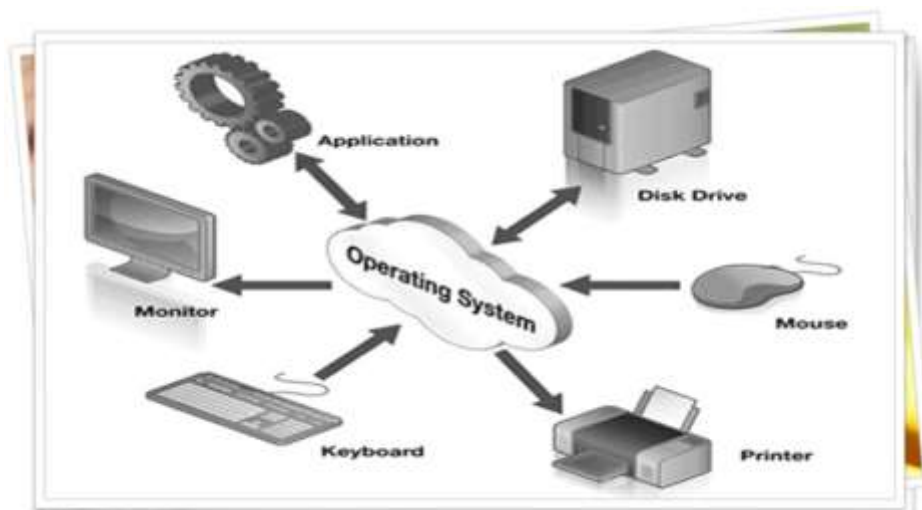
Advantages of Network Operating System:

- **Highly stable centralized servers**
- **Security concerns are handled through servers**
- **New technologies and hardware up-gradation are easily integrated into the system**
- **Server access is possible remotely from different locations and types of systems**

Disadvantages of Network Operating System:

- **Servers are costly**
- **User has to depend on a central location for most operations**
- **Maintenance and updates are required regularly.**

Examples of Network Operating System are: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD, etc.
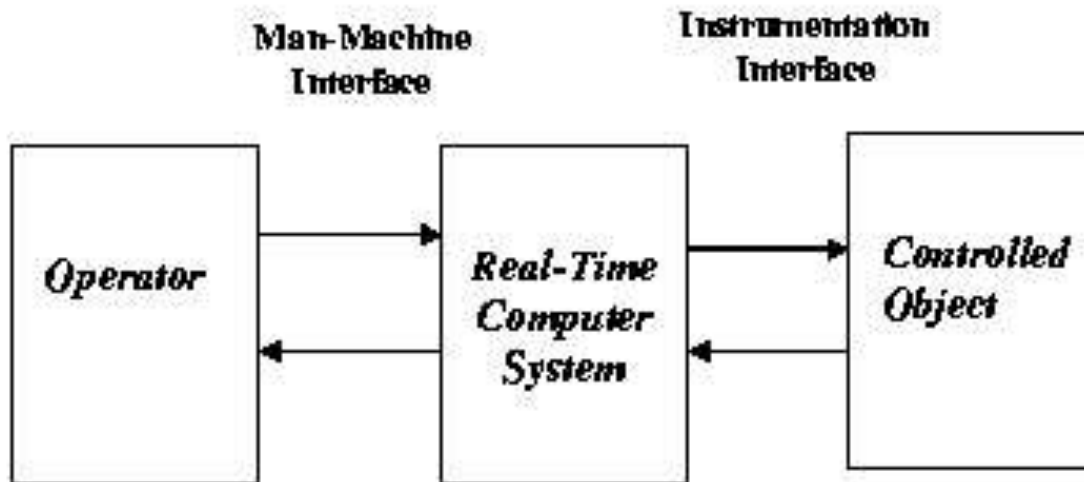
5. Real-Time Operating System –

These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called response time.

Real-time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

Two types of Real-Time Operating System which are as follows:

- **Hard Real-Time Systems:**
  These OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or airbags which are required to be readily available in case of any accident. Virtual memory is rarely found in these systems.

- **Soft Real-Time Systems:**
  These OSs are for applications where for time-constraint is less strict.

Man-Machine Interface · Instrumentation Interface

Operator → Real-Time Computer System → Controlled Object

Advantages of RTOS:

- **Maximum Consumption:** Maximum utilization of devices and system, thus more output from all the resources
- **Task Shifting:** The time assigned for shifting tasks in these systems are very less. For example, in older systems, it takes about 10 microseconds in shifting one task to another, and in the latest systems, it takes 3 microseconds.
- **Focus on Application:** Focus on running applications and less importance to applications which are in the queue.
- **Real-time operating system in the embedded system:** Since the size of programs are small, RTOS can also be used in embedded systems like in transport and others.
- **Error Free:** These types of systems are error-free.
- **Memory Allocation:** Memory allocation is best managed in these types of systems.

Disadvantages of RTOS:

- **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on few applications to avoid errors.
- **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.
- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
- **Device driver and interrupt signals:** It needs specific device drivers and interrupts signals to respond earliest to interrupts.

- **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.

Examples of Real-Time Operating Systems are: Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

MULTI-THREADING IN OPERATING SYSTEMS

A thread is a path which is followed during a program's execution. Majority of programs written now a days run as a single thread.

Lets say, for example a program is not capable of reading keystrokes while making drawings. These tasks cannot be executed by the program at the same time. This problem can be solved through multitasking so that two or more tasks can be executed simultaneously.

Multitasking is of two types:

Processor based and thread based. Processor based multitasking is totally managed by the OS, however multitasking through multithreading can be controlled by the programmer to some extent.

The concept of multi-threading needs proper understanding of these two terms - a process and a thread. A process is a program being executed. A process can be further divided into independent units known as threads.

A thread is like a small light-weight process within a process. Or we can say a collection of threads is what is known as a process.

single-threaded process   multithreaded process
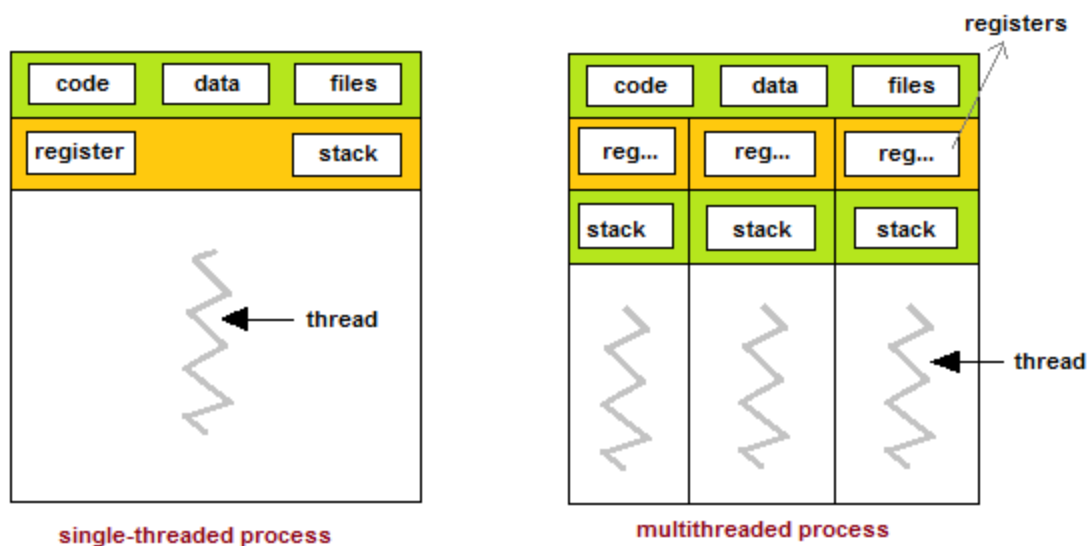
Applications - Threading is used widely in almost every field. Most widely it is seen over the internet now days where we are using transaction processing of every type like recharges, online transfer, banking etc. Threading is a segment which divide the code into small parts that are of very light weight and has less burden on CPU memory so that it can be easily worked out and can achieve goal in desired field. The concept of threading is designed due to the problem of fast and regular changes in technology and less the work in different areas due to less application. Then as says "need is the generation of creation or innovation" hence by following this approach human mind develop the concept of thread to enhance the capability of programming.

DIFFERENCE BETWEEN THREAD AND PROCESS

Process:

Processes are basically the programs that are dispatched from the ready state and are scheduled in the CPU for execution. PCB(Process Control Block) holds the concept of process. A process can create other processes which are known as Child Processes. The process takes more time to terminate and it is isolated means it does not share the memory with any other process.

The process can have the following states new, ready, running, waiting, terminated, and suspended.

Thread:

Thread is the segment of a process means a process can have multiple threads and these multiple threads are contained within a process. A thread has three states: Running, Ready, and Blocked.

The thread takes less time to terminate as compared to the process but unlike the process, threads do not isolate.
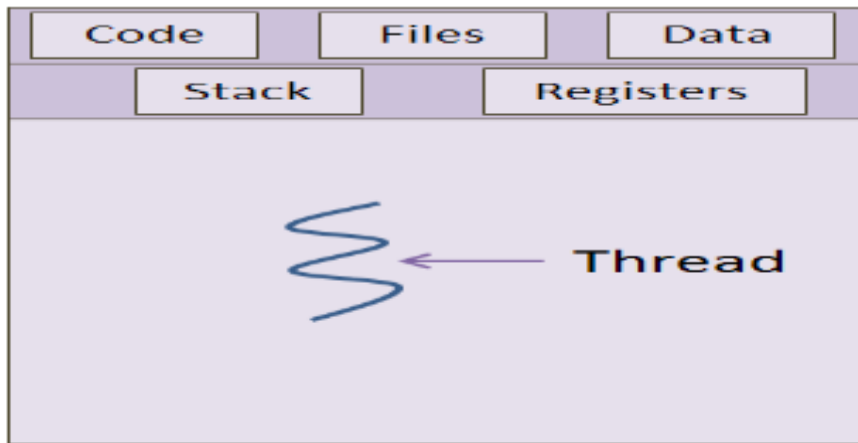
Process vs Thread: -

PROCESS VERSUS THREAD

PROCESS	THREAD
An instance of a computer program that is being executed	A component of a process which is the smallest execution unit
Heavyweight	Lightweight
Switching requires interacting with the operating system	Switching does now require interacting with the operating system
Each has its own memory space	Use the memory of the process they belong to
Requires more resources	Requires minimum resources
Difficult to create a process	Easier to create
Inter-process communication is slow because each process has a different memory address	Inter-thread communication is fast because the threads share the same memory address of the process they belong to
In a multi-processing environment, each process executes independently	A thread can read, write or modify data of another thread

Visit www.PEDIAA.com

USER LEVEL THREAD	KERNEL LEVEL THREAD
User thread are implemented by users.	kernel threads are implemented by OS.
OS doesn't recognized user level threads.	Kernel threads are recognized by OS.
Implementation of User threads is easy.	Implementation of Kernel thread is complicated.
Context switch time is less.	Context switch time is more.
Context switch requires no hardware support.	Hardware support is needed.
If one user level thread perform blocking operation then entire process will be blocked.	If one kernel thread perform blocking operation then another thread can continue execution.
Example : Java thread, POSIX threads.	Example : Window Solaris.

CREDIT-INTERNET