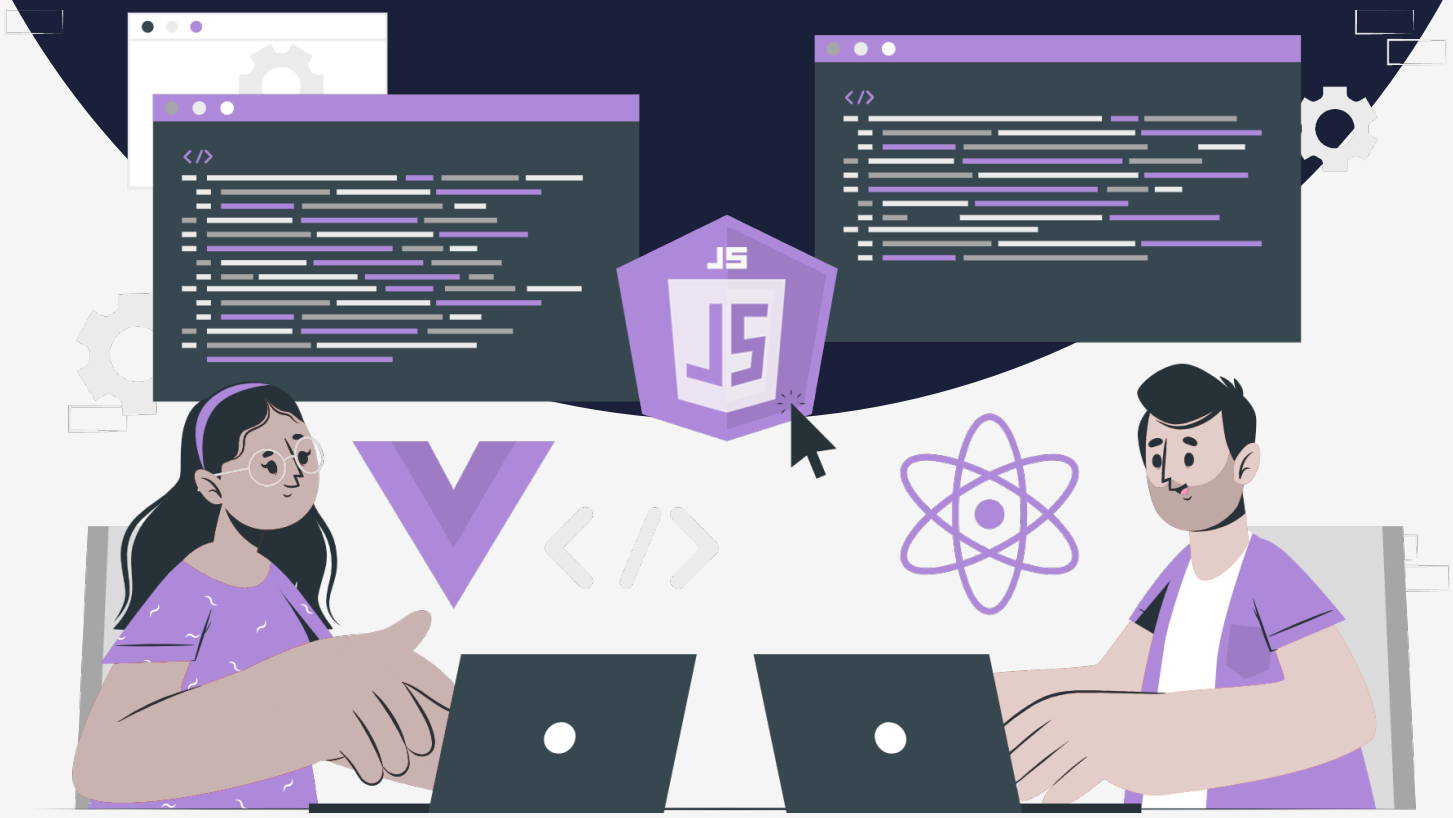
Lesson:

Introduction to OS



PW SKILLS

Topics Covered

1. Introduction.
2. Operating system.
3. What is the need for an OS?
4. Kernel.
5. Development of first OS.
6. Types of OS.
7. Features of OS.

An Operating System (OS) is one of the most important pieces of software that runs on any computer system. It is responsible for managing the hardware resources and providing a layer of abstraction that enables users to interact with the machine.

An OS is a complex and multifaceted piece of software that has evolved over decades to become the powerful tool that it is today. From the earliest days of computing to the modern era of cloud-based systems and mobile devices, the OS has played a critical role in shaping the way we interact with technology, and its importance only continues to grow as we rely more and more on computers in our daily lives.

In this lecture, we will delve into the fascinating world of OS and explore how it has evolved over time to become the backbone of modern computing. We will examine the various types of OS, including desktop, server, mobile, and embedded systems, and learn about their unique features and capabilities. We will also discuss the key concepts and principles that underpin OS design and explore the latest trends and developments in this fast-moving field.

Operating system.

An Operating System (OS) is a software program that serves as a bridge between a computer's hardware and software, enabling the efficient use of system resources and providing an interface for users to interact with the machine. An OS manages system resources such as memory, CPU, storage, and input/output devices, and provides services such as process management, file management, and security. It also abstracts hardware details, providing a layer of abstraction that enables software applications to run on different types of hardware without requiring modification.

There are several types of OS, including desktop, server, mobile, and embedded systems. Desktop OS, such as Microsoft Windows and macOS, are designed to run on personal computers and provide a graphical user interface (GUI) for users to interact with. Server OS, such as Linux and Microsoft Windows Server, is designed to run on servers and provide services such as web hosting, file sharing, and database management.

Mobile OS, such as Android and iOS, is designed for mobile devices such as smartphones and tablets and provides a touch-based interface for users to interact with. Embedded OS, such as VxWorks and QNX, is designed for embedded systems such as routers, medical equipment, and industrial control systems.

The design of an OS is based on several key principles and concepts, including process management, memory management, file management, and security. Process management involves managing the execution of multiple programs or processes simultaneously, ensuring that each process gets a fair share of the system resources. Memory management involves managing the allocation and deallocation of memory,

ensuring that each process has access to the required amount of memory.

The OS is a constantly evolving field, and new trends and developments are emerging all the time. Cloud-based OS, such as Google's Chrome OS, are becoming increasingly popular, offering a lightweight, web-based interface that allows users to access applications and data from anywhere with an internet connection. Virtualization is also a growing trend, enabling multiple OS to run on a single physical machine, improving resource utilization, and reducing hardware costs.

What is the need for an OS?
Operating Systems (OS) is a crucial component of modern computing, providing a platform for running software applications and managing hardware resources. Without an OS, a computer system would be unable to function, as there would be no way to manage and control the underlying hardware components. The need for an OS arises from the fact that computer systems are complex machines that require a sophisticated layer of software to manage their various components and provide a user-friendly interface.

One of the primary functions of an OS is to manage memory and CPU resources. The OS allocates memory and CPU time to different software applications running on the system, ensuring that they run efficiently and without interference. It also manages input/output devices such as keyboards, mice, and printers, providing a standard interface for these devices that software applications can use.

Another critical function of an OS is to provide a layer of abstraction between hardware and software. This abstraction allows software applications to run on different types of hardware without modification, making it possible to develop software that can run on a wide range of devices. For example, an application developed for a Windows desktop can also run on a Windows laptop, as the underlying OS provides a consistent interface to the hardware components. This abstraction also makes it possible to develop software that is independent of the underlying hardware, reducing development costs and increasing portability.

Kernel.
The kernel is a critical component of any operating system, responsible for managing the system's resources and providing a platform for running software applications. It acts as a bridge between the hardware components of the system and the software applications that run on it. The kernel is a complex piece of software that is often described as the "brain" of the operating system.

The relationship between the kernel and the operating system is a close one, with the kernel acting as the core of the operating system. The kernel provides a standard interface to the system's hardware components, allowing software applications to interact with them without needing to know the details of the hardware. The operating system provides a layer of software that sits on top of the kernel, providing additional functionality such as user interfaces, file systems, and networking capabilities.

The kernel and the operating system are often developed together, with the kernel providing the core functionality that is needed to support the operating system's features. The design of the kernel is closely tied to the design of the operating system, as the two components need to work together seamlessly to provide a stable and reliable platform for running software applications.

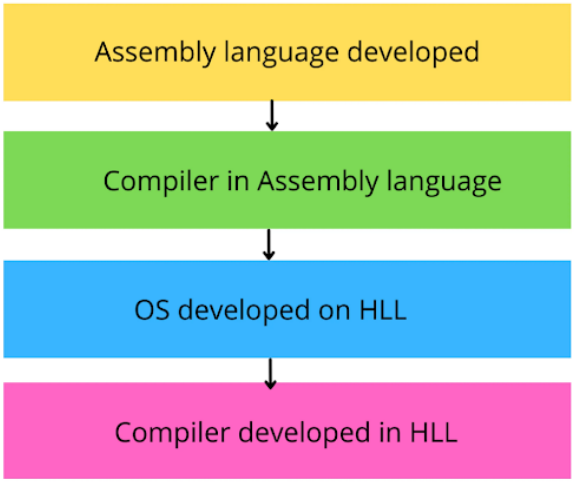
The design of the kernel is critical to the performance and stability of the operating system. A well-designed kernel can provide a stable and reliable platform for running software applications, while a poorly designed kernel can lead to instability and poor performance. The kernel needs to be designed with scalability in mind, as modern operating systems need to support a wide range of hardware configurations and software applications.

First OS.
The mid-1950s marked the emergence of the first Operating System. Prior to this, computers were only able to execute one program at a time. The early Operating Systems were responsible for enabling multiple applications to run concurrently while accurately tracking the CPU's billing expenses. They acted as supervisors, overseeing the single-stream batch processing system. Along with program management, they were also responsible for I/O operations.

General Motors' GMOS for IBM 701 was the first Operating System. It paved the way for other early Operating Systems such as IBM OS/360 and DEC's TOPS-10. These early Operating Systems had limited functionalities and were generally designed for specific hardware platforms. They provided basic functionalities such as file management, device drivers, and memory management, but lacked modern features such as virtual memory and multitasking, which are now standard features of contemporary Operating Systems.

Development of first OS.
Back in the early days of computing, programming involved physically manipulating the machine's circuitry. Later, an assembly language was developed to write code for this purpose. As time progressed, programmers realized the need to write a compiler in this assembly language, which could compile user languages like C into assembly language.

Eventually, Operating Systems were developed using high-level languages like C and C++, and their respective compilers were also developed in the same language. This was a significant shift from earlier times when the Operating System was written in assembly language. By using high-level languages, Operating Systems became easier to develop and maintain. It also allowed for the development of a standard set of interfaces, enabling application programs to be written independently of the hardware platform, which ultimately led to the growth of the software industry.



Types of OS.
There are several types of Operating Systems, each designed for specific environments and purposes. The types of OS include:

1. Single-User OS
2. Multi-User OS
3. Real-Time OS
4. Network OS
5. Mobile OS
6. Embedded OS

Single-User OS
Single-User OS is a type of operating system that is designed to be used by a single user at a time. It is the most common type of operating system used on personal computers and workstations. It provides a user-friendly interface that enables the user to interact with the computer system. Examples of Single-User OS include Microsoft Windows and Mac OS.

Multi-User OS
Multi-User OS is a type of operating system that allows multiple users to access the system and its resources simultaneously. It is typically used in a networked environment where multiple users share the same resources, such as servers, printers, and databases. Examples of Multi-User OS include UNIX, Linux, and Windows Server.

Real-Time OS
Real-Time OS is a type of operating system that is designed to process data in real time. It is used in applications that require immediate and predictable responses, such as control systems, robotics, and aerospace systems. Examples of Real-Time OS include QNX, VxWorks, and Windows CE.

Network OS
Network OS is a type of operating system that is designed to manage and control network resources. It provides services such as file sharing, printer sharing, and remote access. Examples of Network OS include Novell NetWare and Windows Server.

Mobile OS
Mobile OS is a type of operating system that is designed to run on mobile devices such as smartphones and tablets. It provides a platform for mobile applications to run and access device-specific features such as cameras, GPS, and touch screens. Examples of Mobile OS include Android, iOS, and Windows Phone.

Embedded OS
Embedded OS is a type of operating system that is designed to run on embedded systems, such as consumer electronics, medical devices, and automotive systems. It is typically lightweight and optimized for specific hardware and application requirements. Examples of Embedded OS include VxWorks, Nucleus RTOS, and Embedded Linux.

Features of OS.
Some common features of Operating Systems are:

1. Process Management: Operating systems manage the execution of processes or programs. They allocate resources like CPU time and memory to these processes and monitor their status to ensure they are running smoothly.
2. Memory Management: Operating systems manage the allocation of memory to running processes. They ensure that each process has enough memory to run, and protect the memory space of each process from other processes.
3. Device Management: Operating systems manage the interaction of the computer system with external devices such as printers, scanners, and USB drives. They allocate system resources to these devices and provide a standard interface for interacting with them.
4. File Management: Operating systems manage the storage and retrieval of data on a computer system. They organize data into files and directories, manage access control to these files, and ensure data is backed up and recoverable in case of a system failure.
5. Security: Operating systems provide security features to protect the computer system and the data it contains. They authenticate users, control access to system resources, and prevent unauthorized access and malicious attacks.

6. Networking: Operating systems manage the interaction of the computer system with networks, such as the internet. They provide networking protocols and interfaces for data transfer and manage security features such as firewalls.