Operating System Tutorial

An Operating System (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. When you start using a Computer System then it's the Operating System (OS) which acts as an interface between you and the computer hardware. The operating system is really a low level **Software** which is categorised as a **System Software** and supports a computer's basic functions, such as memory management, tasks scheduling and controlling peripherals etc.

This simple and easy tutorial will take you through step by step approach while learning Operating System concepts in detail.

What is Operating System?

An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

Generally, a **Computer System** consists of the following components:

- **Computer Users** are the users who use the overall computer system.

- **Application Softwares** are the softwares which users use directly to perform different activities. These softwares are simple and easy to use like Browsers, Word, Excel, different Editors, Games etc. These are usually written in high-level languages, such as Python, Java and C++.

- **System Softwares** are the softwares which are more complex in nature and they are more near to computer hardware. These software are usually written in low-level languages like assembly language and includes **Operating Systems** (Microsoft Windows, macOS, and Linux), Compiler, and Assembler etc.

- **Computer Hardware** includes Monitor, Keyboard, CPU, Disks, Memory, etc.

So now let's put it in simple words:

> *If we consider a Computer Hardware is body of the Computer System, then we can say an Operating System is its soul which brings it alive ie.*

operational. We can never use a Computer System if it does not have an Operating System installed on it.

Operating System - Examples

There are plenty of Operating Systems available in the market which include paid and unpaid (Open Source). Following are the examples of the few most popular Operating Systems:

- **Windows:** This is one of the most popular and commercial operating systems developed and marketed by Microsoft. It has different versions in the market like Windows 8, Windows 10 etc and most of them are paid.

- **Linux** This is a Unix based and the most loved operating system first released on September 17, 1991 by Linus Torvalds. Today, it has 30+ variants available like Fedora, OpenSUSE, CentOS, UBuntu etc. Most of them are available free of charges though you can have their enterprise versions by paying a nominal license fee.

- **MacOS** This is again a kind of Unix operating system developed and marketed by Apple Inc. since 2001.

- **iOS** This is a mobile operating system created and developed by Apple Inc. exclusively for its mobile devices like iPhone and iPad etc.

- **Android** This is a mobile Operating System based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.

Some other old but popular Operating Systems include Solaris, VMS, OS/400, AIX, z/OS, etc.

Operating System - Functions

To brief, Following are some of important functions of an operating System which we will look in more detail in upcoming chapters:

- Process Management

- I/O Device Management

- File Management

- Network Management

- Main Memory Management

- Secondary Storage Management

- Security Management

- Command Interpreter System

- Control over system performance
- Job Accounting
- Error Detection and Correction
- Coordination between other software and users
- Many more other important tasks

Operating Systems - History

Operating systems have been evolving through the years. In the 1950s, computers were limited to running one program at a time like a calculator, but later in the following decades, computers began to include more and more software programs, sometimes called libraries, that formed the basis for todays operating systems.

The first Operating System was created by General Motors in 1956 to run a single IBM mainframe computer, its name was the IBM 704. IBM was the first computer manufacturer to develop operating systems and distribute them in its computers in the 1960s.

There are few facts about Operating System evaluation:

- Stanford Research Institute developed the oN-Line System (NLS) in the late 1960s, which was the first operating system that resembled the desktop operating system we use today.

- Microsoft bought QDOS (Quick and Dirty Operating System) in 1981 and branded it as Microsoft Operating System (MS-DOS). As of 1994, Microsoft had stopped supporting MS-DOS.

- Unix was developed in the mid-1960s by the Massachusetts Institute of Technology, AT&T Bell Labs, and General Electric as a joint effort. Initially it was named MULTICS, which stands for Multiplexed Operating and Computing System.

- FreeBSD is also a popular UNIX derivative, originating from the BSD project at Berkeley. All modern Macintosh computers run a modified version of FreeBSD (OS X).

- Windows 95 is a consumer-oriented graphical user interface-based operating system built on top of MS-DOS. It was released on August 24, 1995 by Microsoft as part of its Windows 9x family of operating systems.

- Solaris is a proprietary Unix operating system originally developed by Sun Microsystems in 1991. After the Sun acquisition by Oracle in 2010 it was renamed Oracle Solaris.

Why to Learn Operating System

If you are aspiring to become a Great Computer Programmer then it is highly recommended to understand how exactly an Operating System works inside out. This

gives opportunity to understand how exactly data is saved in the disk, how different processes are created and scheduled to run by the CPU, how to interact with different I/O devices and ports.

There are various low level concepts which help a programmer to Design and Develop scalable softwares. Bottom line is without a good understanding of Operating System Concepts, it can't be assumed someone to be a good Computer **Application Software** developer, and even it is unimaginable imagine someone to become a **System Software** developer without knowing Operating System in-depth.

If you are a fresher and applying for a job in any standard company like Google, Microsoft, Amazon, IBM etc then it is very much possible that you will be asked questions related to Operating System concepts.

Target Audience

This tutorial has been prepared for the Computer Science Professionals and Students specially for BCA, MCA, B.Tech, M.Tech Engineering Students to help them understand the basic to advanced concepts related to an Operating System in general. Operating System is one of the core concepts in every University teaching Computer Science and this subject has a lot of weight from exams point of view.

Prerequisites

Before you start learning Operating System using this tutorial, we are making an assumption that you are already aware of Computer Fundaments like What is Computer Hardware, CPU, Primary Memory, Secondary Memory, Devices, Files etc. If you are not already aware of these concepts then it will be difficult to understand various concepts related to Operating System and so it is highly recommended to go through our Computer Fundamentals Tutorial before attempting to learn Operating System.

Frequently Asked Questions about Operating System

There are some very Frequently Asked Questions(FAQ) about Operating System, this section tries to answer them briefly.

> What is an Operating System?   ⌄

An operating system is like the manager of a computer. It is a software program that acts as an intermediary between the user and the computer's hardware, managing the resources of computer and providing a platform for running applications. The operating system controls tasks like managing memory, handling input and output devices (like keyboards and printers), scheduling tasks for the CPU, and providing a user interface for interacting with the computer.

What was the first operating system developed? ⌄

The first operating system developed was called the "GM-NAA I/O," which stands for "General Motors North American Automotive Operations Input/Output." It was created by General Motors in the early 1950s for their IBM 701 mainframe computer. The GM-NAA I/O was designed to manage input and output operations, such as reading data from punch cards and writing results to printers, for General Motors' computing needs.

Which is latest operating system? ⌄

The latest operating systems for personal computers and laptops are −

- **Microsoft Windows 11 −** Developed by Microsoft, Windows 11 is the latest version of the Windows operating system. It offers a refreshed user interface, new productivity features, and improvements in performance and security.

- **macOS Monterey −** Developed by Apple, macOS Monterey is the latest version of the macOS operating system for Mac computers. It introduces new features like Universal Control, Focus mode, and enhancements to Safari and FaceTime.

- **Ubuntu 22.04 LTS −** Developed by Canonical, Ubuntu 22.04 LTS (Long-Term Support) is the latest version of the Ubuntu operating system for Linux-based computers. It offers updated software packages, improved performance, and stability.

How long will it take to learn operating system? ⌄

The time it takes to learn about operating systems can vary depending on factors like your prior knowledge, the depth of understanding you want to achieve, and how much time you dedicate to learning. Learning the basics of operating systems, such as understanding their functions, components, and basic concepts, could take a few weeks to a few months.

However, becoming proficient in operating systems, including understanding advanced topics like process management, memory management, file systems, and networking, may take several months to a year or more of consistent learning and practice.

Who invented the operating system? ⌄

The concept of the operating system evolved over time with contributions from many individuals and organizations. One early example of an operating system was the GM-

NAA I/O, developed by General Motors in the 1950s for their IBM 701 mainframe computer.

However, the modern concept of an operating system, as we know it today, began to take shape in the 1960s with the development of systems like CTSS (Compatible Time-Sharing System) and MULTICS (Multiplexed Information and Computing Service). These systems laid the groundwork for later operating systems like Unix, developed by AT&T Bell Labs in the 1970s, which greatly influenced subsequent operating systems, including Linux and macOS.

What are the 4 types of operating system? ⌄

The four main types of operating systems are −

- **Single-User, Single-Tasking Operating Systems −** These operating systems allow only one user to work on the computer at a time, and they can handle only one task or program at a time. Examples include early versions of MS-DOS.

- **Single-User, Multi-Tasking Operating Systems −** These operating systems allow one user to run multiple programs or tasks simultaneously. The operating system switches between tasks quickly, giving the illusion of running multiple programs at once. Examples include modern versions of Windows, macOS, and Linux.

- **Multi-User Operating Systems −** These operating systems allow multiple users to work on the same computer simultaneously. Each user can have their own user account and run their own programs independently. Examples include UNIX-based systems like Linux and macOS, as well as server versions of Windows.

- **Real-Time Operating Systems (RTOS) −** These operating systems are designed to process data and respond to events in real-time, with strict deadlines for completing tasks. They are used in applications where timing is critical, such as controlling industrial machinery, automotive systems, and aerospace applications. Examples include QNX and VxWorks.

What Is the Most Secure Operating System? ⌄

The security of an operating system depends on various factors, including its design, implementation, and the measures taken to protect it from vulnerabilities and threats. While no operating system can be considered entirely immune to security risks, some operating systems are known for their robust security features and practices.

For example, macOS, developed by Apple, is often praised for its strong security architecture, regular security updates, and built-in security features like Gatekeeper and

FileVault. Similarly, some Linux distributions, such as Ubuntu and Fedora, are highly regarded for their security measures, including timely security patches and a strong emphasis on open-source security practices.

What is Deadlock in Operating System (OS)? ⌄

A deadlock in an operating system occurs when two or more processes are unable to proceed because each is waiting for the other to release a resource they need. It is like a traffic jam where cars are blocking each other and unable to move forward. In a deadlock situation, none of the processes can make progress, leading to a standstill in the system. Deadlocks can happen when processes hold resources while waiting for others, creating a circular dependency that prevents any progress.

What is Spooling in an Operating System? ⌄

Spooling (Simultaneous Peripheral Operations On-line) in an operating system is a technique used to manage input and output (I/O) operations more efficiently. It works by temporarily storing data from input devices (like keyboards) or output devices (like printers) in a buffer, allowing the computer to continue processing other tasks while the data is being transferred to or from the device. This helps prevent the CPU from waiting idle for slow devices to finish their operations, improving overall system performance.

For example, when you print a document, the operating system may spool the data to a print spooler, which queues the print jobs and sends them to the printer in the background, allowing you to continue working on other tasks without waiting for the printing process to complete.

What is virtual memory in OS? ⌄

Virtual memory in an operating system is a technique that allows the computer to use a combination of its physical memory (RAM) and secondary storage (like a hard drive or SSD) to simulate more memory than it actually has. It works by temporarily transferring data between physical memory and secondary storage, allowing the computer to run programs that require more memory than what is physically available. When the physical memory becomes full, the operating system moves less frequently used data to the secondary storage, freeing up space in physical memory for other tasks. This process is transparent to the user and helps improve system performance by allowing more programs to run simultaneously without running out of memory.

TOP TUTORIALS

Python Tutorial

Java Tutorial

C++ Tutorial

C Programming Tutorial

C# Tutorial

PHP Tutorial

R Tutorial

HTML Tutorial

CSS Tutorial

JavaScript Tutorial

SQL Tutorial

TRENDING TECHNOLOGIES

Cloud Computing Tutorial

Amazon Web Services Tutorial

Microsoft Azure Tutorial

Git Tutorial

Ethical Hacking Tutorial

Docker Tutorial

Kubernetes Tutorial

DSA Tutorial

Spring Boot Tutorial

SDLC Tutorial

Unix Tutorial

CERTIFICATIONS

Business Analytics Certification

Java & Spring Boot Advanced Certification

Data Science Advanced Certification

Cloud Computing And DevOps

Advanced Certification In Business Analytics

Artificial Intelligence And Machine Learning

DevOps Certification

Game Development Certification

Front-End Developer Certification

AWS Certification Training

Python Programming Certification

COMPILERS & EDITORS

Online Java Compiler

Online Python Compiler

Online Go Compiler

Online C Compiler

Online C++ Compiler

Online C# Compiler

Online PHP Compiler

Online MATLAB Compiler

Online Bash Terminal

Online SQL Compiler

Online Html Editor

ABOUT US | OUR TEAM | CAREERS | JOBS | CONTACT US | TERMS OF USE |
PRIVACY POLICY | REFUND POLICY | COOKIES POLICY | FAQ'S

tutorialspoint

GET IT ON Google Play

Download on the App Store

Tutorials Point is a leading Ed Tech company striving to provide the best learning material on technical and non-technical subjects.