Practice

Back To Course

LIVE BATCHES

Learn

Classroom

Theory

Quiz

Overview | Learn | Quiz | Filter

We have combined Classroom and Theory tab and created a new Learn tab for easy access. You can access Classroom and Theory from the left panel.

Introduction to Operating System

What is an Operating Mission? 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.

LIVE BATCHES

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 -

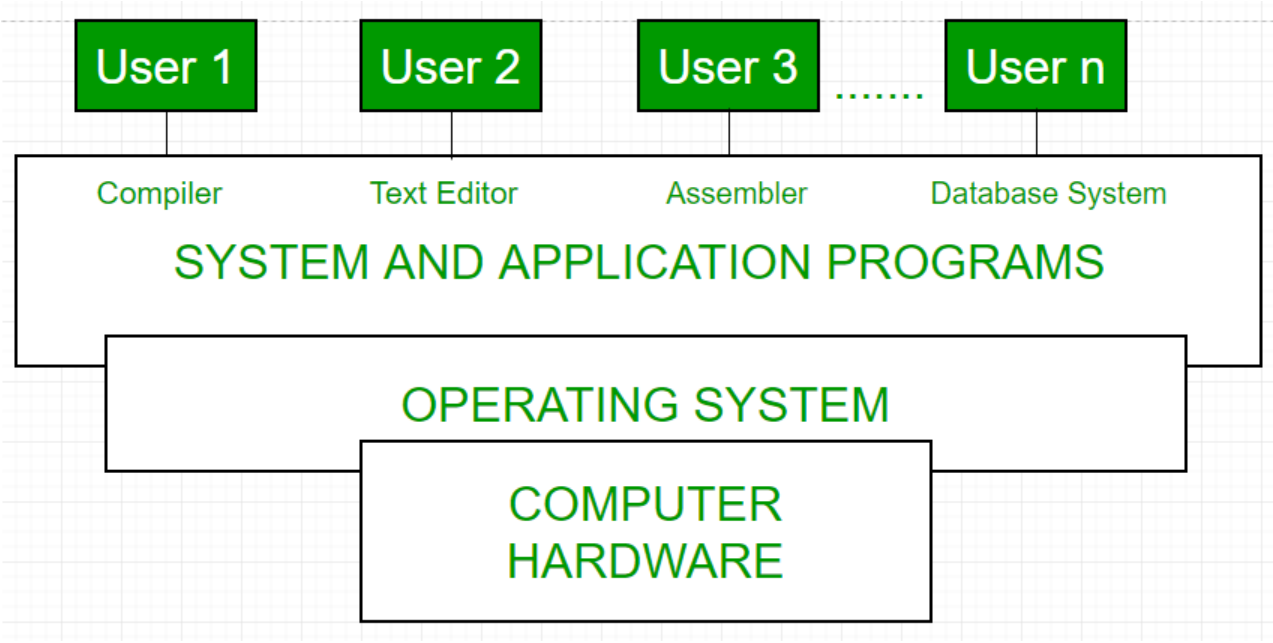
- **Single Tasking Operating System** - This type of Operating System can run only one program at a particular time.
- **Multiprogramming Operating System** - A computer running more than one program at a time (like running Excel and Firefox simultaneously).
- **Multitasking Operating System** – Tasks 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



LIVE BATCHES

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.

LIVE BATCHES

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.

LIVE BATCHES

Report An Issue
If you are facing any issue on this page. Please let us know.

GeeksforGeeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

About Us

Careers

Privacy Policy

Contact Us

Learn

Algorithms

Data Structures

Languages

CS Subjects

Terms of Service

Video Tutorials

Practice

Courses

Company-wise

Topic-wise

How to begin?

Contribute

Write an Article

Write Interview Experience

Internships

Videos

LIVE BATCHES

@geeksforgeeks , All rights reserved