**An operating system**, abbreviated OS, is a product, software type,  that is part of a system, equipment or computerized equipment, which deals with management and coordination of its activities. The computerized system may be a computer, a workstation, a server, a PC, a notebook, a smartphone, a road navigation device or another system with "intelligence" of its own. The operating system plays the role of host for all applications running on the equipment (hardware).

It is mainly used to provide the means for other software to communicate with Hardware at the lowest (fastest) level possible. There's all the bells and whistles, such as GUI (Graphical User Interface) which is what most OS's provide. Hard to give an example of what that means. What it doesn't mean, for example, are early UNIX systems or MS-DOS systems. There was nothing but a command line. No icons, start button, or mouse use for that matter (if you go far back enough).

It has set of programs that mediate access between physical devices (such as a keyboard, mouse, monitor, disk drive or network connection) and application programs (such as a word processor, [World-Wide Web](http://hitachi-id.com/concepts/world_wide_web.html) browser or electronic mail client).

At the foundation of all system software, the OS performs basic tasks such as controlling and allocating memory, prioritizing system requests, controlling input and output devices, facilitating networking, and managing files. It also may provide a graphical user interface for higher level functions. Various services performed by operating systems are discussed below.

**Process management:** It deals with running multiple processes. Most operating system allow a process to be assigned a priority which affects its allocation of CPU time. Interactive operating systems also employ some level of feedback in which the task with which the user is working receives higher priority. In many systems there is a background process which runs when no other process is waiting for the CPU.

**Memory management:** The memory manager in an OS coordinates the memories by tracking which one is available, which is to be allocated or deallocated and how to swap between the main memory and secondary memories. The operating system tracks all memory used by each process so that when a process terminates, all memory used by that process will be available for other processes.

**Disk and file systems:** Operating systems have a variety of native file systems that controls the creation, deletion, and access of files of data and programs.

**Networking:** Most current operating systems are capable of using the TCP/IP networking protocols. This means that one system can appear on a network of the other and share resources such as files, printers, and scanners. Many operating systems also support one or more vendor-specific legacy networking protocols as well.

**Security:** most operating systems include some level of security.

**Device drivers:** A device driver is a specific type of computer software developed to allow interaction with hardware devices. Typically this constitutes an interface for communicating with the device, through the specific computer bus or communications subsystem that the hardware is connected to, providing commands to and/or receiving data from the device, and on the other end, the requisite interfaces to the operating system and software applications**.**

**Some characteristics of an Operating System are:**

* Whether multiple programs can run on it simultaneously: *multi-tasking*
* Whether it can take advantage of multiple processors: *multi-processing*
* Whether multiple users can run programs on it simultaneously: *multi-user*
* Whether it can reliably prevent application programs from directly accessing hardware devices: *protected*
* Whether it has built-in support for graphics.
* Whether it has built-in support for networks.

**There are different types of operating systems. These are as follows:**

* 1. **Real-time Operating System:** It is a multitasking operating system that aims at executing real-time applications.
* 2. **Multi-user and Single-user Operating Systems:** The operating systems of this type allow a multiple users to access a computer system concurrently.
* 3. **Multi-tasking and Single-tasking Operating Systems:**When a single program is allowed to run at a time, the system is grouped under a single-tasking system, while in case the operating system allows the execution of multiple tasks at one time, it is classified as a multi-tasking operating system.
* 4. **Distributed Operating System:** An operating system that manages a group of independent computers and makes them appear to be a single computer is known as a distributed operating system.
* 5. **Embedded System:** The operating systems designed for being used in embedded computer systems are known as embedded operating systems.

**Functions:**

1. Program creation
2. Program execution
3. Access to Input / Output devices
4. Controlled access to files
5. System access
6. Error detection and response
7. Interpreting the commands
8. Managing peripherals
9. Memory management
10. Processor management
11. Information management
12. Process communication
13. Networking

**The major functions of an OS are:**  
  
- resource management,  
- data management,  
- job (task) management, and  
- standard means of communication between user and computer.  
  
The resource management function of an OS allocates computer resources such as CPU time, main memory, secondary storage, and input and output devices for use.  
  
The data management functions of an OS govern the input and output of the data and their location, storage, and retrieval.  
  
The job management function of an OS prepares, schedules, controls, and monitors jobs submitted for execution to ensure the most efficient processing. A job is a collection of one or more related programs and their data.  
  
A job is a collection of one or more related programs and their data.  
  
The OS establishes a standard means of communication between users and their computer systems. It does this by providing a user interface and a standard set of commands that control the hardware.  
  
Typical Day-to-Day Uses of an Operating System  
  
-Executing application programs.  
-Formatting floppy diskettes.  
-Setting up directories to organize your files.  
-Displaying a list of files stored on a particular disk.  
-Verifying that there is enough room on a disk to save a file.  
-Protecting and backing up your files by copying them to other disks for safekeeping.

**Operating system functions**

* To act as interface between hardware and users, an operating system must be able perform the following functions:
* 1. Enabling startup application programs. Thus, the operating system must have:
* - A text editor
* - A translator
* - An editor of links
* 2. The allocation of resources needed to execute programs is done by identifying: the programs that are running, the need for memory, peripheral devices and data protection requirements.
* 3. Facilities for data compression, sorting, mixing, cataloging and maintenance of libraries, through utility programs available.
* 4. Plan implementation works according to certain criteria, for efficient use of central processing unit.
* 5. Assisting implementation of programs through computer-user communication system, at both hardware and software level.

**Types of Operating System**

* DOS (Disk Operating System)
* UNIX
* LINUX
* Windows
* Windows NT.

Some popular Operating System's are:

* [Unix](http://hitachi-id.com/concepts/unix.html): multi-tasking, multi-processing, multi-user, protected, with built-in support for networking but not graphics.
* [Windows NT](http://hitachi-id.com/concepts/windows_nt.html): multi-tasking, multi-processing, single-user, protected, with built-in support for networking and graphics.
* Windows 95/98: multi-tasking, multi-processing, single-user, unprotected, with built-in support for networking and graphics.
* Windows 3.x: single-tasking, single-processing, single-user, unprotected, with built-in support for graphics but not networking.
* DOS: single-tasking, single-processing, single-user, unprotected with no built-in support for graphics or networking.
* [NetWare](http://hitachi-id.com/concepts/netware.html): multi-tasking, multi-processing, single-user, unprotected, with built-in support for networking but not graphics

How Do Operating Systems Differ?  
  
Operating systems for large computers are more complex and sophisticated than those for microcomputers because the operating systems for large computers must address the needs of a very large number of users, application programs, and hardware devices, as well as supply a host of administrative and security features.  
  
Operating system capabilities can be described in terms of  
  
-the number of users they can accommodate at one time,  
-how many tasks can be run at one time, and  
-how they process those tasks.  
  
Number of Users:  
  
A single-user operating system allows only one user at a time to access a computer.  
  
Most operating systems on microcomputers, such as DOS and Window 95, are single-user access systems.  
  
A multiuser operating system allows two or more users to access a computer at the same time (UNIX).  
  
The actual number of users depends on the hardware and the OS design.  
Time sharing allows many users to access a single computer.  
This capability is typically found on large computer operating systems where many users need access at the same time.  
  
Number of Tasks  
  
An operating system can be designed for single tasking or multitasking.  
  
A single tasking operating system allows only one program to execute at a time, and the program must finish executing completely before the next program can begin.  
  
A multitasking operating system allows a single CPU to execute what appears to be more than one program at a time.  
  
Context switching allows several programs to reside in memory but only one to be active at a time. The active program is said to be in the foreground. The other programs in memory are not active and are said to be in the background. Instead of having to quit a program and load another, you can simply switch the active program in the foreground to the background and bring a program from the background into the foreground with a few keystrokes.  
  
Cooperative multitasking in which a background program uses the CPU during idle time of the foreground program. For example, the background program might sort data while the foreground program waits for a keystroke.  
  
Time-slice multitasking enables a CPU to switch its attention between the requested tasks of two or more programs. Each task receives the attention of the CPU for a fraction of a second before the CPU moves on to the next. Depending on the application, the order in which tasks receive CPU attention may be determined sequentially (first come first served) or by previously defined priority levels.  
  
Multithreading supports several simultaneous tasks within the same application. For example, with only one copy of a database management system in memory, one database file can be sorted while data is simultaneously entered into another database file.

## Structure:

