An [operating system](http://www.blurtit.com/q811944.html) (OS) is an interface between hardware and user,   
which is responsible for the management and   
coordination of activities and the sharing of the resources of a   
computer, that acts as a host for computing   
applications run on the machine. One of the purposes of an operating  
system is to handle the resource allocation and access protection of   
the hardware. This relieves the application programmers from having to   
manage these details. There are many functions those are performed by the [Operating System](http://ecomputernotes.com/fundamental/disk-operating-system/what-is-operating-system) but the main goal of Operating system is to provide the Interface between the user. The various Functions those are Performed by the Operating System are as Explained below:-

**1) Operating System as a Resource Manager**

**Operating System Also Known as the Resource Manager** Means Operating System will Manages all the Resources those are Attached to the System means all the Resource like Memory and Processor and all the Input output Devices those are Attached to the System are Known as the Resources of the [Computer](http://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer) System and the Operating system will Manage all the Resources of the System. The Operating System will identify at which Time the CPU will perform which Operation and in which Time the Memory is used by which Programs. And which Input Device will respond to which Request of the user means When the Input and Output Devices are used by the which Programs. So this will manage all the Resources those are attached to the[computer](http://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer) System.

2                   **Storage Management:** **Operating System also Controls the all the Storage Operations means how the data or files will be Stored into the computers** and how the Files will be Accessed by the users etc. All the Operations those are Responsible for Storing and Accessing the Files is determined by the Operating System Operating System also Allows us Creation of Files, Creation of Directories and Reading and Writing the data of Files and Directories and also Copy the contents of the Files and the Directories from One Place to Another Place.

1)               **Process Management :** **The Operating System also Treats the Process Management means all the Processes those are given by the user or the Process those are System ‘s own Process are Handled by the Operating System** . The Operating System will Create the Priorities foe the user and also Start or Stops the Execution of the Process and Also Makes the Child Process after dividing the Large Processes into the Small Processes.

2)               **Memory Management:** Operating System also Manages the Memory of the Computer System means Provide the Memory to the Process and Also Deallocate the Memory from the Process. And also defines that if a Process gets completed then this will deallocate the Memory from the Processes. 

3)               **Extended Machine :** Operating System also behaves like an Extended Machine means Operating system also Provides us Sharing of Files between Multiple Users, also Provides Some Graphical Environments and also Provides Various Languages for Communications and also Provides Many Complex Operations like using Many Hardware’s and Software’s.

4)               **Mastermind:** Operating System also performs Many Functions and for those Reasons we can say that Operating System is a Mastermind. It provides Booting without an Operating System and Provides Facility to increase the Logical Memory of the Computer System by using the Physical Memory of the Computer System and also provides various Types of Formats Like NTFS and FAT File Systems. And Operating System also controls the Errors those have been Occurred into the Program and Also Provides Recovery of the System when the System gets Damaged Means When due to Some Hardware Failure , if System Doesn’t Works properly then this Recover the System and also Correct the System and also Provides us the Backup Facility. And Operating System also breaks the large program into the Smaller Programs those are also called as the threads. And execute those threads one by one.

Operating systems offer a number of services to application programs   
and users. Applications access these services through application  
programming interfaces (APIs) or system calls. By invoking these   
interfaces, the   
application can request a service from the operating system, pass   
parameters, and receive the results of the operation. Users may also   
interact with the operating system with some kind of software user   
interface like typing commands by using command line   
interface (CLI) or using a graphical user interface. For   
hand-held and desktop computers, the user interface is generally   
considered part of the operating system. On large systems such as   
Unix-like  
systems, the user interface is generally implemented as an application   
program that runs outside the operating system.  
While servers generally run Unix or some   
Unix-like operating system, embedded system markets are split amongst   
several operating systems,  
although the Microsoft Windows line of operating   
systems has almost 90% of the client PC market.Multi-user is a term that defines an operating system or application software that allows   
concurrent access by multiple users of a computer.  
Time-sharing systems are multi-user systems.   
Most batch processing systems for mainframe computers may also be considered "multi-user",   
to avoid leaving the CPU idle while it waits for I/O  
operations to complete. However, the term "[multitasking](http://www.blurtit.com/q935817.html)" is more common in this   
context.  
An example is a Unix server where multiple remote users have access   
(such as via Secure Shell) to the Unix   
shell prompt at the same time. Another example uses multiple X Window sessions spread across multiple terminals powered   
by a single machine - this is an example of the use of thin  
client.  
Management systems   
are implicitly designed to be used by multiple users, typically one system administrator or more and an end-user community.  
The complementary term, single-user, is most commonly used   
when talking about an operating system being usable only by one person   
at a time, or in [reference](http://www.blurtit.com/q590903.html) to a single-user software license agreement.   
Multi-user operating systems such as Unix sometimes have a single user   
state or runlevel available for emergency maintenance.

**Machine language**

*Machine language* is the only language that a computer understands. Each statement in a machine language program is a sequence of bits. Each bit may be set to 0 or 1. Series of bits represent instructions that a computer can understand. For example, the number 455 is represented by the bit sequence 111000111. Machine language is a low-level programming language. It is easily understood by computers but difficult to read by people. This is why people use higher level programming languages. Programs written in high-level languages are compiled and/or interpreted into machine language so computers can execute them.

**Assembly language**

*Assembly language* is a representation of machine language. In other words, each assembly language instruction translates to a machine language instruction. The advantage of assembly language is that its instructions are readable. For example, assembly language statements like MOV and ADD are more recognizable than sequences of 0s and 1s. Though assembly language statements are readable, the statements are still low-level. Another disadvantage of assembly language is that it is not *portable*. In other words, assembly language programs are specific to a particular hardware. Assembly language programs for a Mac will *not* work on a PC. But this can be an advantage for programmers who are targeting a specific platform and need full control over the hardware.

**High-level language**

*High-level languages* are what most programmers use. Languages such as C, C++ and Java are all high-level languages. One advantage of high-level languages is that they are very readable. The statements in these languages are English-like. For example, you can gain a basic understanding of what a Java program is doing by simply reading the program source code. High-level languages use English words as statements. Loops in Java programs are indicated by the words *for*, *while* and *do*. To get the current system time (ms) in Java you would call *System.currentTimeMillis()*. Another advantage of high-level languages is that they are less tedious to use. A single statement in a high-level language can translate into many machine language statements. Finally, high-level languages are usually portable.

A disadvantage of high-level languages is that they are usually less powerful and less efficient. Since statements are high-level, you cannot code at the bit level the way you can with assembly language. High-level languages also need to be compiled and/or interpreted into machine language before execution.

That is a basic description of the three basic types of programming languages. Until people can process information like computers, we will leave machine language to computers and use high-level programming languages instead.

Interpretation and compilation

Programming languages may be roughly divided into two categories:

* **interpreted languages**
* **compiled languages**

Interpreted language

A programming language is by definition different to machine code This must therefore be translated so that the processor can understand the code. A program written in an interpreted language requires an extra program (the interpreter) which translates the programs commands as needed.

Compiled language

A program written in a "**compiled**" language is translated by an additional program called a **compiler**which in turn creates a new stand-alone file which does not require any other program to execute itself, such a file is called an **executable**.

A program written in a compiled language has the advantage of not requiring an additional program to run it once it has been compiled. Furthermore, as the translation only needs to be done once, at compilation it executes much faster.   
However, it is not as flexible as a program written in an interpreted language, as each modification of the source file (the file understandable by humans: the file to be compiled) means that the program must be recompiled for the changes to take effect.

On the other hand, a compiled program has the advantage of guaranteeing the security of the source code. In effect, interpreted language, being a directly legible language, means that anyone can find out the secrets of a program and thus copy or even modified the program. There is therefore a risk of copyright violation. On the other hand, certain secure applications need code confidentiality to avoid illegal copying (bank transactions, on-line payments, secure communications...).