A process is a program in execution. It is a unit of work within the system. Program is a ***passive entity***, process is an ***active entity***.

A process is an active program. It can also be said as a program that is under execution. It is more than the program code as it includes the program counter, process stack, registers, program code etc. Compared to this, the program code is only the text section. A process passes through different states as it executes. These states may be different in different operating systems. However, the common process states are explained below ,

**New**

This is the state when the process has just been created. It is the initial state in the process life cycle.

**Ready**

In the ready state, the process is waiting to be assigned the processor by the short term scheduler, so it can run. This state is immediately after the new state for the process.

**Ready Suspended**

The processes in ready suspended state are in secondary memory. They were initially in the ready state in main memory but lack of memory forced them to be suspended and gets placed in the secondary memory.

**Running**

The process is said to be in running state when the process instructions are being executed by the processor. This is done once the process is assigned to the processor using the short-term scheduler.

**Blocked**

The process is in blocked state if it is waiting for some event to occur. This event may be I/O as the I/O events are executed in the main memory and don't require the processor. After the event is complete, the process again goes to ready state.

**Blocked Suspended**

This is similar to ready suspended. The processes in blocked suspended state are in secondary memory. They were initially in the blocked state in main memory waiting for some event but lack of memory forced them to be suspended and gets placed in the secondary memory. A process may go from blocked suspended to ready suspended if its work is done.

**Terminated**

The process is terminated once it finishes its execution. In the terminated state, the process is removed from main memory and its process control block is also deleted.

Three general methods used to pass parameters to the OS

1. Simplest: pass the parameters in registers

* + - In some cases, may be more parameters than registers

2.Parameters stored in a block*,* or table, in memory, and address of block passed as a parameter in a register

* + - This approach taken by Linux and Solaris

Block and stack methods do not limit the number or length of parameters being passed.

3.Parameters placed, or pushed*,* onto the stackby the program and poppedoff the stack by the operating system.

**1.Short-term scheduler** or **CPU scheduler** – selects which process should be executed next and allocates CPUs sometimes the only scheduler in a system. Short-term scheduler is invoked frequently (milliseconds) (must be fast)

**2.Long-term scheduler** or **job scheduler**– selects which processes should be brought into the ready queue. It is invoked infrequently (seconds, minutes) (may be slow).The long-term scheduler controls the degree of multiprogramming

Processes can be described as either:

**A .I/O-bound process** – spends more time doing I/O than computations, many short CPU bursts

**B .CPU-bound process** – spends more time doing computations; few very long CPU bursts

Long-term scheduler strives for good *process mix*

**3.Medium-term scheduler** can be added if degree of multiple programming needs to decrease remove process from memory, store on disk, bring back in from disk to continue execution: swapping

When a thread is created the threads does not require any new resources to execute the thread shares the resources like memory of the process to which they belong to. The benefit of code sharing is that it allows an application to have several different threads of activity all within the same address space. Whereas if a new process creation is very heavyweight because it always requires new address space to be created and even if they share the memory then the inter process communication is expensive when compared to the communication between the threads.

Advantages :

1.Easier to extend a microkernel

2.Easier to port the operating system to new architectures

3.More reliable (less code is running in kernel mode)

4.More secure

Disadvantage: Performance overhead of user space to kernel space communication.

As described microkernel use message passing as a form to communicate and exchange information between the user and the system services. The inter process communication unfortunately comes with overheads because of the frequent use of the operating systems messaging functions.

The difference is that clustered systems contain two or more independent systems ( in multiprocessor systems, processors are not so independent of each other)

For clustered systems to be efficient, we need to be sure that the system will not half if one or more systems in the cluster fail. This is done by adding a level of redundancy in the system. For example , if some system fails, some other system can take control of its storage and rerum the required applications.

Applications should:

Operating system is defined as the program which acts as the interface between the hardware and the computer user, controlling and allocating the resources, thus they are part of the operating system.

Also, the operating systems generally include all those applications which are provided by vendors. As many vendors provide mail programs and web browsers along with operating systems, they are also included in operating systems.

Applications should not:

As per computer architecture, the computer system is categorized into four major components. They are: application programs, operating system, hardware and user. The mail program and web browser belongs to application program, not to operating system.

Operating system is defined as the program, which is running all the time on the computer system. But the mail program and the web browsers do not run all the time on the system. Hence they can not be regarded as operating systems.

The microkernel must provide communication between the client program and the various services that are also running in user space. In this case communication is provided through message passing. The client program and service never interact directly only by exchanging messages with the microkernel.