what is an operating system?

A set of organised collection of programs or softwares which are used to manage the resources of a system

system-any electronic device

resources-I/O devices, CPU, memory

A **real-time operating system** (**RTOS**) is an [operating system](https://en.wikipedia.org/wiki/Operating_system) (OS) intended to serve [real-time](https://en.wikipedia.org/wiki/Real-time_computing) applications that process data as it comes in

A **real-time operating system**(RTOS) is software that supplements computer hardware complexities.

An **embedded operating system** is an [operating system](https://en.wikipedia.org/wiki/Operating_system) for [embedded computer systems](https://en.wikipedia.org/wiki/Embedded_system). Embedded operating systems are computer systems designed for a specific purpose, to increase functionality and reliability for achieving a specific task.

what is a process

**A process is a program under execution. A program is a passive entity that resides on a disk where as a process is an active entity residing in the RAM.**

what are resources

The process needs certain resources such as CPU and memory to perform the tasks.

Now we will look into the related commands and system calls to know the information on resource utilization and monitoring. Also there are certain limits by default for each process on the resources, and if required the limits can be enhanced to accommodate the application requirements.

* Two resource allocation techniques:
  + Resource partitioning approach
  + Pool based approach

The Operating System allocates resources when a program need them. When the program terminates, the resources are de-allocated, and allocated to other programs that need them. Now the question is, what strategy does the operating system use to allocate these resources to user programs?

There are two Resource allocation techniques:

1. **Resource partitioning approach –**  
   In this approach, the operating system decides beforehand, that what resources should be allocated to which user program. It divides the resources in the system to many *resource partitions*, where each partition may include various resources – for example, 1 MB memory, disk blocks, and a printer.

Then, it allocates one resource partition to each user program before the program’s initiation. A resource table records the resource partition and its current allocation status (Allocated or Free).

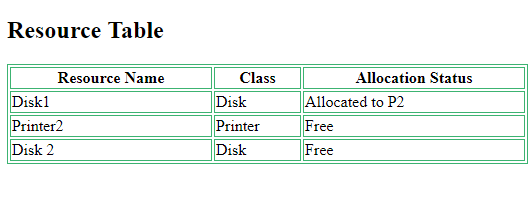
**Advantages:**

* + Easy to Implement
  + Less Overhead

**Disadvantages:**

* + **Lacks flexibility –** if a resource partition contains more resources than what a particular process requires, the additional resources are wasted.
  + If a program needs more resources than a single resource partition, it cannot execute (Though free resources are present in other partitions).

An example resource table may look like:



1. **Pool based approach –**  
   In this approach, there is a *common pool of resources*. The operating System checks the allocation status in the resource table whenever a program makes a request for a resource. If the resource is free, it allocates the resource to the program.

**Advantages:**

* + Allocated resources are not wasted.
  + Any resource requirement can be fulfilled if the resource is free (unlike Partitioning approach)

**Disadvantages:**

* + Overhead of allocating and de-allocating the resources on every request and release.

**Process States**

**System level context involves process id, user id, process states etc. In the life time of a process, it passes through different states which are as mentioned below.**

* **New: The process is being created**
* **Ready: The process is waiting for the CPU to execute it.**
* **Running: Instructions are being executed.**
* **Waiting: The process is waiting for some event (I/O operation) to occur.**
* **Terminated: The process has finished execution.**

when a process is created what are the attributes associated with it

process id

program counter

pcb

process state

priority

genereal purpose resisters

list of open files

list of open devices protection

**Process Descriptor**

process id

priority

process state

program counter