**Operating system**

A piece of software which interacts directly with the hardware, manages resources, and usually presents a user interface to the user.

Purpose of operating system let us you use the interface.

**Kernel**

Kernel is the heart of the operating system.

Process management for application execution

Memory management, allocation, and I/O

Device management using device drivers

System call control, which is essential for the execution of kernel services

**Real time operating systems**

Like most operating systems, real time operating systems run on hardware. The way tasks are managed is what sets it apart from a regular operating system. It is possible for multiple processes to run simultaneously in real-time operating systems. Best example: Traffic control, antilock brake system and airline reservations.

**Single - user single task**

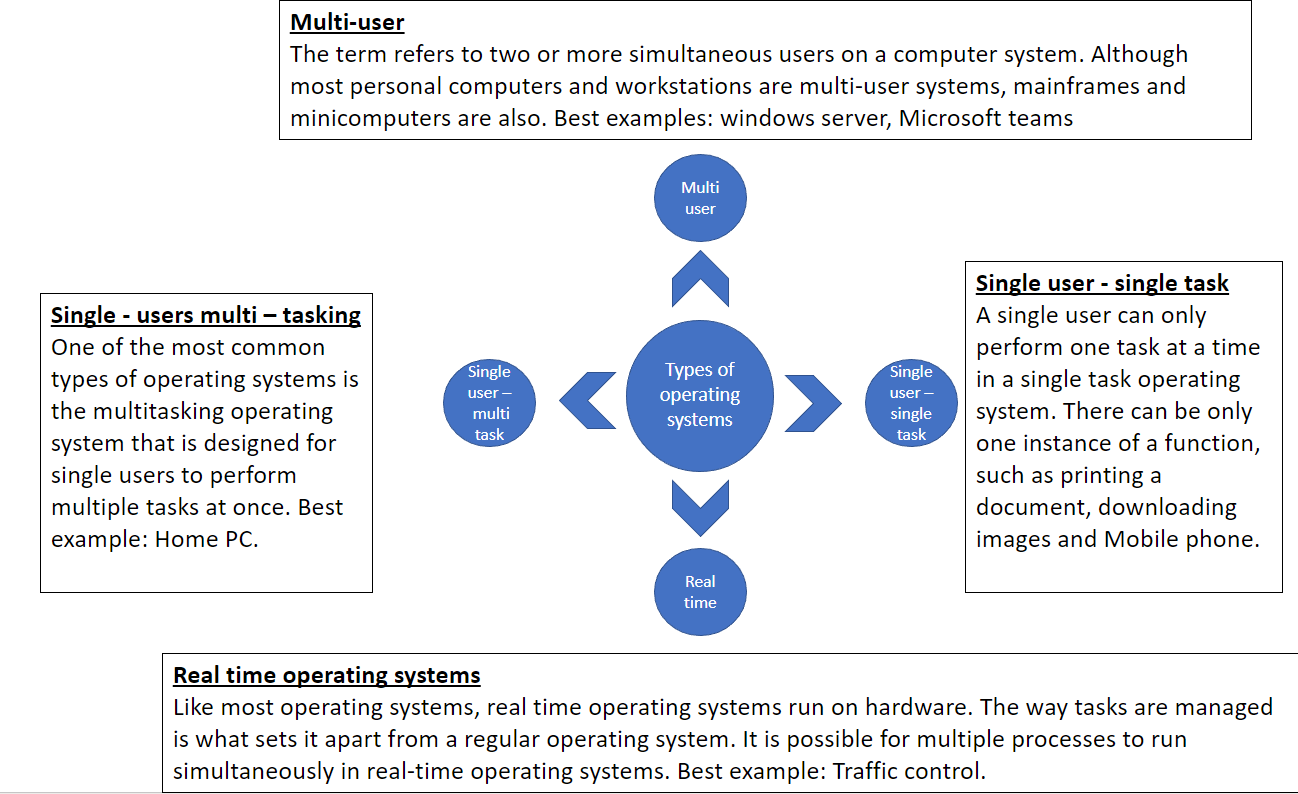
A single user can only perform one task at a time in a single task operating system. There can be only one instance of a function, such as printing a document, downloading images and Mobile phone.

**Single - users multi – tasking**

One of the most common types of operating systems is the multitasking operating system that is designed for single users to perform multiple tasks at once. Best example: Home PC.

**Multi-user**

The term refers to two or more simultaneous users on a computer system. Although most personal computers and workstations are multi-user systems, mainframes and minicomputers are also. Best examples: windows server, Microsoft teams



The kernel is the most important component of an operating system. It has the highest level of privilege (meaning it can access anything) without being challenged.

When a computer starts up, the pattern is:

* BIOS is loaded and executed.
* Bootloader is launched.
* Kernel loads.

The kernel manages all running programs and meditates between them and hardware resources.

* Programs are given tiny slices of time running on the CPU in a Rota system. This is called processing scheduling, and the kernel is responsible for determining which process should currently be running on.
* Usually this is achieved using a priority system – processes are marked as low through to high priority.
* When programs want to communicate with the hardware, this is achieved using system calls: calls to the kernel.

The kernel is also responsible for:

•Memory management

•Multi-tasking

•Disk access

•File systems

•Device drivers