The functions of an Operating System

 # An operating system is the core software (collections of applications) that allows a computer to run as a useful device; it manages the hardware, the user interface and all other software running on the computer making it a practical machine.

The operating system has four main parts: (no matter the type, they all share these 4 main components):

- The KERNEL
- The DEVICE DRIVERS
- The USER INTERFACE
- The SYSTEM UTILITIES

Types of operating systems

No.	Function	Description
1	**Manages processor**	Decides processing time for the execution of tasks (multi-tasking) & scheduling Handles interruptions to running processes.
2	**Manages memory**	• Programs are loaded into the memory address. • The allocation and management of memory space is controlled by the operating system. Memory spaces are later freed for reuse. • Dedicates virtual memory that can be used to compensate for the lack of RAM.
3	**Manages peripherals**	Uses device drivers to communicate with hardware Manages input and output signals to I/O devices. Data to be sent to a device is stored in buffer of CPU so CPU can continue with other tasks. Writes to peripheral devices such as storage attached to the computer
4	**Handle utility programs**	Tools to organise, and manager the hardware: File management (backing store), defragmenter, Firewall, auto updates/installers, antivirus, clipboard manager, system monitor, encryption and file compression.
5	**Networking**	Manages ingoing and outgoing signals from networks – interacting with other computers via cable/WIFI
6	**Handles security**	Prevents unauthorised access to user files by assigning access rights Protects user accounts with logins (usernames and passwords)
7	**User interface**	Provides a user interface so it is easy to interact with the computer: WIMP (Windows, icons, menus, pointers) Touch gestures

1 – Single user, single application

These operating systems are only capable of running one application at a time for only one user at a time. An example of this would be on a basic phone or an mp3 player.

The components tend to be very basic and only a small amount of main memory is needed.

2 - Multi-tasking operating system

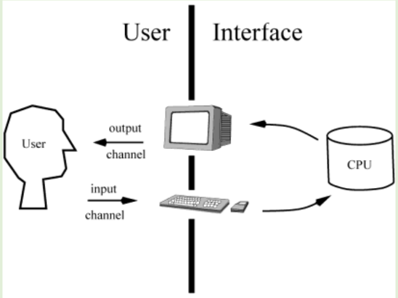
These operating systems are (appear to be) capable of running multiple programs at one time. Compared to a single user/single application operating system. It is quite possible for a multitasking computer to have multiple gigabytes of RAM (2GB to 16GB typically on a single user machine).

a) Single-user (& multi-tasking)

This is designed so that only one user can use the computer resources at any given time. It can **appear to** run multiple applications at the same time. For example: word processing while playing music.

This is the typical operating system for a personal computer.

Examples: Windows, Linux, Mac OS



b) Multi-user (& multi-tasking)

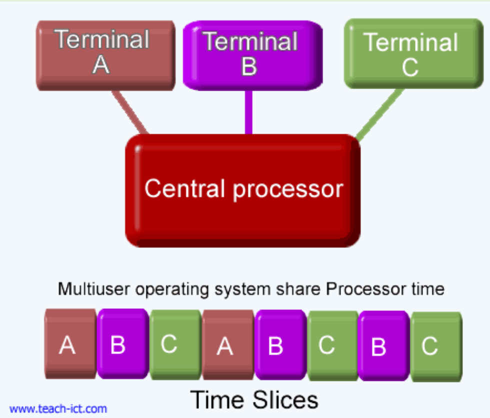
Usually found on large scale mainframes/supercomputers where many users are accessing the same resource simultaneously. They are highly sophisticated and designed to handle many users running multiple programs on the same computer at one time.

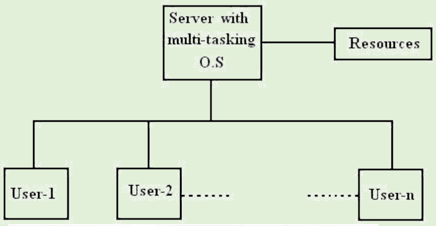
Additional requirements the operating system now must handle:

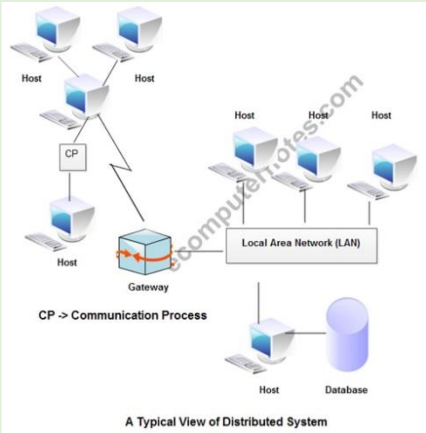
- Each user logged onto the system, their workspace and so on.
- Allocate resources to the jobs they want to run.
- Keep logs of how much processing time and resources they use
- Work out the most efficient use of computer processing cycles
- Maintain security

When a program is being executed in memory, this is called a '**process**'. You can have many people using the same process at the same time. Each person is running a '**thread**' **of execution within the process.** This makes the processing very complicated as there could be hundreds of processes running in the memory and each having many threads. The complicated task managing is known as **scheduling.**

Examples: UNIX, Linux, IBM's z/OS







3 - Distributed operating system

When a group of computers are combined to work on one particular task. A single task is split up into smaller tasks that are assigned to run on each individual computer. The results are then gathered at the end. A distributed operating system coordinates the activity of many computers, presenting them to the end user as if they were one single system.

These are useful in the film and CGI (computer-generated-graphics) industry. Rendering can take vast amounts of processing time.
A render farm is a large collection of computers networked together and controlled by a distributed operating system. Individual film frames can be rendered on each computer and gathered at the end.


4 - Embedded operating system

An embedded operating system is a specialized operating system for use in the computers that are integrated into larger electrical systems. The embedded computer - running an embedded operating system specialised for only one particular need of the computer - provides data processing power to handle incoming data.

They are specifically configured for a certain hardware configuration. These are designed to be compact, resource efficient and reliable. The OS is installed on an internal memory chip rather than a hard disk

Examples: Printers, washing machines, mp3 players, radios, Symbian used in cell phones (Nokia's) and Palm OS on PDA's.



5 - Real-time operating system

A real-time operating system is designed to handle input data, typically without buffering delays, within a guaranteed latency (time). They found in systems which aim to control a series of actions by using a computer (embedded computers).

The operating system carries out operations in real-time without large delay. It must run and so process at guaranteed times.
There can be dozens of rapidly changing input signals. It must take all these inputs and process them to produce a set of outputs that control another device **in designated orders with thin specified time windows, instead of dynamically switching between processes based on resource availability.**

This sometimes means it will process inputs with little noticeable delay, almost instantaneously. This is useful when there are safety risks associated with delay. For example, life or death situations – planes, braking systems, nuclear power plants. They are also common in car engines, e.g. for managing brakes, exhaust and fuel injection in carburettor.
However, real-time does not necessarily mean fast. Traffic lights use real-time operating systems but they will not act instantly, they act within seconds.
Real-time just means that they process and handle incoming inputs, typically without buffering delays and return an output in a guaranteed amount of time.