Sure trust

Difference between Computer and Embedded System

1. Computer: A computer is a combination of hardware and software resources that integrate together and provides various functionalities to the user.
2. Embedded Device : An embedded device is a part of an integrated system that is formed as a combination of computer hardware and software for a specific function and which can operate without human interaction.
Difference between Computer and Embedded System are as follows:

Category	Computer	Embedded device
Description	A computer is a combination of hardware and software resources which integrate together and provides various functionalities to the user.	An embedded device is a part of an integrated system that An embedded is formed as an combination of computer hardware and software for a specific function and which can operate without human interaction.
Human Interaction	A computer needs Human Interaction to perform tasks.	Embedded device does not need Human Interaction to perform tasks.
Types based on architecture	Analog computer, Digital computer, Hybrid computer, Harvard architecture, Von Neumann architecture, Reduced instruction set computer	Small Scale Embedded System, Medium Scale Embedded Systems, Sophisticated or Complex Embedded Systems
Parts	It has 2 parts: Hardware and Software.	It has 3 parts: Hardware, Firmware and Software.
Tasks	It can perform many tasks.	It performs limited tasks.

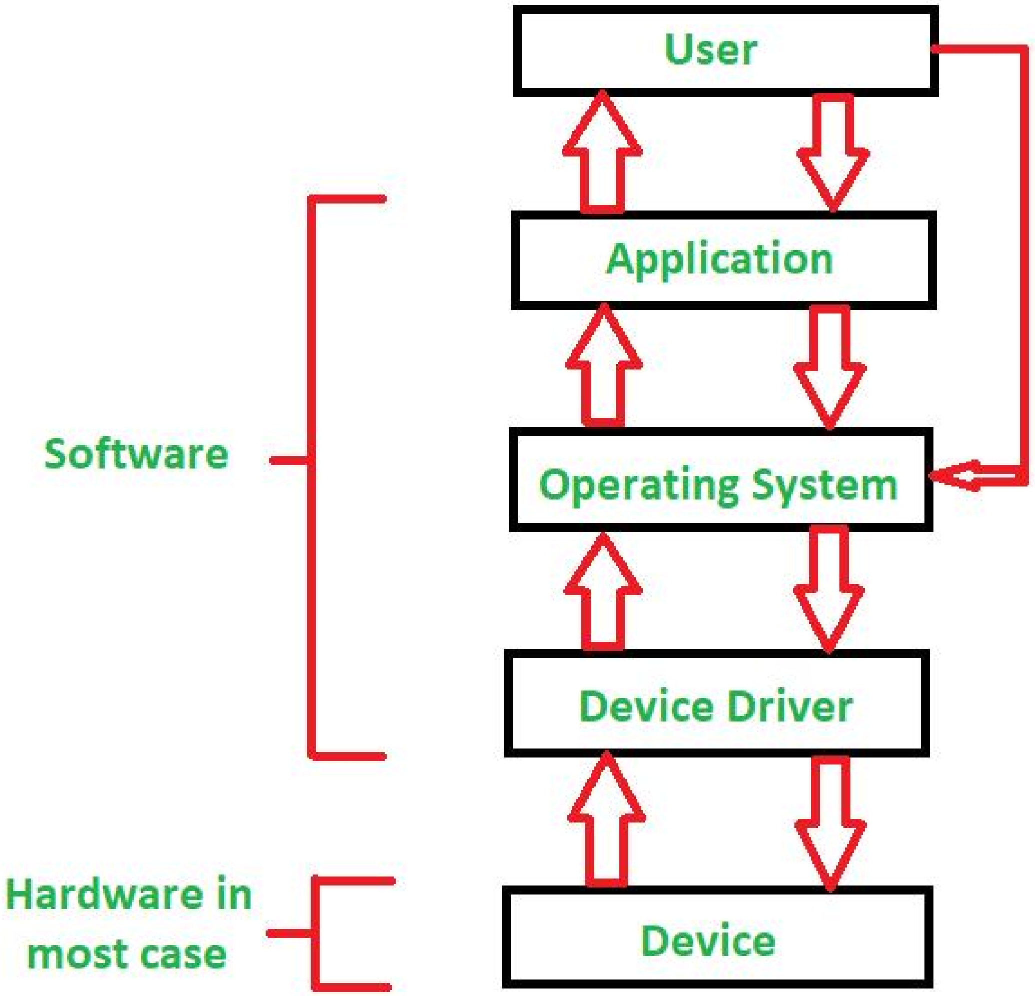
Category	Computer	Embedded device
Cost to user	The user has to pay more for a computer.	The user incurs lesser cost for an embedded system.
Peripherals	Computers have peripherals such as keyboard and mouse, display, printer, Hard disk drives, floppy disk drives, optical disc drives etc.	Embedded Devices have peripherals such as Serial Communication Interfaces (SCI), Synchronous Serial Communication Interface, Universal Serial Bus (USB), Multi Media Cards (SD cards, Compact Flash) etc.
Purpose	Computers can be reprogrammed to for a new purpose.	Embedded Devices are made only for a specific set of purposes.
Power Consumption	Computer needs more operational power than Embedded Devices.	Embedded Device needs lesser operational power than a Computer.
Complexity	Computers are more complex devices than Embedded Devices.	Embedded Devices are less complex devices than Computers.
Need of another device	Computers may be installed in other devices but are self-sufficient to exist.	Embedded Devices only exist inside other Systems.
Usage Difficulty	Computers are more Difficult when used, compared to an Embedded System.	Embedded are easier to use than Computers.
User Interfaces	It requires more user interface than Embedded Devices.	It requires lesser to no user interface than Computers.
Time Specificity	Computers are not time specific. They may need to perform tasks which are not time bound and take days to perform as well.	Embedded Devices are time specific. The tasks assigned to them need to be performed within a specific time frame.
Size	Computers are usually bigger	Embedded Devices are smaller in size

Category	Computer	Embedded device
	in size with larger hardware and input output devices attached to it.	than Computers, with limited hardware.
Developed in	1833 A.D.	1965 A.D.
Developer	Charles Babbage	Charles Stark Draper
Memory Requirement	Computers have larger memory requirement due to a lot of storage of data.	Embedded Devices need lesser Memory.

What is a device driver?

A device driver is a special kind of software program that controls a specific hardware device attached to a computer. Device drivers are essential for a computer to work properly.

These programs may be compact, but they provide the all-important means for a computer to interact with hardware, for everything from mouse, keyboard and display -- user input/output -- to working with networks, storage and graphics.



Comparison between the Real-Time and General Purpose Operating System

Here, you will learn the head-to-head comparison between the Real-Time and General Purpose operating Systems. Some of the head-to-head comparisons between Real-Time and General Purpose Operating Systems are as follows:

Real-Time Operating System	General Purpose Operating System
The RTOS always uses priority-based scheduling.	Task scheduling in a GPOS isn't necessarily based on which application or process is the most important. Threads and processes are often dispatched using a "fairness" policy.

The time response of the RTOS is deterministic.	The time response of the general-purpose operating system is not deterministic.
A low-priority job in an RTOS would be pre-empted by a high-priority one if required, even executing a kernel call.	A high-priority thread in a GPOS cannot preempt a kernel call.
The real-time operating system optimizes memory resources.	The GPOS does not optimize the memory resources.
The RTOS is mainly used in the embedded system.	GPOS is mainly used in PC, servers, tablets, and mobile phones.
The real-time operating system has a task deadline.	The general-purpose operating system has no task deadline.
It doesn't have large memory.	It has a large memory.
GPOS code is not often modular in nature when it comes to development.	RTOS kernel code is intended to be scalable, allowing developers to selectively select kernel objects.
RTOS is designed and developed for a single-user environment.	GPOS is designed for a multi-user environment.
Examples: FreeRTOS, Contiki source code, etc.	Examples: Linux, Windows, IOS, etc.