19/02/2022

IS RTOS an OS?

A Real-time operating system (RTOS) is an operating system (OS) for real time applications that process data and events that have critically defined time constraints.

→ What makes an OS a RTOS?

RTOS is an OS with two key features:
↳ Predictability
↳ determinism

In an RTOS, repeated tasks are performed with in a tight time boundary, while in a general-purpose operating system, this is not necessarily so.

→ Microsoft windows, MacOS, Unix, and Linux are not 'real-time'. They are often completely unresponsive for seconds at a time.

→ RTOS are OS that will always respond to an event in a guaranteed amount of time, not in seconds or milliseconds, but in micro or nanoseconds.

→ Types of RTOS:- monolithic & Microkernel
3 types ⇒ hard time, soft time, firm time.

→ OS:- An OS is a software programme required to manage and operate a computing device like smartphones, tablets, computers, super computers, webservers, cars, network towers, smartwatches, etc; It is a layer of graphical user interface (GUI), which acts as a platform between the User and the computer hardware.

Most significant difference between the two is in how they approach ~~to~~ each task.

OS	RTOS
→ process execution is based on time sharing.	processes are executed based on order of their property.
→ OS is a program that enables user communicates with a hardware.	RTOS is multitasking operating system designed for real applications.
→ Memory allocation in ~~op~~ OS is not that critical compared to RTOS	Memory allocation is more critical in RTOS than any other ~~ope~~ OS.
→ Ex:- Microsoft windows, Linux/Unix, Mac OSX.	Ex:- QNX, RTAI, Symbian OS, Windows CE, Vx works.
→ Applications:- Systems for home, office and data centres.	→ Applications:- Industrial robots, scientific research equipment.
→ time constrained processes.	→ yields results immediately

Summary:-

1. A regular OS focusses on computing throughout while an RTOS focuses on very fast response time.
2. OSes are used in a wide variety of applications while RTOSes are generally embedded in devices that require real time response
3. Os. use a time sharing design to allow for multitasking while RTOSes either Use a timesharing design or an event driven design
4. The coding of an RTOS is stricter compared to a Standard OS.

→ In RTOS, code needs to perform consistently all the time.

→ Standard OSes are not that concerned since response time is not of great importance in its application.

OS ⟶ time sharing

↳ due to fast switching – it appears as real time to users.

RTOS → uses time sharing or event driven design with lower density that processor not to get Loaded.

→ Components:-

↳ Scheduler

↳ Symmetric Multi processing

↳ Function library

↳ Memory management

↳ Fast dispatch latency

↳ User defined data objects and classes.

→ Biggest drawback of RTOS is that the system only concentrates on a few tasks.