①

⇒ General purpose systems are versatile, they aren't always fully optimised to perform specified tasks.

⇒ Embedded systems are designed to perform a small number of tasks efficiently. An example of an embedded system is a placemaker, a small device placed inside a person that monitors and regulates their heartbeat.

parameters	General purpose computer	Embedded system
1. purpose	* multipurpose	* single functioned.
2. Size of system	* Big	* small
3. power consume	* more	* Less
4. cost of system	* costly	* cheap
5. Memory	* Higher memory	* lower memory
6. performance	* fast & better performance	* fixed runtime required
7 user interface	* keyboard, display mouse, touch screen	* Button, sensors, (gas IR)

2) what are device drivers :-

Sol:- 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.

Eg:- A printer driver tells the printer in which format to print after getting instruction from os. Similarly, A sound card drive is there due to which 1's and 0's data of the MP3 file is converted to audio signals and you enjoy the music.

Purpose

The main purpose of device drivers to provide abstraction by acting as a translator between a hardware device and the applications or operating systems that use it.

① firstly all the code the user write is translated into a set of 1's and 0's by a complier.

→ All the computer understands is "high" and "low" voltages, or 1's and 0's.

→ Each instruction generated by the complier is excetued in a cycle.

→ First the hardware accesses the memory to retrive an Instruction.

→ In computer doesn't actually learn the language, instead it parses the language and does what the lines of code tell it to do.

→ To break it down further, the cpu is at the heart of the computer,

→ It only understands something called machine code, which is language consisting of ones and zeros.

① Difference between os and RTOS.

⇒ In general, an operating system is responsible for managing the hardware resources of a computer and hosting applications that run on the computer

⇒ In RTOS performs these tasks, but is also specially designed to run applications with very precise timing and a high degree of reliability.

Complexity

⇒ In RTOS Light weight and designed for minimal overhead and reduced complexity.

⇒ In os more complex supporting a wide variety of applications and hardware configurations.

Examples

⇒ RTOS → FreeRTOS, Vxworks, QNX, ecos.
os → windows, macos, Linux and Unix.

Cost:-

⇒ RTOS → High cost
⇒ OS → low cost

Determinism:-

→ RTOS - Deterministic execution with guranteed timing and deadlines.
→ OS → primarily focus on multitasking and resource sharing.