

Assignment - 1

① What is the difference between General purpose system and embedded system?

Ans The difference between General purpose system and embedded system as follows:

General purpose system	Embedded system.
<ul style="list-style-type: none"> <li>→ They are n't always fully optimised to perform specific tasks</li> <li>→ General purpose system can perform multiple tasks</li> <li>→ Programmable by end users.</li> <li>→ Generally <u>high power consumption</u> required by general purpose system.</li> <li>→ Contain a general purpose operating system (GPOS)</li> <li>→ A system which is combination of generic hardware and General purpose operating system for executing a variety of applications.</li> <li>→ Response time is not critical</li> <li>→ performance is the key deciding factor in the selection of the system. faster is better.</li> <li>→ laptops, uses General purpose system</li> </ul>	<ul style="list-style-type: none"> <li>→ Embedded systems are designed to perform a small number of tasks efficiently.</li> <li>→ Embedded system is designed to do some specific set of tasks.</li> <li>→ Not programmable by end users</li> <li>→ Generally <u>Low power consumption</u> is required by embedded system.</li> <li>→ may or may not contain an operating system for functioning</li> <li>→ A system which is combination of special purpose hardware and embedded OS for execution a specific set of applications.</li> <li>→ Response time is critical for some applications.</li> <li>→ Application Specific requirements (like performance, power requirements, memory usage, etc) are key deciding factors.</li> <li>→ Embedded system used in MP3 players, video game consoles, mobile phones, digital cameras, etc.</li> </ul>



## ② what are device drivers?

Ans A device driver is a piece of software that allows your computer's operating system to communicate with a hardware device the driver is written for.

- Generally a driver communicates with a device through computer bus, which connects the device with the computer.
- Device drivers work within the kernel layer of the operating system.
- The kernel is the part of operating system that directly interacts with the system's physical structure.

### \* Types of device drivers:-

- 1) kernel mode device drivers
- 2) Usermode device drivers.

#### 1) kernel mode device drivers:-

This kernel mode device driver includes some generic hardware that loads with an operating system as part of the OS.

- These are BIOS, motherboard, processor, and some other hardware that are part of kernel software.
- These include the minimum system requirement device drivers for each operating system.

### \* BIOS:-

BIOS (Basic input/output system) is the most basic computer driver in existence.

- It is designed to be the first program that boots when a PC turns on.
- The BIOS is stored on memory built into the motherboard and is designed to boot the hardware connected to the PC, including the hard drives, video display output, keyboard and mouse.

### \* Motherboard drivers:-

motherboard drivers are small programs that are read by either windows or Linux and allow, for basic computer functions while inside the operating system.



- These drivers normally include programs that allow broadband ports, USB ports and I/O ports for the mouse and keyboard.
- Depending on the making of the motherboard, the drivers may also have basic drivers for video and audio support.

## 2) User mode device drivers:-

Other than the devices brought by the kernel for working of the system, the user also brings some devices for use during the using of a system that devices need device drivers to functions those drivers fall under user mode device drivers.

For example, the user needs only plug and play action that comes under this.

## \* Applications of device drivers:-

Because of the diversity of modern hardware and operating systems, drivers operate in many different environments.

- Device drivers may interface with printers, video adapters, Network cards, sound cards, Local buses of various sorts, image scanners, digital cameras.
- Also interface with computer storage devices such as hard disk, CD ROM and floppy disk buses.

## ③ How can hardware understands the codes that we write in embedded system?

Ans:- Firstly all the code the user write is translated into a set of 1's and 0's by a compiler.

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

→ each instruction generated by the compiler is executed in a cycle.

→ only the computer understands the machine language.

→ First the hardware accesses the memory to retrieve an instruction.

→ The computer doesn't actually learn the language, instead it passes 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 a language consisting of ones and zeros.

Q) What is the difference between RTOS and General purpose OS?

Ans- The difference between Real time operating system and General purpose operating system as follows:-

RTOS	GPOS
<ul style="list-style-type: none"> <li>→ RTOS: Real time operating system</li> <li>→ The RTOS always uses priority based scheduling</li> <li>→ The time response of the RTOS is deterministic</li> <li>→ The RTOS optimizes memory resources</li> <li>→ The RTOS is mainly used in the embedded system</li> <li>→ The RTOS has a task deadline</li> <li>→ It doesn't have large memory</li> <li>→ RTOS is designed and developed for a single user environment.</li> <li>→ <u>Examples:-</u> Free RTOS, Contiki source code etc,...</li> </ul>	<ul style="list-style-type: none"> <li>→ GPOS:- General purpose operating system.</li> <li>→ Tasks scheduling in a GPOS isn't necessarily based on which application or process is the most important.</li> <li>→ The time response of the GPOS is not deterministic.</li> <li>→ The GPOS does not optimize the memory resources.</li> <li>→ GPOS is mainly used in PC, servers, tablets, and mobile phones.</li> <li>→ The GPOS has no task deadline.</li> <li>→ It has large memory.</li> <li>→ GPOS is designed for a multi user environment</li> <li>→ <u>Examples:-</u> Linux, windows, IOS etc,...</li> </ul>



⑤ How .c files are converted into .exe files?

Ans → C files first go through the processor, then compiler compiles it into Assembly and creates objects file (main.o). Then linker links the main.o with required header objects and libraries and creates an executable file (program.exe).

→ To convert a .c file into a .exe file, we need to compile it using a C compiler.

→ The compiler translates the human-readable code in the .c file into machine-readable instructions in the .exe file.