

EMBEDDED SYSTEM AND INTERNET OF THINGS

RESEARCH WORK-1

1.What is the boot process of a computer and what is a BIOS?

The boot process is the series of steps that a computer goes through when it is turned on or restarted. The process starts when computer's power button is pressed, and it goes through several stages before it is ready to be used.

1. Power On: The computer is turned on by pressing the power button.
2. BIOS (Basic Input/Output System) Initialization: The BIOS is responsible for initializing the hardware components of the computer, including the CPU, memory, and storage devices.
3. POST (Power-On Self Test): The POST is a diagnostic process that verifies the functionality of various components such as the CPU, RAM, and storage devices.
4. Boot Loader: Once the POST is complete, the boot loader program is loaded from the storage device. The boot loader is responsible for loading the operating system into memory.
5. Operating System Initialization: Once the boot loader has loaded the operating system, the system initializes the operating system, which involves loading drivers, initializing services, and starting system processes.
6. User Login: Finally, the user is prompted to log in to the system, and the desktop or user interface is loaded.

BIOS stands for Basic Input/Output System. It is a firmware that is built into a computer's motherboard and is responsible for initializing and controlling the hardware during the boot process. When you turn on your computer, the BIOS performs a series of tests to ensure that all of the hardware components are functioning properly. It then loads the operating system into memory and hands over control to the operating system.

2.How operating system works? List down 5 tasks of operating system.

An operating system (OS) is a software program that manages computer hardware and software resources and provides common services for computer programs. Here are five tasks that an operating system typically performs:

Memory Management: The operating system manages a computer's memory, allocating memory to programs as needed and ensuring that different programs do not interfere with each other's memory space.

Process Management: The operating system manages the execution of programs, or processes, on the computer.

Input/Output (I/O) Management: The operating system manages the flow of data between the computer and its input/output devices, such as keyboards, mice, printers, and storage devices.

File Management: The operating system manages the creation, deletion, and organization of files on a computer's storage devices

Security Management: The operating system provides a range of security features to protect a computer from unauthorized access and malware.

3.What is real time operating system?

A real-time operating system (RTOS) is an operating system designed for applications that require a high level of reliability and predictability in their response times. An RTOS is characterized by its ability to provide a deterministic response to events, typically within a specific time frame.

In an RTOS, tasks are typically prioritized based on their importance and urgency, and the operating system schedules tasks according to their priorities. This ensures that the most critical tasks are executed first and that deadlines are met. Real-time operating systems are commonly used in embedded systems, such as medical devices, avionics, automotive systems, and industrial control systems. These applications require precise timing and reliable operation, and an RTOS can provide the necessary level of control and predictability.

4.What is bootloader and how does it works?

A bootloader is a small program that is executed when a computer or device is turned on or restarted. Its main purpose is to load and start the operating system (OS) that is installed on the device.

Working:

- ✓ The computer's BIOS or UEFI firmware loads the bootloader from storage device into memory.
- ✓ Initializes the hardware and perform any necessary checks to ensure is functioning properly.
- ✓ The bootloader locates the operating system on storage device and loads it into memory.
- ✓ The bootloader transfer control to operating system, allowing it to take over and begin its execution.

5.Difference between Baremetal and RTOS.

Functionality: Baremetal systems are generally used for simple, low-level tasks such as controlling hardware peripherals or implementing low-level algorithms. RTOS, on the other hand, provide a complete set of services and functions for building complex, multi-tasking applications.

Determinism: RTOS are designed to provide deterministic performance, meaning that they can guarantee that a certain task will be completed within a specific time frame.

Memory Management: RTOS typically provide memory management features such as memory protection and virtual memory, which can help prevent memory-related errors. Baremetal systems do not include these features.

Multitasking: RTOS are designed to support multitasking, which means that multiple tasks can run concurrently on a single processor. Baremetal systems typically do not support multitasking, meaning that only one task can run at a time.

Complexity: RTOS can be more complex than baremetal systems, as they provide a larger set of features and functions. This can make development more challenging, as developers need to understand the RTOS and its various features in order to use it effectively

6.How to choose between baremetal and RTOS for project?

Define the project requirements: Consider the project's functional and non-functional requirements, including performance, power consumption, memory usage, and required peripherals.

Evaluate available resources: Consider the available hardware resources, including the processor, memory, and storage. Evaluate whether the available resources are sufficient for the project requirements.

Consider developer experience: Consider the developer's experience and expertise with baremetal systems and RTOS. Determine if the developer has the necessary skills and knowledge to develop software for the selected platform.

Evaluate development time: Consider the development time required for each option. Baremetal systems are generally quicker to develop as they do not require a full operating system.

Evaluate project cost: Consider the cost of each option, including hardware, software, and development costs. Baremetal systems may be less expensive as they require less hardware and software resources.

Consider long-term maintenance: Consider the long-term maintenance and support requirements of the project. RTOS may provide better long-term support and maintenance as they are designed to be more scalable and maintainable.