Research Work 1

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

→Booting is basically the process of starting the computer. When the CPU is first switched on it has nothing inside the Memory. In order to start the computer, load the Operating System into the Main Memory and then Computer is ready to take commands from the User.

Booting happens when you start the computer. This happens when we turned ON the power or the computer restarts. The system BIOS (Basic Input/Output System) makes the peripheral devices active. Further, it requires that the boot device loads the operating system into the main memory.

- The boot process of a computer involves the following steps:
- Power on: When the power button is pressed, the computer starts up.
- BIOS: The Basic Input/Output System (BIOS) checks the computer's hardware and starts the boot process.
- MBR: The Master Boot Record (MBR) is loaded, which contains information about the location of the operating system and its boot loader.
- Boot loader: The boot loader, such as GRUB or Windows Boot Manager, is executed, which then loads the operating system.
- Operating System: The operating system is loaded into memory and starts its initialization process.
- Login: The user logs in and the operating system starts up the necessary services and applications.
- Desktop: The desktop is displayed, and the user can start using the computer.

History of BIOS

- The term BIOS was first coined in 1975 by American computer scientist Gary Kildall. It was incorporated into IBM's first personal computer in 1981 and, in the years to come, gained popularity within other PCs, becoming an integral part of computers for some time.
- BIOS stands for Basic Input/Output System. It is a low-level software that is stored on a chip on the motherboard of a computer. The BIOS is responsible for performing several important tasks during the boot process

Uses of BIOS

The main use of BIOS is to act as a middleman between OSes and the hardware they run on. BIOS is theoretically always the intermediary between the microprocessor and I/O device control information and data flow. Although, in some cases, BIOS can arrange for data to flow directly to memory from devices, such as video cards, that require faster data flow to be effective.

01-02-2023

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

➔ An **operating system** (**OS**) is system software that manages computer hardware, software resources, and provide common services for computer programs.

An operating system is software that enables applications to interact with a computer's hardware. The software that contains the core components of the operating system is called the kernel.

The primary purposes of an Operating System are to enable applications (software) to interact with a computer's hardware and to manage a system's hardware and software resources.

Important functions of an operating System:

1. **Security –**
   The operating system uses password protection to protect user data and similar other techniques. it also prevents unauthorized access to programs and user data.

2. **Control over system performance –**
   Monitors overall system health to help improve performance. records the response time between service requests and system response to having a complete view of the system health. This can help improve performance by providing important information needed to troubleshoot problems.

3. **Job accounting –**
   Operating system Keeps track of time and resources used by various tasks and users, this information can be used to track resource usage for a particular user or group of users.

4. **Error detecting aids –**
   The operating system constantly monitors the system to detect errors and avoid the malfunctioning of a computer system.

5. **Coordination between other software and users –**
   Operating systems also coordinate and assign interpreters, compilers, assemblers, and other software to the various users of the computer systems.

3. What are the single board computers (SBC)?

➔ A single board computer (SBC) is a complete computer system that is built on a single circuit board. It integrates all the necessary components, including a CPU, memory, storage, and various I/O interfaces, onto a single board. Single board computers are typically smaller in size and lower in cost compared to traditional desktop computers, making them popular for a variety of applications, including hobbyist projects, industrial automation, and Internet of Things (IoT) devices.

List of single board computer:

1. Arduino mega 2560
2. Orange Pi One Plus
3. Banana Pi P2 Zero

4. Which SoC is used in Raspberry pi, Beagle board black, Banana Pi, Jetson Nano, Coral Dev Board.

→

- **The Raspberry Pi 4** uses a Broadcom BCM2711 SoC with a 1.5 GHz (later models: 1.8 GHz) 64-bit quad-core ARM Cortex-A72 processor, with 1 MB shared L2 cache.
- The Raspberry Pi Zero and Zero W use the same Broadcom BCM2835 SoC as the first generation Raspberry Pi, although now running at 1 GHz CPU clock speed.

- **Beagle Board Black :** Launched on April 23, 2013 , The BeagleBone Black uses an AM335x Sitara System-on-Chip (SoC) manufactured by Texas Instruments. The AM335x is a low-power, high-performance ARM Cortex-A8 based SoC designed specifically for embedded applications. It includes a 700 MHz ARM Cortex-A8 processor, 3D graphics accelerator, PRU-ICSS for programmable real-time control, and a variety of peripherals such as Ethernet, USB, and more.

- **Banana Pi** - **Banana Pi** is a line of single-board computers produced by the Chinese company Shenzhen SINOVOIP Co., Ltd. , its spin-off Guangdong BiPai Technology Co., Ltd. and supported by Hon Hai Technology (Foxconn).

- The hardware design of the Banana Pi computer model was influenced by the Raspberry Pi and both lines use the same 40 pin I/O connector.

- Banana Pi also can run NetBSD, Android, Ubuntu, Debian, Arch Linux, Raspberry Pi OS operating systems,
- The **Banana Pi BPI-M1** is a single-board computer featuring a Allwinner dual-core SoC at 1 GHz, 1GB of DDR3 SDRAM, Gigabit Ethernet, SATA, USB, and HDMI connections, and built-in 3.7V Li-ion battery charging circuit.

Key Features:

- Allwinner A20 Dual-core 1.0 GHz CPU
- Mali-400 MP2 with Open GL ES 2.0/1.1.
- 1 GB DDR3 memory.
- 1x SATA interface.
- 1x Gigabit LAN
- 1x USB otg and 2x USB 2.0
- 1X MIC
- Composite video out
- HDMI out
- IR
- CSI camera interface
- DSI display interface
- 26 PIN GPIO

- **Jetson Nano -** **Nvidia Jetson** is a series of embedded computing boards from Nvidia. The Jetson TK1, TX1 and TX2 models all carry a Tegra processor (or SoC) from Nvidia that integrates an ARM architecture central processing unit (CPU) which includes a quad-core ARM Cortex-A57 CPU and a NVIDIA Maxwell GPU with 128 CUDA cores. Jetson is a low-power system and is designed for accelerating machine learning applications.

- It supports various AI frameworks such as TensorFlow, PyTorch, and Caffe, and is designed for use in edge computing applications such as computer vision, robotics, and autonomous machines.

- **Coral Dev Board :**

  The Google Coral Dev Board is a single-board computer that uses the NXP i.MX 8M System-on-Chip (SoC). The NXP i.MX 8M is a quad-core ARM Cortex-A53 processor with a clock speed of up to 1.5GHz, combined with a Vivante GC7000 Lite GPU.

- The SoC also includes 4GB of LPDDR4 memory, eMMC storage, Gigabit Ethernet, Wi-Fi, Bluetooth, and various other peripheral interfaces such as USB, MIPI-DSI, and MIPI-CSI. The Coral Dev Board is designed for use in edge computing applications and supports the TensorFlow Lite framework for machine learning and computer vision tasks.

- Overall, the NXP i.MX 8M SoC provides high computational performance and low power consumption, making the Coral Dev Board a suitable solution for building AI-powered devices at the edge.

5.What is Real time operating system.

→An RTOS or real-time operating system is a special-purpose OS for computers that must accomplish tasks within severe time limitations. It is primarily used in systems where the results of calculations are used to impact a process as it is running. When an event occurs outside of the computer, it is conveyed to the computer via a sensor that is used to monitor the event. The sensor generates a signal that the OS interprets as an interrupt. When the

operating system receives an interrupt, it starts a specific process or a collection of processes to handle it.

All real-time operating systems are built to complete their tasks in a specific amount of time, so they must be quick enough to meet their deadline. Time restrictions in real-time systems simply refer to the time interval provided for the continuing program's reaction. This deadline indicates that the task must be performed within the specified time frame.

Types of Real-time OS

1. Hard RTOS

The term "hard real-time operating system" is not a widely used or well-defined term in the field of operating systems. It is typically used to refer to a real-time operating system that provides a high degree of determinism and reliability for real-time tasks, and is typically used in applications that require a high level of safety and criticality, such as aerospace and defence systems.

A hard real-time operating system is usually designed to provide deterministic response times for real-time tasks, even in the face of high system load and other unpredictable events. To achieve this, a hard real-time operating system typically has a small and simple kernel, with minimal interrupt latency, and predictable scheduling and synchronization mechanisms.

Examples of RTOS that are often considered to be "hard real-time" include VxWorks, RT Linux, and INTEGRITY. These RTOS are designed to provide reliable and predictable response times for real-time tasks, making them suitable for use in high-criticality and safety-critical applications, such as avionics, defence systems, and medical devices.

Example

Consider airbags and a handle in the driver's seat, both given by automakers. When the driver applies the brakes at a certain point, the airbags expand to protect the driver's head from colliding with the steering wheel. There would have been an accident if there had been even a millisecond of delay.

Consider using online stock trading software. If someone wants to sell a specific stock, the system must ensure that the command is carried out within a certain amount of time. Otherwise, if the market drops suddenly, the trader may suffer a significant loss.

2. Soft RTOS

A soft real-time operating system (SRTOS) is a type of RTOS that makes a best-effort attempt to complete real-time tasks within a specified deadline, but does not guarantee that the deadline will always be met. In contrast to a firm real-time operating system, which guarantees that real-time tasks will always be completed within the specified deadline, a SRTOS prioritizes real-time tasks but may allow some jitter or delay in their completion.

An example of a SRTOS is the open-source RTOS, FreeRTOS. FreeRTOS is a compact and scalable RTOS that can be used in a wide range of embedded systems and Internet of Things (IoT) devices. While FreeRTOS does not guarantee the completion of real-time tasks within a specific deadline, it provides deterministic behavior and low latency for real-time tasks, making it suitable for a wide range of real-time applications, including control systems and robotics.

Another example of a SRTOS is the real-time version of the popular open-source operating system, Android. Android for Embedded Systems, also known as Android Things, is a version of Android designed for use in IoT devices and embedded systems. Android Things provides a best-effort real-time performance, allowing real-time tasks to be completed within a specified deadline in most cases, but not guaranteeing it in all situations

Example

Both online transaction systems and livestock price quote systems employ this type of system.

4. Firm RTOS

A firm real-time operating system (FIRTOS) is a type of RTOS that is designed to guarantee the completion of real-time tasks within a specific deadline. In contrast to a soft real-time operating system, which makes a best-effort attempt to complete real-time tasks within the specified deadline, a FIRTOS is designed to guarantee that real-time tasks will always be completed within their deadline, even if the system is under heavy load.

An example of a FIRTOS is RT Linux, which is a real-time version of the Linux operating system. RT Linux provides a deterministic response time for real-time tasks, which makes it suitable for applications that require a high degree of predictability and reliability, such as aerospace, defence, and robotics.

Another example of a FIRTOS is the Real-Time Executive for Multiprocessor Systems (RTEMS), which is an open-source RTOS designed for use in embedded systems and real-time control applications. RTEMS provides deterministic response times and precise timing control, making it an excellent choice for applications that require firm real-time performance, such as aircraft control systems and medical devices.

Example

This mechanism can be found in a variety of multimedia applications.