INTRODUCTION

Andrew's Team

Toumba

Toumba
1. WHAT IS AN OPERATING SYSTEM?

In a database, search, and retrieval system, each operation on a collection of data is performed by a query language. The database management system (DBMS) provides a set of services that allow users to define, manipulate, and retrieve data in a consistent, secure, and efficient manner. The DBMS also includes a user interface to interact with the database, and a set of tools for administration and maintenance.

1.1 THE OPERATING SYSTEM AS AN EXECUTED MACHINE

You hear mostly about one function of the other. Let us now look at both.

The machine manages resources, and the operating system manages resources, but in different ways. An operating system provides services to the machine, while the machine provides resources to the operating system. The machine provides the raw processing power, while the operating system provides a user-friendly interface to the raw processing power.

Most computer users have some experience with an operating system, but few computer users have much experience with a storage system.
171. The more common system calls later in the chapter.

1.7. The Operating System as a Resource Manager

What Is an Operating System?
1.12. The Second Generation (1955–59), Transistors and Batch Systems

In contrast to the vacuum tube systems discussed earlier, the process of designing and manufacturing computers changed significantly with the introduction of transistors. Transistors were simpler and more reliable than vacuum tubes, which made them ideal for use in computer design.

By the early 1950s, the transistor had become a standard component in electronic devices. It was small, reliable, and easy to manufacture, making it ideal for use in computers. Transistors allowed for the design of smaller and more efficient computers, which led to the development of batch systems.

In these early days, a single batch system could process data from many users at the same time. The user would submit a job to the computer, and the computer would process the job in a batch, which meant that the user would have to wait for the entire batch to be processed before receiving their results.

2. HISTORY OF OPERATING SYSTEMS

Operating systems evolved over time, reflecting the changing needs of computer users and the technology available at the time.

Early computer systems were simple and straightforward, with users directly interacting with the hardware. As computers became more complex, the need for a way to manage and control these systems grew.

The development of operating systems was a response to the need for a way to manage and control the hardware. Early operating systems, such as the IBM 1401, included features such as batch processing, which allowed multiple users to submit jobs to the computer simultaneously. These early operating systems were simple and focused on basic functions, such as file management and job scheduling.

As computers became more powerful and complex, operating systems evolved to include features such as multitasking, which allowed multiple programs to run simultaneously. Operating systems also became more sophisticated, with features such as virtual memory and network support.

Today, operating systems are essential components of modern computers, providing a platform for users to run applications and access the underlying hardware. Operating systems have evolved significantly over time, adapting to the changing needs of users and technology.
The primary role of the CPU is to execute programs and perform various tasks. The CPU reads instructions from the memory, processes them, and sends the results back to memory. It contains a control unit and an arithmetic logic unit (ALU).

The control unit fetches instructions from memory, decodes them, and generates control signals to control the execution of the ALU. The ALU performs the actual arithmetic and logical operations based on the instructions.

In this diagram, we can see the components of a simple personal computer:

- **Bus**: Connects various components
- **CPU**: Central Processing Unit
- **Memory**: Stores data and instructions
- **Controller**: Manages data flow
- **Hard Drive**: Stores data
- **CD-ROM Drive**: Reads and writes to CDs
- **Keyboard**: Input device
- **Video Controller**: Handles video output
- **Monitor**: Displays video output

These components work together to process data and perform tasks. The CPU is the brain of the computer, and it's responsible for executing instructions and managing the flow of data between other components.

**Questions for Review**

1. What is the role of the CPU in a computer system?
2. How does the control unit work in the context of the CPU?
3. What are the primary components of a computer system shown in the diagram?
The second microprocessor in my computer is the memory. Really, a memory.

1.4 Memory

I wonder how much can go back to the program to deal with the problem when the problem is an instruction that it wants to eliminate error of control. I'm not sure that the control can be improved (an instruction that can be set to 0). The memory, when the program is an instruction that it wants to eliminate error of control, I'm not sure that the control can be improved (an instruction that can be set to 0). The memory, when the program is an instruction that it wants to eliminate error of control, I'm not sure that the control can be improved (an instruction that can be set to 0). The memory, when the program is an instruction that it wants to eliminate error of control, I'm not sure that the control can be improved (an instruction that can be set to 0). The memory, when the program is an instruction that it wants to eliminate error of control, I'm not sure that the control can be improved (an instruction that can be set to 0). The memory, when the program is an instruction that it wants to eliminate error of control, I'm not sure that the control can be improved (an instruction that can be set to 0). The memory, when the program is an instruction that it wants to eliminate error of control, I'm not sure that the control can be improved (an instruction that can be set to 0).
use CMOS memory to hold the current time and date. The CMOS memory and the real-time clock are a separate component in the motherboard.

1. Introduction

CMOS memory (Complementary Metal-Oxide-Semiconductor) is a type of memory that retains data even when the power is turned off. It is used to store information such as the system time, configuration settings, and security data. CMOS memory is non-volatile, meaning it does not lose data when power is removed. It is implemented using a combination of transistors and capacitors, which work together to maintain the stored data. CMOS memory is typically used in personal computers and other electronic devices to store configuration information.

2. CMOS Memory

CMOS memory uses a combination of transistors and capacitors to store information. The transistors are used to control the flow of current, while the capacitors are used to store the data. CMOS memory is typically implemented using CMOS technology, which is a type of semiconductor technology that is used to manufacture integrated circuits.

3. Function of CMOS Memory

CMOS memory is used to store various types of information, including the system time, configuration settings, and security data. It is an important component in personal computers and other electronic devices, as it allows the system to maintain information even when power is removed.

4. Types of CMOS Memory

There are several types of CMOS memory, including SRAM (Static Random Access Memory), DRAM (Dynamic Random Access Memory), and ESRAM (Enhanced Static Random Access Memory). Each type of memory has its own advantages and disadvantages, and is used in different applications.

5. Conclusion

CMOS memory is a crucial component in personal computers and other electronic devices, as it allows the system to maintain information even when power is removed. It is an important part of the overall system, and its performance and reliability are critical to the overall performance of the system.
Figure 1.9(a): Use of one-page and multi-page memory organization.

- The check and compute result is in computing an address generated by the page control word. This is the memory address of the data to be retrieved.

- The check and compute result is in computing an address generated by the page control word. This is the memory address of the data to be retrieved.

### How to Handle Exception

1. **How to Protect the Programs from One Another and the Kernel from Other Processes:**

   - **Memory:** Each process has a separate memory space. The kernel controls access to memory to prevent processes from accessing each other's memory.

   - **CPU:** Each process has its own CPU registers. The kernel manages these registers to prevent processes from interfering with each other.

   - **File System:** Each process has its own file system. The kernel manages file system access to prevent processes from accessing each other's files.

### Example

1. **Simultaneous Access to Two Files:**

   - **Scenario:** Two processes, A and B, need to access the same file.

   - **Solution:** The kernel uses file locks to prevent both processes from accessing the file simultaneously. Each process must acquire a lock before it can access the file.

2. **Concurrency:**

   - **Scenario:** Multiple processes need to access the same resource, such as a file or a database.

   - **Solution:** The kernel uses locking mechanisms to ensure that only one process can access the resource at a time. This prevents conflicts and ensures that the resource is accessed correctly.

### Conclusion

By understanding the principles of exception handling and memory management, developers can create more robust and secure systems.
The diagram above shows the different layers of the Operating System. The Operating System is the software that allows the computer to interact with the users and other software applications. It manages the resources of the computer, such as memory, processors, and storage devices.

The first layer is the Hardware Abstraction Layer (HAL). This layer provides an interface between the hardware and the Operating System kernel. The HAL hides the details of the hardware from the Operating System, allowing the kernel to interact with the hardware in a uniform way.

The second layer is the Kernel. The Kernel is responsible for managing the system resources and providing services to the application layers. It performs tasks such as process management, memory management, and device management.

The third layer is the Application Layer. This layer provides interfaces for applications to interact with the Operating System. Applications use this layer to access resources such as files, networks, and other system services.

The Operating System Architecture diagram above illustrates these layers and their interactions. Understanding the layers and their responsibilities is crucial for anyone working with Operating Systems.
SCSI. The IDE bus is for attaching peripheral devices such as drives and CD-ROM drives. In addition, the system contains these specialized buses: IDE, USB, and a few others.

Functions in the system involve the connection to the CLP of the device. A key element of these functions is the transfer of data over a dedicated link between the CLP and the device. In this configuration, the CLP acts as the bridge between the local bus and the device, allowing for the transfer of data.

Even some non-RISC-based systems use the PCI bus due to the large number of I/O devices. Most high-speed I/O devices use the PCI bus, either 32-bit or 64-bit, to communicate with the CPU. The PCI bus is used for this purpose.

The PCI (Peripheral Component Interconnect) bus is the interface between the CPU and the system board. It is used for devices such as hard drives, network adapters, and sound cards. The PCI bus allows for high-speed data transfer and is designed to be compatible with a wide range of devices.

In a configuration of this type, a large number of devices can be connected to the system board via the PCI bus. This allows for a flexible and scalable system that can be easily updated as new devices are added. The PCI bus is also used for devices such as modems, network adapters, and game controllers.

1.4.4 Bus

By device type -

The highest priority bus in this system is the PCI bus, which is responsible for the transfer of data between the CPU and the devices connected to it. The PCI bus is a high-speed bus that is used for a variety of purposes, including the transfer of data between the CPU and peripheral devices, the transfer of data between the CPU and memory, and the transfer of data between the CPU and input/output devices.

In summary, the PCI bus is a critical component of the system board that enables the transfer of data between the CPU and a wide range of devices. It is a high-speed, flexible, and scalable bus that is used in many modern systems.

![Diagram of PCI bus](image-url)
INTRODUCTION

The BIOS (Basic Input Output System) is a firmware program that initializes and configures the computer when it is turned on. It performs tasks such as checking hardware components, loading the operating system, and setting up the computer's configuration. The BIOS is stored on a Read-Only Memory (ROM) chip and is executed by the computer's processor upon startup.

The BIOS is responsible for handling low-level device operations and is crucial for the proper functioning of the computer. It provides a bridge between the hardware and the operating system, allowing the system to boot and load necessary drivers and components.

In the context of USB devices, the BIOS plays a significant role. USB devices require specific configurations and settings to be recognized by the computer. The BIOS configuration ensures that USB devices are detected and initialized properly.

For developers and system administrators, understanding the BIOS is essential for troubleshooting and optimizing computer performance. Knowledge of the BIOS can help in identifying hardware issues, configuring custom startup settings, and ensuring compatibility with various devices.

From an architectural perspective, the BIOS is a critical component that facilitates the integration of hardware and software, making it a fundamental aspect of computer system design.

Diagram: A diagram illustrating the BIOS's role in the computer startup process, showing connections and relationships between various system components.
The process is the fundamental unit of execution in a computer system. A process is an instance of a program in execution. Each process in a computer system is associated with a process context, which contains the current state of the process. The context includes the process's memory state, program counter, and other state information.

A process can be in one of several states: running, waiting, or suspended. The running state indicates that the process is currently executing instructions. The waiting state indicates that the process is waiting for some event to occur, such as a disk I/O operation. The suspended state indicates that the process is not currently executing and is not expected to execute in the near future.

A program can create a new process by fork()ing it. When a process fork()s, a new process is created that is a copy of the original process. Each process has its own address space, which allows it to execute its own code and manipulate its own data. The address space of a process includes the process's memory, file descriptors, and environment variables.

The operating system manages the resources available to processes, including memory, disk space, and CPU time. The operating system also provides services to processes, such as file I/O, network communication, and process scheduling.

In a multi-process system, a process can communicate with other processes using system calls. These system calls provide a common interface for all processes to access the operating system services. The operating system manages the execution of processes and ensures that they run in a controlled environment.

The operating system also provides services to support the execution of processes, such as memory management, process scheduling, and process synchronization. These services help to ensure that processes run efficiently and do not interfere with each other.

In summary, the operating system provides a platform for processes to execute and communicate with each other. The operating system manages the resources available to processes and provides services to support the execution of processes.
1.5.3 Memory Management

When can we be sure that a process is not competing for memory? We will study deadlocks and their prevention in Chap. 3. We will study the process's memory management techniques in Chap. 4. The operating system must ensure that processes do not access memory locations that are not part of their virtual address space. The virtual address space is the part of the memory that a process can access. The operating system manages the virtual address space and provides a mapping between the virtual address space and the physical memory. This mapping is called the page table. The page table is stored in memory and is managed by the operating system. When a process needs to access a memory location, it is responsible for finding the corresponding page in the physical memory. If the page is not already in memory, it must be brought in from secondary storage. If the page is already in memory, the process can access it directly. If the page is not found in the page table, it must be brought in from secondary storage. Once the page is in memory, the process can access it directly. The operating system manages the memory allocation and deallocation, ensuring that the memory is used efficiently and fairly among all processes.