CLASSIFICATION AND FUNCTIONS OF OPEARATING SYSTEMS

CLASSIFICATION AND FUNCTIONNS OF OPERATING SYSTEMS

Operating systems are essential software components that manage computer hardware and provide common services for computer programs. Operating systems(OS) act as intermediaries between the hardware and software of the Computer, facilitating communication and management of resources.

Operating systems are classified based on their structure, their functionality and usage. The primary functions of an operating system include; Resource management, memory management, process management, file System management, Device management and user interface.

CLASSIFICATION OF OPERATING SYSTEMS

1. **SINGLE USER VS MULTI-USER OS**

Single user operating systems are designed for individual users, they only allow one user to access the system at a time. They are intended for use on devices like wireless phones and two way messaging devise where only one user interacts with the system. They can handle tasks like memory management, hardware connectivity and running applications for single users.

On the other hand multi user operating systems allows multiple users to access a single machine simultaneously thus enabling shared access to resources like hardware and software. Different individuals can have different accounts and access the system at the same time each with their own settings and access permissions, These systems are commonly used in network environments and servers where multiple people need to access shared resources.

2. **REAL-TIME OPERATING SYSTEMS**

A real time operating system (RTOS) s an operating system for real time computing applications that process data and events that have critically defined time constraints. It is crucial for applications where timely and deterministic responses are required, such applications include; Industrial automation, robotics, medical devices and aerospace systems.

Real-Time operating systems ensure that tasks are completed within a specified deadline hence providing predictability and reliability in time critical applications, they are categorized into two, based on their ability to meet deadlines, that is;

a) Hard Real-Time Operating systems; Taks have strict deadlines that must be met, failure to meet the deadlines can result in system failure, examples include; airbag deployment in cars, medical devices and missile guidance systems.
b) Soft Real-Time Operating systems; Soft real time operating systems also have deadlines but missing a deadline does not lead to system failure, These systems prioritize tasks based on theor importance and attempt to meet deadlines.

3. DISTRIBUED OPERATING SYSTEMS

Distributed operating system is a type of operating system that runs on multiple computers and coordinates their activities. Each computer in a distributed operating system is called a node, and they work together to provide a unified computing environment. This allows for better resource utilization, improved performance.

One of the key features of a distributed operating system is transparency, which means that the system appears as a single, unified entity to its users. This transparency can be achieved in several different ways, including location transparency, which allows users to access resources without needing to know where they are physically located, and access transparency, which allows users to access resources using the same methods regardless of their location.

Another important feature of distributed operating systems is fault tolerance. By distributing tasks across multiple nodes, the system can continue to function even if one or more nodes fail.

Security is also a major concern in distributed operating systems. Because resources are shared across multiple nodes, it is important to ensure that access is restricted only to authorized users and that data is protected from unauthorized access or modification.

There are many different types of distributed operating systems, each with its own strengths and weaknesses. Some examples include network operating systems, which are designed to run on multiple computers connected by a network, and distributed operating systems for cloud computing, which are designed to run on virtualized hardware in a data center.

4. Batch processing vs. Interactive Operating System.

Batch processing involves the execution of a series of jobs in a non-interactive manner, where the tasks are queued up and processed one after the other without user intervention. This approach is commonly used in scenarios where large volumes of data need to be processed efficiently, such as in payroll processing, billing systems, and report generation. Batch processing is well-suited for repetitive, time-consuming tasks that can be automated and scheduled to run at off-peak hours to optimize system resources.

On the other hand, interactive operating systems enable real-time user interaction with the computer system. Users can input commands, receive immediate feedback, and interact with applications in a dynamic and responsive manner. Interactive OS is essential for tasks that require user input, such as gaming, web browsing, graphic design, and real-time data analysis. This type of operating system provides a platform for users to interact with the computer system in a flexible and intuitive way.

One of the key differences between batch processing and interactive OS is the level of user involvement. In batch processing, users typically do not interact with the system during job execution, whereas interactive OS relies heavily on user input and feedback. Additionally, batch processing is designed for handling large-scale data processing tasks efficiently, while interactive OS prioritizes real-time user interaction and responsiveness.

5. EMBEDED OPERATING SYSTEM

Embedded operating systems are specialized operating systems designed to perform specific functions within a larger system. These operating systems are embedded directly into the firmware of a device and are tailored to the requirements of the specific hardware and application. They are commonly found in devices such as smartphones, digital cameras, medical devices, and automotive systems.

One of the key characteristics of embedded operating systems is their ability to operate within the constraints of the hardware on which they are installed. This means that they are often designed to be lightweight and efficient, with minimal resource requirements. They are also typically optimized for real-time processing, as many embedded systems require immediate responses to external stimuli.

Another important feature of embedded operating systems is their reliability and stability. Since many embedded systems are used in critical applications such as medical devices or automotive control systems, it is essential that the operating system is robust and can operate continuously without failure.

Embedded operating systems can be categorized into two main types: **Real-time operating systems (**RTOS) and **General-purpose operating systems.** RTOS are designed for applications that require precise timing and quick response times, such as industrial control systems and robotics. General-purpose operating systems, on the other hand, are more versatile and can be used in a wider range of applications, such as consumer electronics and networking equipment

FUNCTIONS OF OPERATING SYSTEMS

1. Process Management:

Operating systems manage processes, which are instances of executing programs. They allocate system resources to processes, such as CPU time, memory, and input/output devices. The operating system also provides mechanisms for process synchronization and communication**.**

2. Memory Management:

Operating systems manage a computer's memory, ensuring that each process has enough memory to execute and that memory is used efficiently. This includes allocating and deallocating memory, swapping processes in and out of memory, and implementing virtual memory**.**

3. File System Management:

Operating systems manage files on storage devices, providing a file system that organizes and stores data. This includes creating, deleting, and modifying files, as well as controlling access to files through file permissions.

4. Device Management:

Operating systems manage input/output devices such as keyboards, mice, printers, and disk drives. They provide device drivers to communicate with hardware devices and handle input/output requests from processes**.**

5. Security and Protection:

Operating systems enforce security measures to protect the computer system from unauthorized access and malicious software. This includes user authentication, access control, and data encryption**.**

6. User Interface:

Operating systems provide a user interface that allows users to interact with the computer. This can be a command-line interface, a graphical user interface, or a combination of both**.**

7. Networking:

Operating systems support networking capabilities, allowing computers to communicate with each other over a network. This includes managing network connections, protocols, and network configurations**.**

8. Error Handling:

Operating systems handle errors and exceptions that occur during the operation of the computer system. They provide mechanisms for error detection, reporting, and recovery**.**

REFERENCE

Silberschatz, P.B. Galvin, G. Gagne. Operating Systems Concepts, Sixth edition, Addison Wesley, 2001.

S. Tanenbaum. Distributed Operating Systems, Prentice-Hall, 1995.

S. TanenbaumModern Operating Systems, Prentice-Hall, 19924.

.