



Question & Answers

INTRODUCTION

Sercan Külcü | Operating Systems | 10.04.2023

Contents

What is an operating system?.....	3
Operating system's role in a computer system.....	3
What are some examples of popular operating systems?	3
What is the function of an operating system's kernel?.....	4
How does an OS manage resources?.....	4
Main purpose of an operating system?	5
What is kernel?	5
What are real-time systems?.....	6
What is the time-sharing system?	6
What is caching?.....	6
What is spooling?	7
What is GUI?.....	7
Difference between process and program?.....	8
What is a dispatcher?	8
Define the term dispatch latency?.....	9
Associative Memory and Cache memory?	9
What is "Locality of reference"?.....	10
How does an operating system abstract the underlying hardware from the applications that run on it?	10
How do operating systems provide security mechanisms such as access control and authentication?.....	11
How do operating systems handle interrupts and exceptions?	11
What is virtual memory, and how does it work in an operating system?	11

What are the different types of operating system architectures, and how do they differ? 12

How do real-time operating systems differ from general-purpose operating systems? 12

Some of the challenges in designing an operating system for a massively parallel computer system? 13

How does an operating system handle distributed file systems and distributed computing?..... 13

What is the role of microkernels in operating system design, and what are some advantages and disadvantages of this approach? 13

What are some emerging trends and technologies in operating system design and implementation, and how are they likely to impact the future of computing? 14

What is an operating system?

An operating system (OS) is a software program that manages a computer's hardware and provides a platform for other software applications to run on. It serves as an intermediary between the user and the hardware, providing a user-friendly interface to the underlying hardware components.

Operating system's role in a computer system

The primary role of an operating system is to manage and coordinate the activities and sharing of resources of a computer system. It provides services such as memory management, process and thread management, device management, file management, security management, and networking.

What are some examples of popular operating systems?

Some examples of popular operating systems are:

- Windows (Microsoft)
- macOS (Apple)
- Linux (Various distributors)
- Android (Google)
- iOS (Apple)
- Solaris (Oracle)
- FreeBSD (Open source community)

What is the function of an operating system's kernel?

The kernel is the core component of an operating system. Its main function is to manage the computer's hardware and provide a bridge between the software and the hardware. The kernel provides low-level services such as memory management, process management, device management, and CPU scheduling.

How does an OS manage resources?

Operating systems manage resources such as memory and CPU through various mechanisms such as process scheduling, memory allocation, and virtual memory. The operating system uses algorithms and policies to allocate resources to different processes and applications based on their priority, usage patterns, and availability. For example, CPU scheduling algorithms decide which process should be given access to the CPU at a given time, while memory management algorithms decide how to allocate memory to different applications and processes to ensure optimal usage and prevent memory-related errors. Virtual memory enables the operating system to provide a larger address space to applications than is physically available by using disk storage as an extension of main memory.

Main purpose of an operating system?

An operating system is a software that functions as an intermediary between the computer hardware and its user. Its primary role is to create an environment where users can execute programs in a convenient and efficient manner. The operating system manages the resources of a computer, such as the processor, memory, and input/output devices, to ensure that the programs run smoothly without interfering with each other. It also provides users with a graphical user interface (GUI) that enables them to interact with the computer and perform various tasks like running applications, accessing files, and configuring system settings. The overall goal of an operating system is to make computing easier and more productive for users by providing a seamless and intuitive computing experience.

What is kernel?

The kernel is the crucial component of an operating system that oversees the operations of the computer and its hardware. It serves to manage the operations of memory and CPU time, making it a fundamental element of the operating system. Acting as a bridge between applications and the hardware level data processing, the kernel facilitates inter-process communication and system calls. Its primary role is to serve as the intermediary between the software and the hardware, providing a platform for software applications to communicate with the hardware and manage system resources efficiently.

What are real-time systems?

A real-time system means that the system is subjected to real-time, i.e., the response should be guaranteed within a specified timing constraint or the system should meet the specified deadline.

What is the time-sharing system?

Time-sharing is a natural progression of multiprogramming. It involves the CPU performing numerous tasks through frequent switches that allow the user to interact with each program while it is running. With a time-sharing operating system, several users can share a computer simultaneously.

What is caching?

Cache memory is a high-speed access memory that stores copies of frequently used data from main memory to minimize the average time to access the data. It is a smaller and faster memory compared to the main memory. The CPU typically has several independent caches that store both data and instructions. When the CPU needs to access data or instructions, it first checks the cache memory. If the data is found in the cache, it is called a cache hit, and the data is retrieved much faster than if it had to be fetched from the main memory. On the other hand, if the data is not in the cache, it is called a cache miss, and the CPU must retrieve it from the main memory, which takes more time. The use of cache memory helps improve the overall performance of a computer system by reducing the average time to access data.

What is spooling?

Spooling is a technique used in computing to allow for simultaneous peripheral operations online (SPOOL), which refers to the process of putting jobs in a buffer or a special area in memory or on a disk, allowing a device to access them when it is ready. The main advantage of spooling is that it enables a device to access data at a different rate than the main memory or processor, which means that multiple devices can share a single resource without causing conflicts. For example, a printer can spool print jobs into a queue, allowing the user to continue working on their computer while the printer processes the jobs in the background. This results in better utilization of system resources and increased efficiency in the use of peripheral devices.

What is GUI?

Graphical User Interface (GUI) is a type of user interface that utilizes graphics, icons, and visual indicators to represent the available commands and actions. It provides an interactive and intuitive way for users to interact with software applications or computer systems. GUI is designed to make computing more accessible and easier to use for non-technical users. It allows users to perform various tasks by clicking on buttons, selecting options from menus, or dragging and dropping objects using a pointing device like a mouse or touchpad. GUI has become an essential component of modern computing and is widely used in operating systems, software applications, and websites.

Difference between process and program?

A program and a process are two important terms used in computer science. A program is a set of instructions that is designed to complete a specific task. On the other hand, a process is an instance of an executing program that is loaded into the main memory during execution. A program is a passive entity that resides in the secondary memory until it is deleted, while a process is an active entity that exists for a limited span of time and gets terminated after the completion of a task. A program is a static entity that does not have any resource requirement and only requires memory space for storing the instructions. In contrast, a process has a high resource requirement and needs resources like CPU, memory address, and I/O during its lifetime. Finally, the program does not have any control block, whereas the process has its own control block called Process Control Block (PCB) that is used to track its execution status.

What is a dispatcher?

The dispatcher is a crucial component of the operating system responsible for handing over the control of the CPU to the selected process by the short-term scheduler. The process of handing over control of the CPU to a new process requires the dispatcher to perform several critical functions, such as switching context, switching to user mode, and jumping to the appropriate location in the user program to resume execution. Context switching refers to the process of saving the current state of a process and restoring the state of the new process that will take over the CPU. The dispatcher also switches to user mode, which allows the process to execute in its context, and jumps to the correct location in the user program to restart its execution. Effective

dispatchers are essential to achieving a well-performing operating system that provides a seamless user experience.

Define the term dispatch latency?

Dispatch latency refers to the time interval between a request for process execution and the start of that process on the CPU. It is an important metric for real-time operating systems that must ensure that critical tasks are executed within specified time constraints. A bounded dispatch latency ensures that an application can predict and rely on the time it takes for a request to be processed, which is crucial for systems that require timely and predictable responses. To achieve a bounded dispatch latency, the system must have a scheduler that is optimized for the application's priorities, and the dispatcher must be able to quickly switch between processes to minimize the delay. By minimizing dispatch latency, the system can ensure that all processes, including real-time applications, run smoothly and without delay.

Associative Memory and Cache memory?

Associative memory and cache memory are two types of memory systems that are used in computing. Associative memory is a type of memory unit that is accessed by content. It reduces the time required to find the item stored in memory, as data is accessed by its content rather than by its address. It is used in situations where the search time needs to be very short. Its basic characteristic is its logic circuit for matching its content. On the other hand, cache memory is a fast and small memory that reduces the average memory access time. It is used when a particular group of data is accessed repeatedly. Its basic

characteristic is its fast access. In cache memory, data is accessed by its address, and it stores frequently accessed data in the small fast memory.

What is “Locality of reference”?

The locality of reference is a critical concept in computer science and is important in the design of computer systems, especially in memory management. It is based on the observation that computer programs tend to access a small set of memory locations for a particular period of time. This phenomenon is known as temporal locality. Spatial locality is another aspect of the locality of reference. It refers to the tendency of computer programs to access data or instructions that are located near each other in memory. By exploiting the locality of reference, computer systems can improve performance by minimizing the amount of time spent in accessing memory. This can be achieved through techniques such as caching, prefetching, and virtual memory.

How does an operating system abstract the underlying hardware from the applications that run on it?

An operating system abstracts the underlying hardware from the applications that run on it through a layer of abstraction called the system call interface. This interface provides a standard set of functions that applications can use to interact with the hardware, without needing to know the specifics of the hardware itself.

How do operating systems provide security mechanisms such as access control and authentication?

Operating systems provide security mechanisms such as access control and authentication through various means, including user accounts and permissions, encryption, and firewalls. Access control ensures that only authorized users can access system resources, while authentication ensures that users are who they claim to be.

How do operating systems handle interrupts and exceptions?

Operating systems handle interrupts and exceptions by providing interrupt handlers and exception handlers. Interrupts are used to signal the CPU that an external event has occurred, while exceptions are used to signal that an error or abnormal condition has occurred during program execution. The handlers provide the necessary actions to be taken when these events occur.

What is virtual memory, and how does it work in an operating system?

Virtual memory is a technique used by operating systems to manage memory resources. It enables applications to access a larger, virtual address space than is physically available, by using a combination of RAM and disk space. The operating system allocates a portion of the hard disk as a swap file or page file, and when RAM becomes full, pages

of memory are moved to the disk to free up space. When an application needs to access a page that has been moved to disk, the operating system retrieves it from the disk and moves it back into RAM.

What are the different types of operating system architectures, and how do they differ?

There are different types of operating system architectures, including monolithic kernels, microkernels, and hybrid kernels. Monolithic kernels contain most operating system services in a single address space and run in kernel mode, while microkernels move most of the services to user space and run in user mode. Hybrid kernels, as the name suggests, combine elements of both monolithic and microkernel architectures. The choice of architecture depends on factors such as performance, reliability, and security.

How do real-time operating systems differ from general-purpose operating systems?

Real-time operating systems (RTOS) are designed to execute tasks and respond to events in a deterministic and timely manner, whereas general-purpose operating systems are optimized for flexibility and multitasking. RTOS typically prioritize tasks based on their timing requirements, use a small kernel with minimal overhead, and provide mechanisms for real-time communication and synchronization.

Some of the challenges in designing an operating system for a massively parallel computer system?

Designing an operating system for a massively parallel computer system requires addressing issues such as load balancing, fault tolerance, and communication and synchronization across multiple processors. Achieving high performance and scalability while maintaining reliability and consistency is a major challenge.

How does an operating system handle distributed file systems and distributed computing?

Operating systems for distributed file systems and distributed computing must provide mechanisms for transparent access to remote resources, handle data consistency and synchronization across multiple nodes, and provide fault tolerance and security.

What is the role of microkernels in operating system design, and what are some advantages and disadvantages of this approach?

Microkernels are a modular approach to operating system design that separate the minimal kernel functionality from other system services, which can run as separate processes or in user space. This approach can provide improved reliability, security, and flexibility, but can also introduce performance overhead and complexity.

What are some emerging trends and technologies in operating system design and implementation, and how are they likely to impact the future of computing?

Some emerging trends and technologies in operating system design include containerization, which provides lightweight virtualization and resource isolation; serverless computing, which abstracts away server management and scaling; and edge computing, which brings computing resources closer to the devices and sensors generating data. These trends are likely to lead to more decentralized, flexible, and efficient computing systems in the future.