

Distributed Systems

Operating Systems Support

Operating System Support

- Operating systems play a crucial role in the functionality and efficiency of distributed systems, where multiple independent entities collaborate to achieve common goals.
- One of their fundamental contributions lies in resource management. such as processing power, memory, and storage, networks across interconnected nodes.
- This orchestration ensures that each component receives adequate resources, optimizing performance and preventing resource contention, ultimately enhancing the system's overall productivity.

Operating System Support

In order for an Operating system to function of a distributed system machine, It has to have the following components:

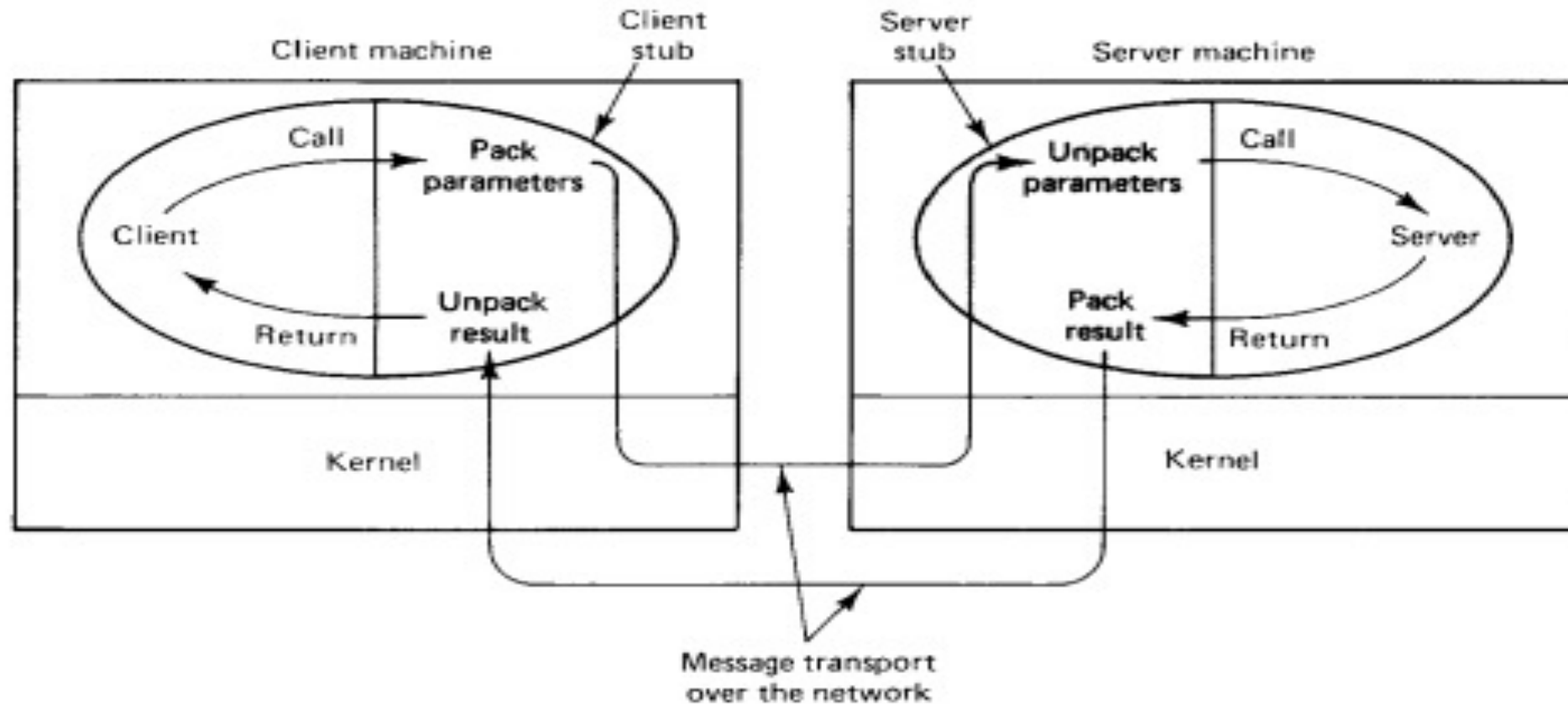
- Kernel
- Distributed Processes Execution
- Interprocess Communication
- Network Interfaces
- I/O and Memory Resource Management
- Network File System
- Device Driver
- Coordination and Agreement Interfaces for Shared Resources
- Security and Access Control for different distributed Resources

Operating System Support

Kernel

- The core of the operating system responsible for managing hardware resources, process scheduling, and memory allocation.
- In a distributed environment, the kernel oversees these tasks not only on a single machine but also coordinates across multiple nodes, ensuring efficient resource utilization and communication.
- It has system calls that are executed in the kernel mode

Operating System Support



Operating System Support

Kernel

User Mode:

- Applications and user processes run with restricted access to system resources, interacting with the operating system through controlled interfaces.
- It is called unprivileged mode.

Operating System Support

Kernel

Kernel Mode:

- known as supervisor or privileged mode
- Provides unrestricted access to the system's hardware and resources, allowing the operating system's core components,
- System calls that execute critical tasks such as Processes, managing memory, I/O device drivers and all operating system services.

Operating System Support

Kernel

- The transition between user mode and kernel mode is controlled by the operating system, ensuring that sensitive operations are carried out securely while protecting the system's stability and integrity.
- It provides protection for the underlying resources from the direct intervention of the user space applications.
- Helps for Application developer so they will not need to spend time programming the Hardware.

Operating System Support

Distributed Processes Execution:

- The execution of processes spans across multiple machines, demanding a sophisticated approach from the operating system.
- The distributed operating system must manage the allocation of computing resources to execute processes efficiently.
- This involves load balancing mechanisms to ensure that no single node is overwhelmed, optimizing overall performance.
- Fault tolerance becomes paramount, necessitating the ability to migrate processes between machines in case of failures.

Operating System Support

Distributed Processes Execution:

- The operating system must offer a environment that abstracts the complexities of distributed computing, allowing processes to run across the networked infrastructure.
- This requires a careful balance between resource allocation, performance optimization, and fault tolerance strategies, all orchestrated by the operating system to deliver a reliable and efficient distributed process execution environment.
- Linux-based clusters can efficiently execute distributed processes, ensuring optimal resource utilization.

Operating System Support

Interprocess Communication(IPC):

- Takes on a heightened significance in distributed systems, where processes operate independently across different machines.
- The operating system must provide robust IPC mechanisms to facilitate efficient data exchange and synchronization between these distributed processes.
- This involves implementing communication protocols that can traverse the network, managing data serialization and deserialization to ensure compatibility across diverse systems.

Operating System Support

Interprocess Communication(IPC):

- Synchronization primitives, such as semaphores or message queues, become essential for coordinating the activities of distributed processes.
- The operating system should shield developers from the complexity of network communication, providing higher-level abstractions that make IPC easy.

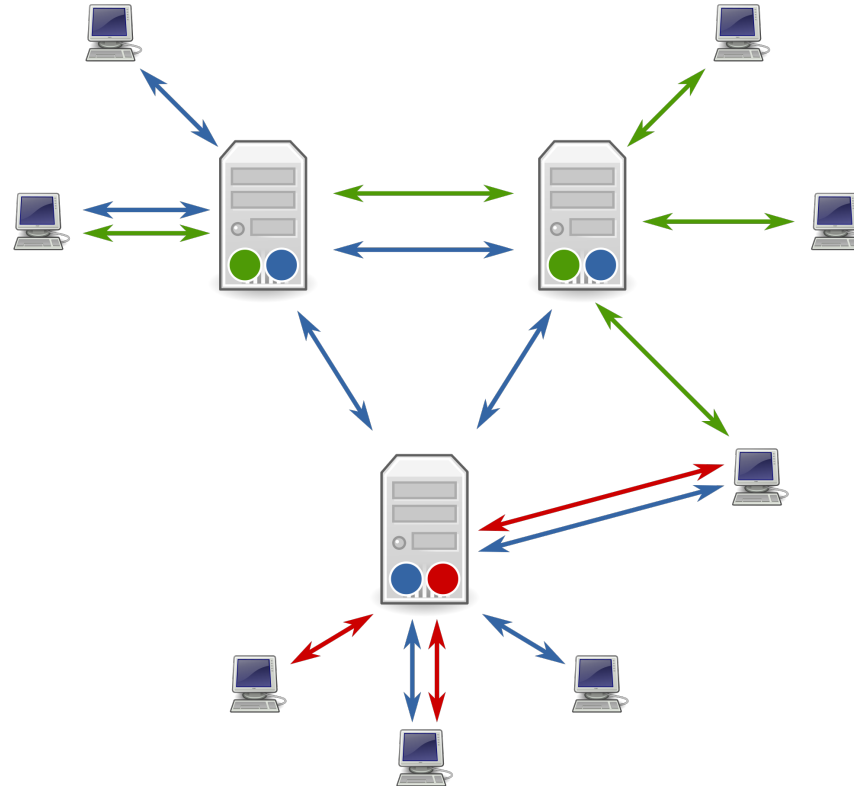
Operating System Support

Interprocess Communication(IPC):

- This requires an approach to IPC design within the operating system, considering latency, reliability, and the diverse nature of networked communication to enable smooth collaboration between processes in a distributed environment.
- Linux MPI is an example of Distributed Interprocess Communication.

Operating System Support

Interprocess Communication(IPC):



Operating System Support

Network Interfaces:

- Network interfaces are the main issue through which distributed systems communicate.
- The operating system must act as a mediator between applications and the network.
- It involves managing network protocols, handling data transmission, and ensuring reliable communication between nodes in the distributed system.
- The operating system should provide abstractions for network communication, shielding applications from the complexities of low-level networking details.

Operating System Support

Network Interfaces:

- The operating system should provide abstractions for network communication, shielding applications from the complexities of low-level networking details.
- It must handle tasks such as routing, addressing, and error recovery to ensure data exchange across the distributed environment.
- The operating system needs to adapt to varying network conditions, optimizing communication for factors such as latency and bandwidth.

Operating System Support

Network Interfaces:

- By providing a robust set of network interfaces, the operating system enables applications to interact effectively in a distributed setting, fostering efficient communication and collaboration across the networked infrastructure.
- Linux and Windows Server operating systems provides variety of libraries and APIs for network programming in which the developer has no need to worry about the details of actual operation with the underlying network.
- APIs provide programming for TCP/IP protocols including UDP and TCP

Operating System Support

I/O and Memory Resource Management:

- Effective resource management is fundamental in a distributed system
- the operating system must oversee the allocation and deallocation of memory while managing input/output operations efficiently.
- In memory management, the operating system must employ distributed memory allocation strategies, considering factors such as locality and access patterns to optimize performance.

Operating System Support

I/O and Memory Resource Management:

- In the context of I/O operations, the operating system should handle the coordination of diverse storage and retrieval tasks across the network.
- This involves caching mechanisms, buffering strategies, and intelligent scheduling to prevent bottlenecks and ensure a smooth flow of data.
- Balancing the demands of memory and I/O in a distributed context requires a sophisticated approach from the operating system, necessitating a deep understanding of the distributed architecture and the dynamic nature of resource utilization.

Operating System Support

Network File System (NFS)

- Offer a distributed file system protocol that connects and facilitates file sharing across multiple machines within a network.
- Originating from Sun Microsystems,
- NFS operates on the client-server model, where the NFS server exports directories and files,
- NFS clients can mount these resources, making them accessible as if they were local.

Operating System Support

Network File System (NFS)

- This architecture provides transparent collaboration and data sharing in distributed environments, allowing different machines, regardless of their operating systems, to access and interact with shared files and directories.
- Operating over the Open Network Computing Remote Procedure Call (ONC RPC) protocol
- Its use of both UDP and TCP for data transmission accommodates diverse network configurations

Operating System Support

Network File System (NFS)

- The distributed nature of NFS becomes clear in its platform independence, and is capable to function in heterogeneous computing environments where collaboration across different operating systems is substantial.
- NFS facilitates the integration of various machines into a cohesive distributed system, enabling efficient and standardized file sharing practices.

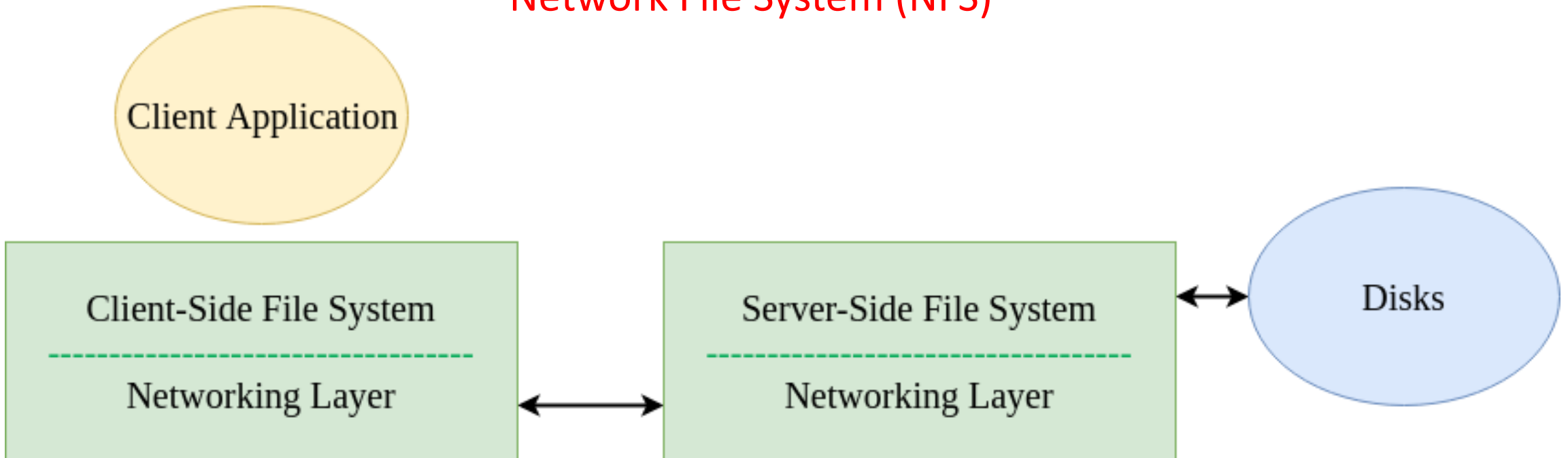
Operating System Support

Network File System (NFS)

- NFS server maintains an export list, defining accessible directories and files, mirroring the decentralized nature of data storage and accessibility in distributed environments.
- Clients, by mounting these exported resources, extend the reach of their local file systems across the network.
- In this way, NFS serves as a distributed file-sharing solution, bridging the gap between multiple machines and contributing to the cohesive functioning of distributed systems.
- All cloud storage systems are examples on NFS

Operating System Support

Network File System (NFS)



<https://www.geeksforgeeks.org/network-file-system-nfs/>

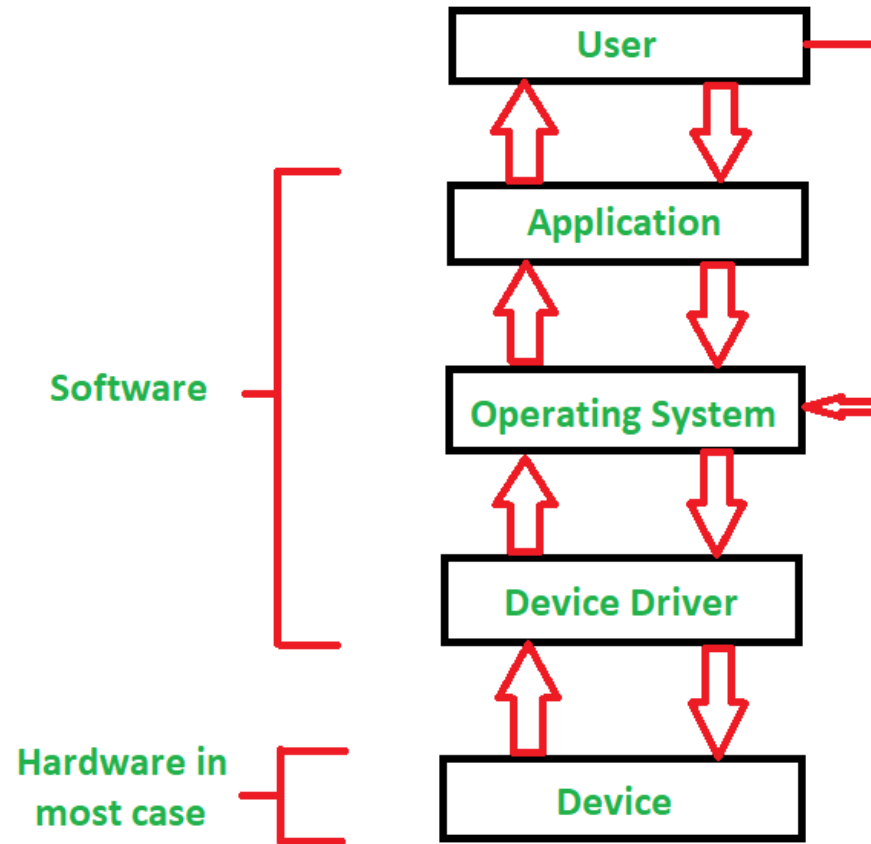
Operating System Support

Device Driver

- Device drivers interface with hardware components.
- In a distributed system, device drivers must be aware of the distributed nature of resources, managing communication with devices across the network.
- This includes handling distributed input/output operations and ensuring synchronization across devices.

Operating System Support

Device Driver



Operating System Support

Coordination and Agreement Interfaces for Shared Resources:

- Shared resources, such as databases or files, demand careful coordination in a distributed system to maintain consistency and avoid conflicts.
- The operating system must provide interfaces for processes to coordinate access to these shared resources, employing distributed coordination mechanisms.
- This involves distributed locking protocols, transaction management, and consensus algorithms to ensure that multiple processes can safely access and modify shared data.

Operating System Support

Coordination and Agreement Interfaces for Shared Resources:

- The operating system plays a important role in managing the complexities of distributed coordination, offering developers the tools and abstractions needed to handle shared resources effectively.
- This requires a nuanced understanding of distributed algorithms and a robust implementation of coordination interfaces within the operating system to guarantee the integrity and consistency of shared resources across the networked environment.

Operating System Support

Coordination and Agreement Interfaces for Shared Resources:

- Consider a collaborative document editing platform where multiple users concurrently edit the same document stored on distributed servers.
- The operating system provides coordination interfaces to manage access to the shared document. It employs distributed locking mechanisms and consensus algorithms to ensure that users can edit the document without conflicting changes.

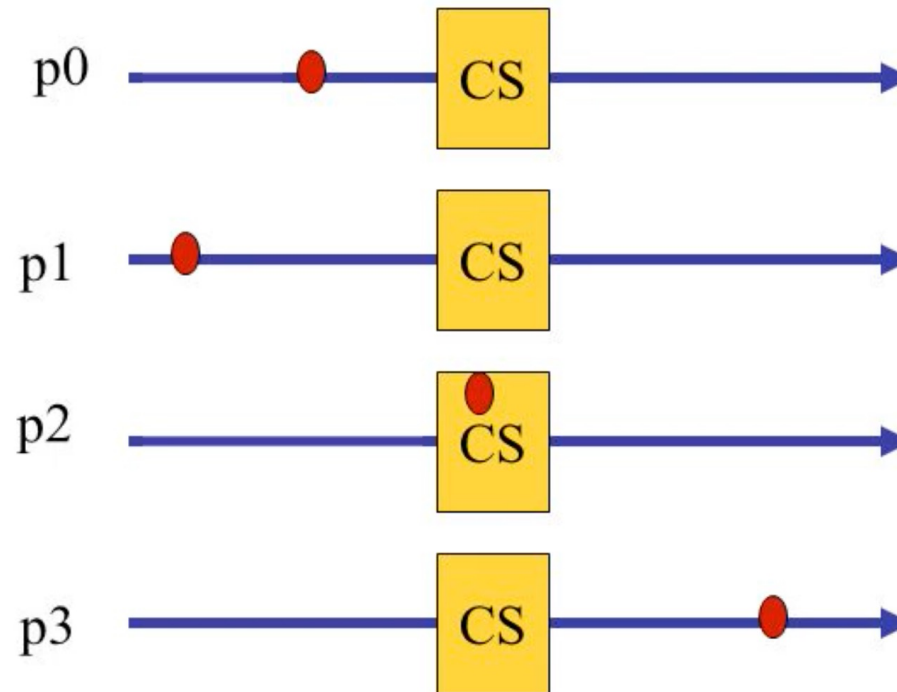
Operating System Support

Coordination and Agreement Interfaces for Shared Resources:

- This coordination prevents data inconsistencies and maintains document integrity, showcasing the operating system's role in managing shared resources in a distributed collaboration environment.
- Distributed Mutual Exclusion (DMX).

Operating System Support

Distributed Mutual Exclusion



Operating System Support

Security and Access Control for Different Distributed Resources:

- Security is an crucial concern in distributed systems, and the operating system must implement robust measures to protect against unauthorized access, data breaches, and other security threats.
- Authentication mechanisms are crucial for verifying the identity of processes and users in a distributed environment.
- Encryption becomes essential to secure data transmission over the network, safeguarding the confidentiality and integrity of information.

Operating System Support

Security and Access Control for Different Distributed Resources:

- Access control mechanisms within the operating system dictate permissions and privileges, ensuring that only authorized entities can access and modify specific resources.
- The operating system must adapt to the dynamic nature of distributed systems, where nodes may join or leave the network. This requires a flexible and scalable approach to security, involving strategies such as key management and secure communication protocols.
- By providing comprehensive security and access control interfaces, the operating system acts as a guardian of the distributed infrastructure, preserving the confidentiality, integrity, and availability of resources across the networked environment.

Operating System Support

Security and Access Control for Different Distributed Resources:

- In a financial services application operating in a distributed cloud environment, the operating system enforces security and access control.
- It implements robust authentication mechanisms to verify user identities and employs encryption for secure communication between distributed components.
- Access control mechanisms within the operating system gives permissions for accessing financial data, ensuring that only authorized users or services can retrieve sensitive information.

Operating System Support

Security and Access Control for Different Distributed Resources:

- The operating system plays a critical role in facilitating encryption, ensuring the security and confidentiality of data. The operating system is involved in encryption in the following:
 - File-Level Encryption
 - Full Disk Encryption
 - Network Encryption
 - Key Management
 - Secure Sockets Layer (SSL) and Transport Layer Security (TLS)

Operating System Support

Commonly Used Operating Systems For distributed Systems

Linux:

- Distributions like Ubuntu Server, CentOS, and others, is a widely used operating system in the of distributed systems and cloud computing.
- Open-source nature, robust networking capabilities, and support for a variety of distributed computing tools make it a popular choice for deploying distributed applications and managing cloud infrastructure.

Operating System Support

Commonly Used Operating Systems For distributed Systems

Windows Server:

- Microsoft's Windows Server editions, such as Windows Server 2019, are tailored for distributed computing environments.
- These operating systems provide features like Active Directory for user management, Hyper-V for virtualization, and various networking services, making them suitable for building and managing distributed infrastructures.

Operating System Support

Network Operating System vs Distributed Operating System

Networked Operating Systems:

- Networked operating system primarily focuses on enabling communication and resource sharing across a network of interconnected computers.
- It provides features such as file sharing, printer sharing, and basic communication services over a network.
- Linux Server and Windows Server are considered as networked operating systems.

Operating System Support

Network Operating System vs Distributed Operating System

Distributed Operating Systems:

- Distributed operating system goes beyond basic networking functions and emphasizes the distribution of tasks and resources across multiple interconnected machines.
- It involves the coordination and management of distributed resources, supporting parallel processing, fault tolerance, and load balancing.
- Linux, Windows Server, and others can be configured to operate in a distributed manner, with features that support load balancing, distributed file systems, and parallel processing.

Operating System Support

Support for cloud systems

- Cloud computing relies heavily on operating systems to efficiently manage and orchestrate virtualized resources across vast and dynamic infrastructures.
- Operating systems in cloud systems play an important role in facilitating the deployment, scaling, and management of virtual machines (VMs) or containers.
- They provide the necessary abstraction layers, allowing cloud users to interact with their applications while the underlying operating system efficiently allocates and manages computing resources.

Operating System Support

Support for cloud systems

- Popular cloud providers, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP), rely on sophisticated operating system technologies to optimize resource utilization and provide a good experience for users.
- One critical aspect of operating systems in cloud environments is their role in enabling multi-tenancy and resource isolation.
- With multiple users and applications sharing the same physical hardware, the operating system must ensure that each tenant's data and processes remain isolated and secure.

Operating System Support

Support for cloud systems

- Techniques such as virtualization and containerization, supported by operating system features, allow cloud providers to achieve this isolation, creating a secure and efficient environment for running diverse workloads concurrently.
- Operating systems also play a key role in managing the elasticity of cloud services, dynamically allocating and deallocating resources based on demand to optimize performance and cost-effectiveness.

Operating System Support

Support for cloud systems

- Security is a major concern in cloud computing, and operating systems contribute significantly to the overall security of cloud environments.
- They implement security measures such as access controls, encryption, and secure boot processes to safeguard against unauthorized access and data breaches.
- The operating systems in cloud systems often integrate with identity and access management services provided by the cloud platform, enhancing security through user authentication, authorization, and auditing.

References

- <https://www.geeksforgeeks.org/difference-between-network-os-and-distributed-os/>
- <https://www.javatpoint.com/distributed-operating-system>
- <https://www.geeksforgeeks.org/interprocess-communication-in-distributed-systems/>
- <https://www.geeksforgeeks.org/inter-process-communication-ipc/>
- <https://www.geeksforgeeks.org/device-driver-and-its-purpose/>

References

- <https://www.geeksforgeeks.org/mutual-exclusion-in-distributed-system/>
- <https://www.tutorialspoint.com/mutual-exclusion-in-a-distributed-system>
- <https://www.geeksforgeeks.org/network-file-system-nfs/>
- <https://slideplayer.com/slide/6594818/>
- https://www.tutorialspoint.com/operating_system/os_security.htm