



Remote Procedure Calls (RPC)



#1: Introduction to Threads and Remote Procedure Calls (RPC)

- Threads: A thread is the smallest unit of a process that can be scheduled and executed by the operating system.
- Threads within the same process share the same memory space but have separate execution contexts, such as the program counter and stack.
- Threads can be implemented at the user level or kernel level:
- User-Level Threads: Managed by a user-level library, invisible to the kernel.
- Kernel-Level Threads: Managed directly by the operating system.



Remote Procedure Calls (RPC):

- RPC is a protocol that one program can use to request a service from a program located on another computer in a network.
- Abstracts the procedure call mechanism across a network, making a remote service invocation appear as a local procedure call.
- Key components:
 - Client: The entity requesting the service.
 - Server: The entity providing the service.
 - Stub: Generated code that acts as an intermediary, handling communication between the client and server.



System Models

- A. The Workstation Model :-
- Each user has a dedicated workstation, and all the processes initiated by that user are executed on their workstation.
- Advantages:
 - Users have control over their work environment.
 - Can handle moderate workloads effectively.
- Disadvantages:
 - Underutilization of resources if the workstation is idle.
 - Scalability issues when workload exceeds the capacity of a single workstation.



B.Using Idle Workstations

- Utilizes the idle processing power of workstations when they are not in use by their primary user.
- Mechanism:
- Idle workstations are identified by the system.
- Distributed tasks (such as background processing) are offloaded to these idle machines.
- Advantages:
- Better utilization of resources across the network.
- Reduces the need for additional hardware for background processing tasks.



Continue

- Disadvantages:
- Complexity in managing and scheduling tasks across multiple workstations.
- Possible disruption of the primary user's tasks if the workstation becomes active.



C. The Processor Pool Model

- Instead of dedicating workstations to individual users, a pool of processors is maintained that can be dynamically allocated to tasks.
- Mechanism:
- Users submit tasks to the system.
- The system allocates processors from the pool based on the task requirements.
- Advantages:
- High flexibility in resource allocation.
- Efficient utilization of processing power, especially in environments with varying workloads.



Continue

- Disadvantages:
- Increased overhead in managing the processor pool.
- Potential for contention if multiple tasks require a large number of processors simultaneously.



Comparison of System Models

Model	Advantages	Disadvantages
Workstation Model	- User control, moderate workload handling	- Resource underutilization, scalability issues
Using Idle Workstations	Resource optimization, reduced hardware costs	Complex management, potential user disruption
Processor Pool Model	High flexibility, efficient resource use	Management overhead, possible processor contention



Homework Questions

- 1) Compare and contrast the efficiency of the Workstation Model and the Processor Pool Model in handling a large-scale distributed system. Which model would perform better in a system with unpredictable workloads, and why?
- 2) How does the "Using Idle Workstations" model address the problem of resource underutilization in the Workstation Model? What potential challenges might arise when integrating this model in a corporate network with varying levels of workstation usage?
- 3) Discuss the scalability limitations of the Workstation Model. How does the Processor Pool Model overcome these limitations, and what trade-offs might it introduce?



Homework Questions

4. In a distributed system using the Processor Pool Model, how would thread management differ from a system using the Workstation Model? Consider factors such as synchronization, scheduling, and load balancing in your analysis.
5. Analyze the role of RPC in the Processor Pool Model. How does RPC facilitate communication and resource allocation in this model compared to the Workstation Model?
6. In the "Using Idle Workstations" model, how does the variability in user behavior impact system performance? What strategies could be implemented to mitigate any negative effects?



Thank You!