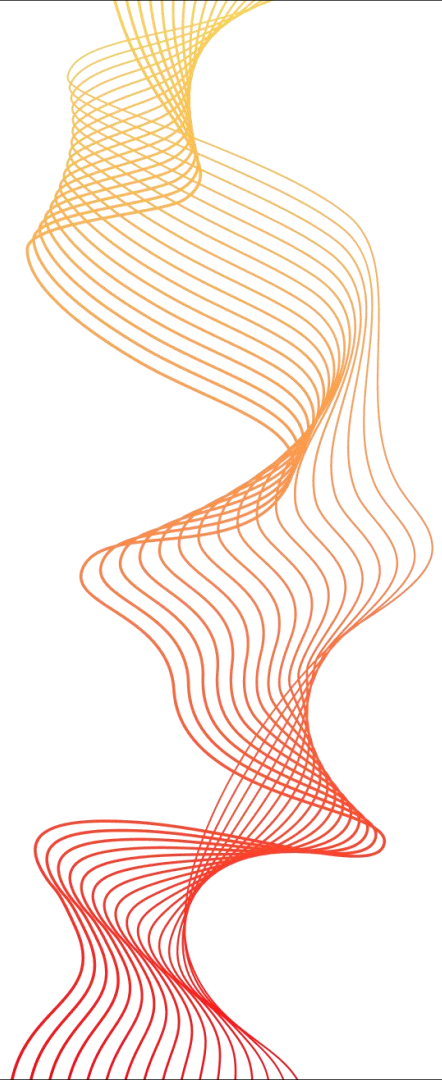
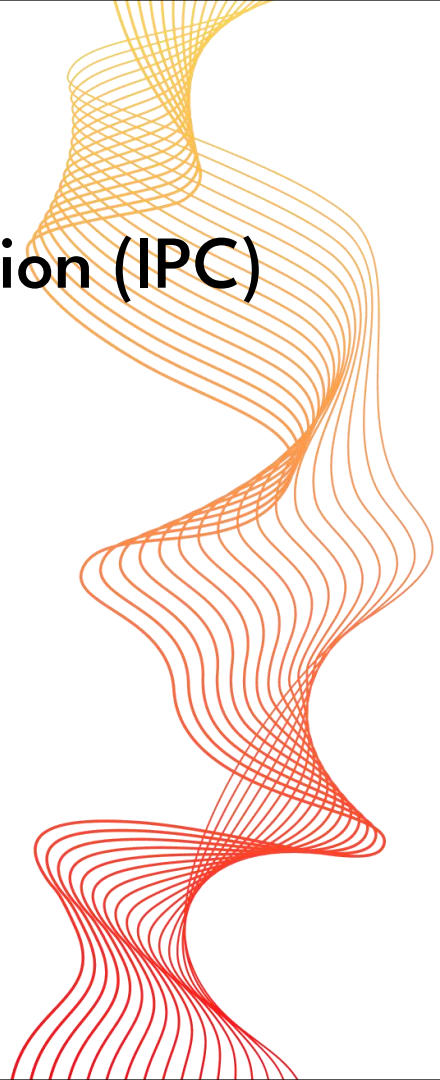
Interprocess Communication

Chapter 6



Introduction to Process Intercommunication (IPC)

This presentation covers the basics of IPC, from the importance of communication channels to the challenges and types of IPC highways. It delves into shared memory, semaphores, mutexes, message queues, and pipes, and their real-world applications in inter-process communication tools, client-server architectures, and distributed systems. By the end of this chapter, you will have a better understanding of the different types of IPC and their significance in applications, operating systems, and networks.

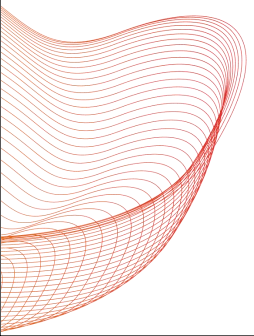


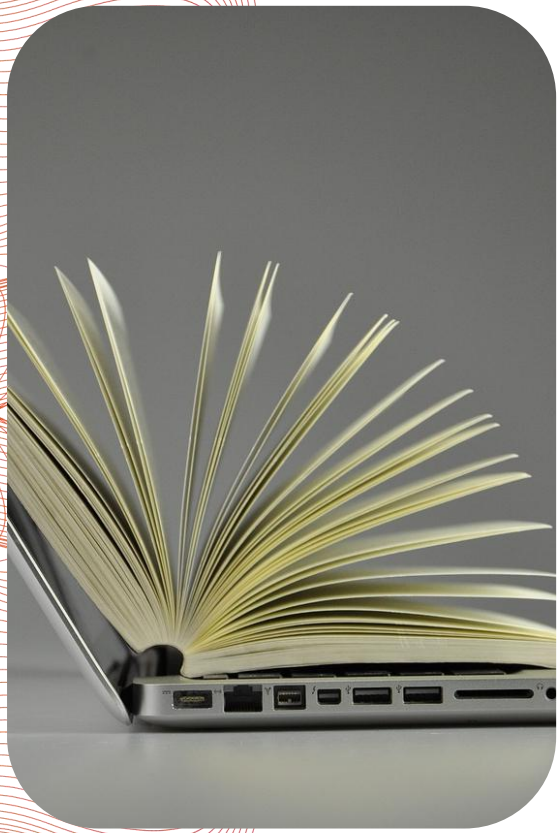
Why Talk About IPC?

Imagine a city without traffic lights: Processes, like citizens, need communication channels to avoid chaos.

Sharing is caring: Processes need to exchange data, synchronize actions, and share resources efficiently.

From simple tasks to complex systems: Effective IPC empowers applications, operating systems, and distributed networks.



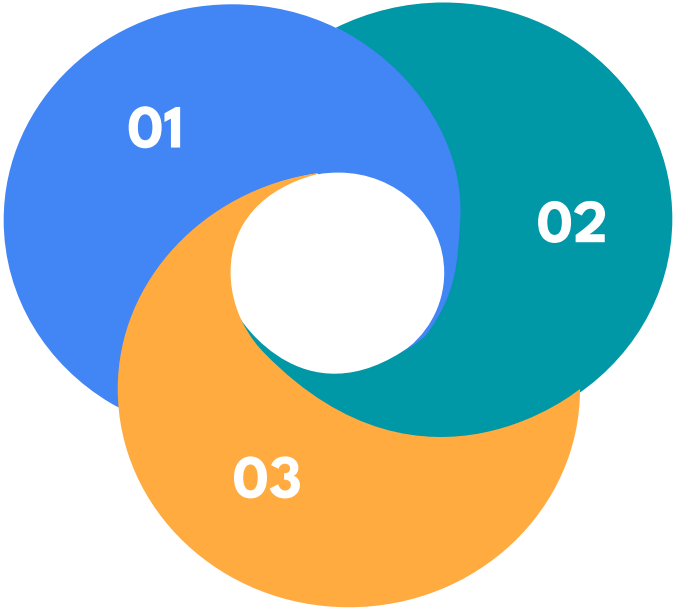


Challenges (IPC)

- **Synchronization**: Avoiding collisions! Processes need to take turns when accessing shared resources.
- **Security**: Who has the keys? Protecting data and resources from unauthorized access.
- **Reliability**: Delivering the mail! Ensuring messages reach their destination without errors or loss.
- **Performance**: Speeding up traffic! Balancing communication overhead with application responsiveness.

Types of IPC Highways

Direct Communication: Sharing the main road (shared memory), taking turns at crosswalks (semaphores).



Indirect Communication: Sending letters (message queues), talking through tubes (pipes).

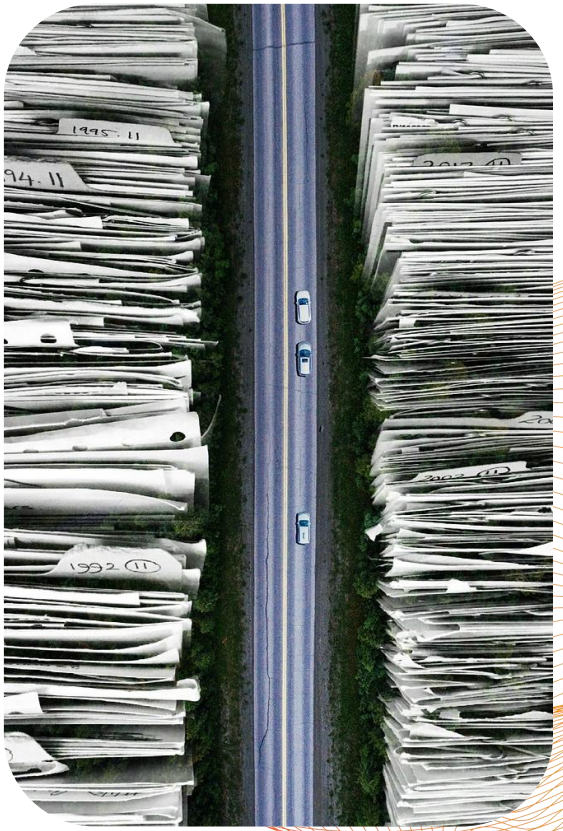
Remote Procedure Calls: Borrowing a neighbor's phone (RPCs) to make distant calls.

Shared Memory - Sharing the City Square

01 Real-world examples: Collaborative editing, shared databases, efficient inter-process data transfer.

02 Everyone sees the same billboard: Processes access a common memory segment to exchange data directly.

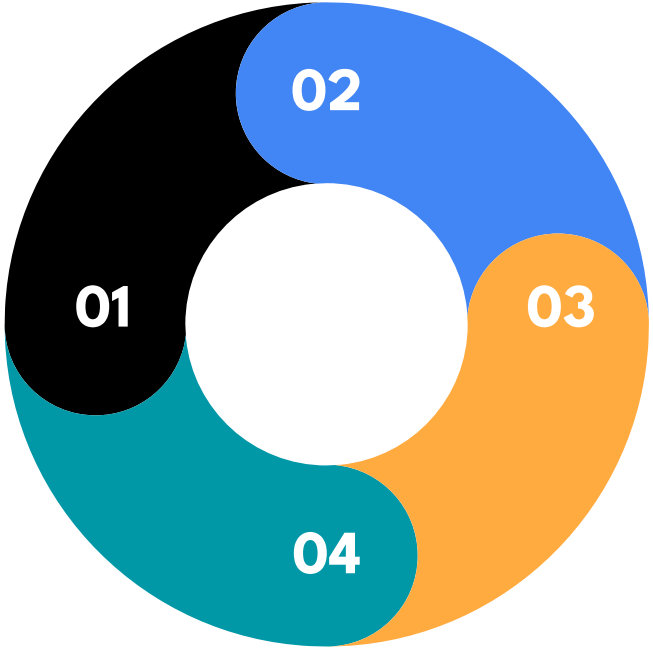
03 Careful crossing! Semaphores and critical sections keep traffic flowing smoothly and avoid accidents.



Semaphores and Mutexes - Traffic Lights for Resources

Controlling intersections: Semaphores regulate access to shared resources like printers or files.

Binary semaphores: One car at a time! Ensuring mutual exclusion for critical sections.



Counting semaphores: Managing resource availability, letting multiple cars park in a shared garage.

Mutexes: Like a one-way street, granting exclusive access to critical resources.

Message Queues and Pipes - Mailboxes and Speaking Tubes

No direct access, but messages still get through: Processes send and receive messages without sharing memory.

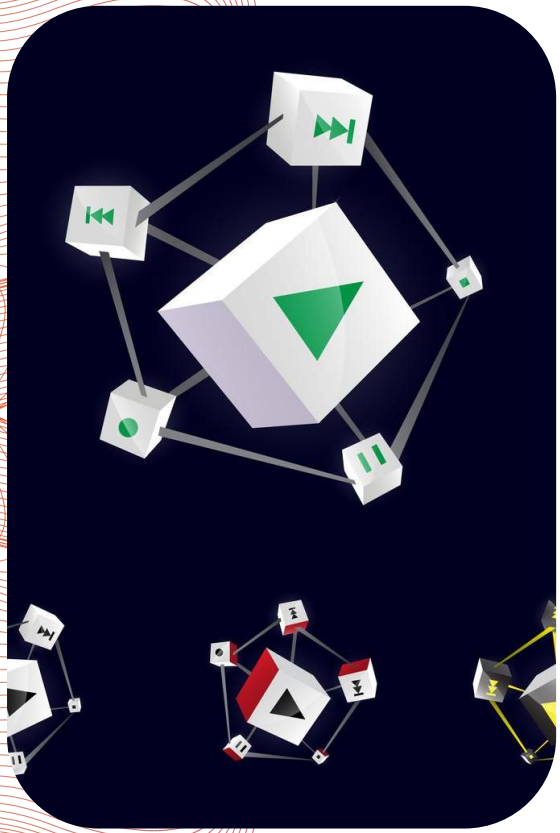
- **Message queues: Like a post office box, messages are stored and delivered in order.**
- **Pipes: Oneway or two-way communication channels, like talking through a tube.**

Remote Procedure Calls - Borrowing the Neighbor's Phone

- **Making calls without leaving your house:** Invoke procedures on remote processes seamlessly.
- **Client-server model:** Like using a public phone, the client sends requests and receives responses.
- **Transparent communication:** Hide the underlying network details, making programming easier.
- **Benefits:** Simplifies distributed programming, hides communication complexity.

Choosing the Right Communication Channel

- **Shared memory:** Fast and efficient for small data, but requires careful synchronization.
- **Semaphores and mutexes:** Ideal for resource control, but low-level and error-prone.
- **Message queues and pipes:** Flexible for asynchronous communication, but may introduce latency.
- **RPCs:** High-level abstraction, good for distributed systems, but adds complexity.



Take Away

- Understanding IPC is crucial for building efficient and reliable applications and operating systems.
- Different mechanisms cater to specific needs, choose the right tool for the job.
- Security, performance, and scalability are key considerations in modern IPC designs.

Thank you. Please feel free to ask any questions.

?