Sistemas Operativos

Cursada 2022

Comisión S21 y S22

IPC (Comunicación entre Procesos)

Habíamos hablado anteriormente de 2 tipos de procesos o 2 formas de caracterizarlos

- **Independientes**

  No requieren intervención de otros procesos

- **Cooperativos**

  La problemática en lugar de resolverla un solo proceso, lo hacen varios procesos cooperando entre ellos

  Ventajas (solapamiento de tiempos)

IPC (Comunicación entre Procesos)

El avance de los procesos cooperativos hizo que el modulo fuera creciendo cada vez mas

- ***Administración de la CPU***
- ***Administración de la memoria***
- ***Administración del Sistema de archivos***
- ***IPC (soporta todos los protocolos de red)***
- ***Seguridad***

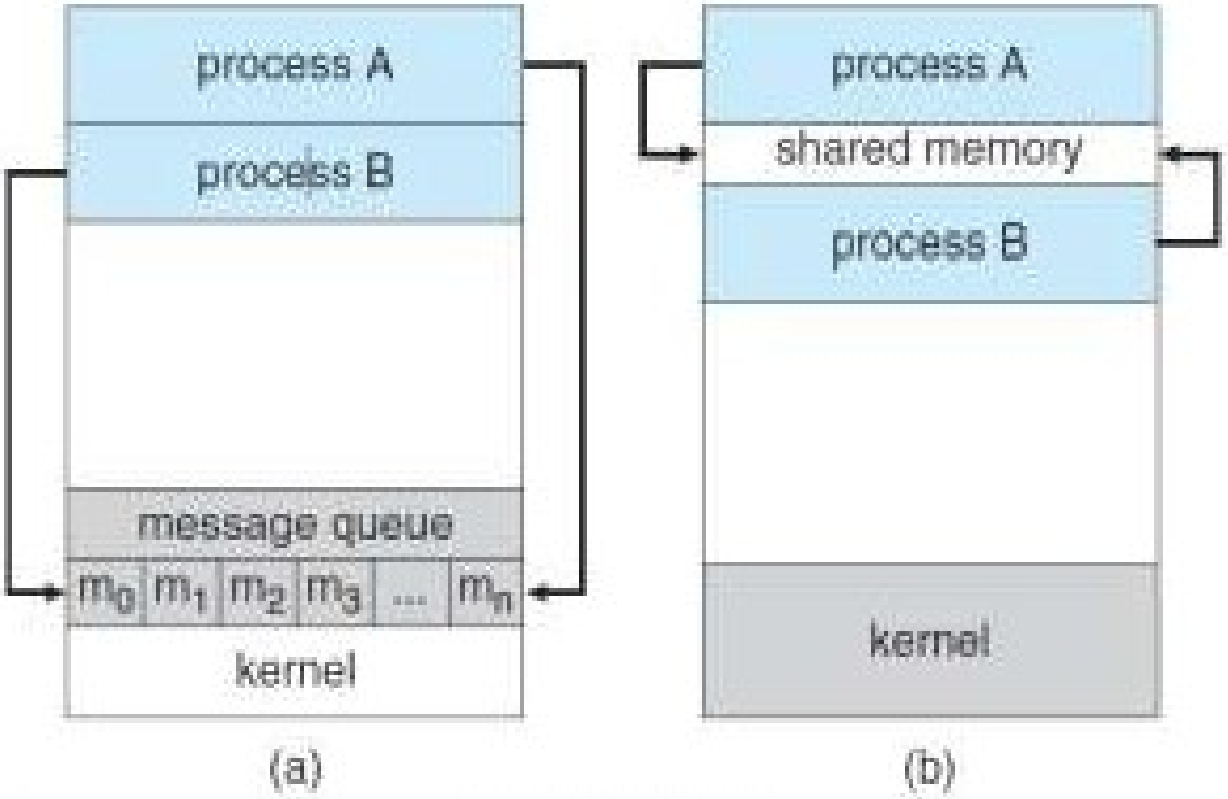
IPC (Comunicación entre Procesos)

Interprocess Communication

- Processes within a system may be *independent* or *cooperating*
- Cooperating process can affect or be affected by other processes, including sharing data
- Reasons for cooperating processes:
  - Information sharing
  - Computation speedup
  - Modularity
  - Convenience
- Cooperating processes need interprocess communication (IPC)
- Two models of IPC
  - Shared memory
  - Message passing

Como se comunican los procesos entre si:

- **Pasaje de mensaje**
- **Memoria compartida**



IPC (Comunicación entre Procesos)

Procesos Independientes

- **Espacio de memoria**
- **No interviene el S.O.**

Procesos Cooperativos

- **Intervención de S.O.**
- **Buzones**

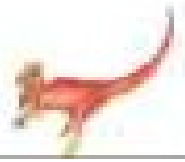
Otros temas relacionados con el IPC

- **Puertos**
- **Socket**
- **Buffering**

IPC (Comunicación entre Procesos)

Interprocess Communication – Message Passing

- Mechanism for processes to communicate and to synchronize their actions
- Message system – processes communicate with each other without resorting to shared variables
- IPC facility provides two operations:
  - **send**(*message*) – message size fixed or variable
  - **receive**(*message*)
- If *P* and *Q* wish to communicate, they need to:
  - establish a *communication link* between them
  - exchange messages via send/receive
- Implementation of communication link
  - physical (e.g., shared memory, hardware bus)
  - logical (e.g., logical properties)



IPC (Comunicación entre Procesos)

Implementation Questions

- How are links established?
- Can a link be associated with more than two processes?
- How many links can there be between every pair of communicating processes?
- What is the capacity of a link?
- Is the size of a message that the link can accommodate fixed or variable?
- Is a link unidirectional or bi-directional?

IPC (Comunicación entre Procesos)

Direct Communication

- Processes must name each other explicitly:
  - **send** (*P, message*) – send a message to process P
  - **receive**(*Q, message*) – receive a message from process Q
- Properties of communication link
  - Links are established automatically
  - A link is associated with exactly one pair of communicating processes
  - Between each pair there exists exactly one link
  - The link may be unidirectional, but is usually bi-directional

IPC (Comunicación entre Procesos)

Indirect Communication

- Messages are directed and received from mailboxes (also referred to as ports)
  - Each mailbox has a unique id
  - Processes can communicate only if they share a mailbox
- Properties of communication link
  - Link established only if processes share a common mailbox
  - A link may be associated with many processes
  - Each pair of processes may share several communication links
  - Link may be unidirectional or bi-directional

IPC (Comunicación entre Procesos)

Indirect Communication

- Operations
  - create a new mailbox
  - send and receive messages through mailbox
  - destroy a mailbox
- Primitives are defined as:

  send(*A, message*) – send a message to mailbox A

  receive(*A, message*) – receive a message from mailbox A

IPC (Comunicación entre Procesos)

Indirect Communication

- Mailbox sharing
  - P_1, P_2, and P_3 share mailbox A
  - P_1, sends; P_2 and P_3 receive
  - Who gets the message?
- Solutions
  - Allow a link to be associated with at most two processes
  - Allow only one process at a time to execute a receive operation
  - Allow the system to select arbitrarily the receiver. Sender is notified who the receiver was.

IPC (Comunicación entre Procesos)

Synchronization

- Message passing may be either blocking or non-blocking
- **Blocking** is considered **synchronous**
  - **Blocking send** has the sender block until the message is received
  - **Blocking receive** has the receiver block until a message is available
- **Non-blocking** is considered **asynchronous**
  - **Non-blocking** send has the sender send the message and continue
  - **Non-blocking** receive has the receiver receive a valid message or null

Radio

Tel, what

IPC (Comunicación entre Procesos)

Buffering

- Queue of messages attached to the link; implemented in one of three ways
  1. Zero capacity – 0 messages
     Sender must wait for receiver (rendezvous)
  2. Bounded capacity – finite length of *n* messages
     Sender must wait if link full
  3. Unbounded capacity – infinite length
     Sender never waits

IPC (Comunicación entre Procesos)

Como ejemplos **de Sistemas IPC tenemos los:**

- **Sistemas POSIX (mundo unix)**

  Ej pipe en linux

  ls –l | wc -l

- **LPC – Local Process Call (mundo windows)**

  Windows usa lo que se conoce también como RPC

  Lo trabaja con el mismo mecanismo que para el modelo cliente/servidor

  type aa.txt |findstr 22

Ejemplo de IPC Windows

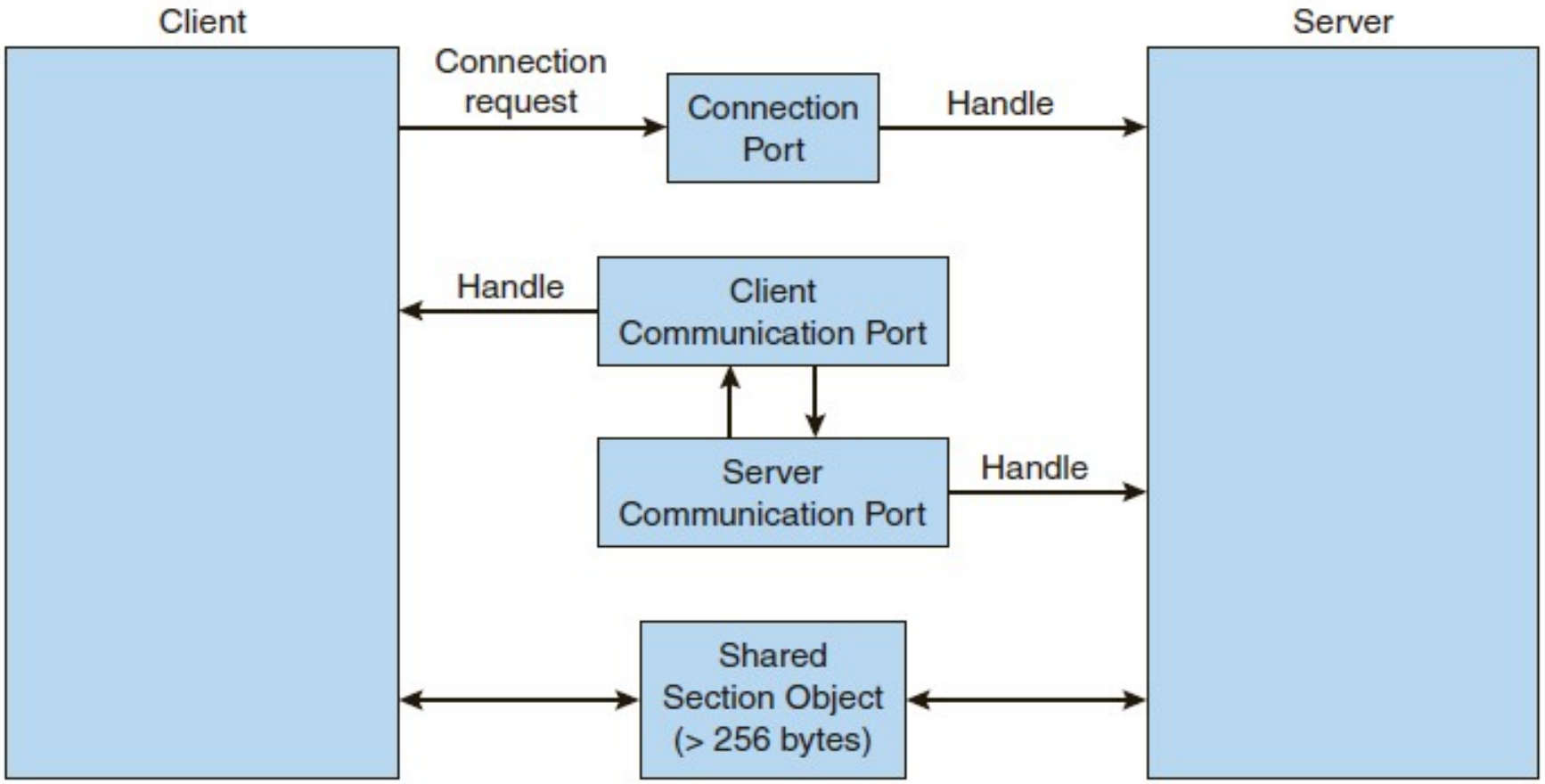


Figure 3.19 Advanced local procedure calls in Windows.

IPC (Comunicación entre Procesos)

Communications in Client-Server Systems

- Sockets
- Remote Procedure Calls
- Remote Method Invocation (Java)

Windows utiliza dentro de sus propios procesos el modelo cliente/servidor

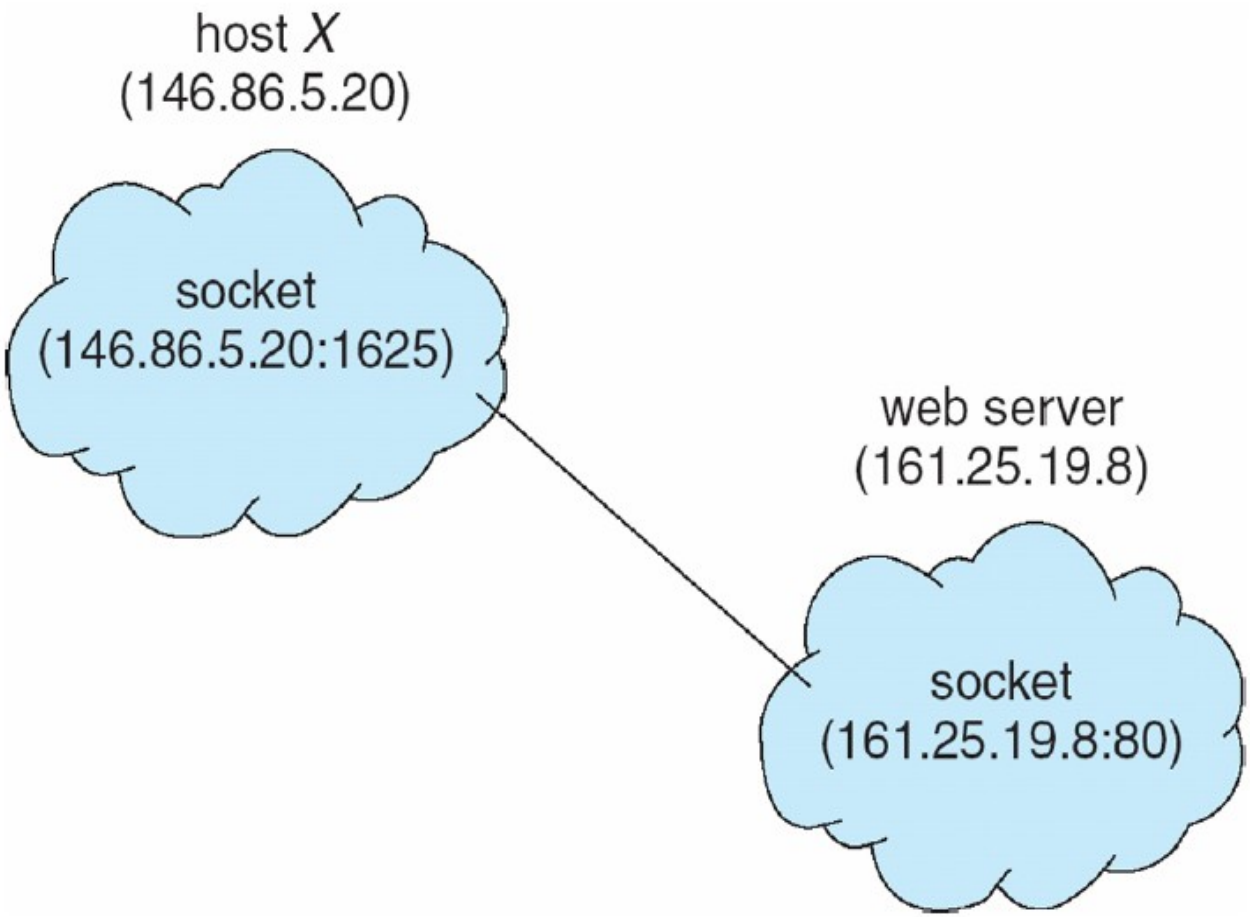
IPC (Comunicación entre Procesos)

Sockets

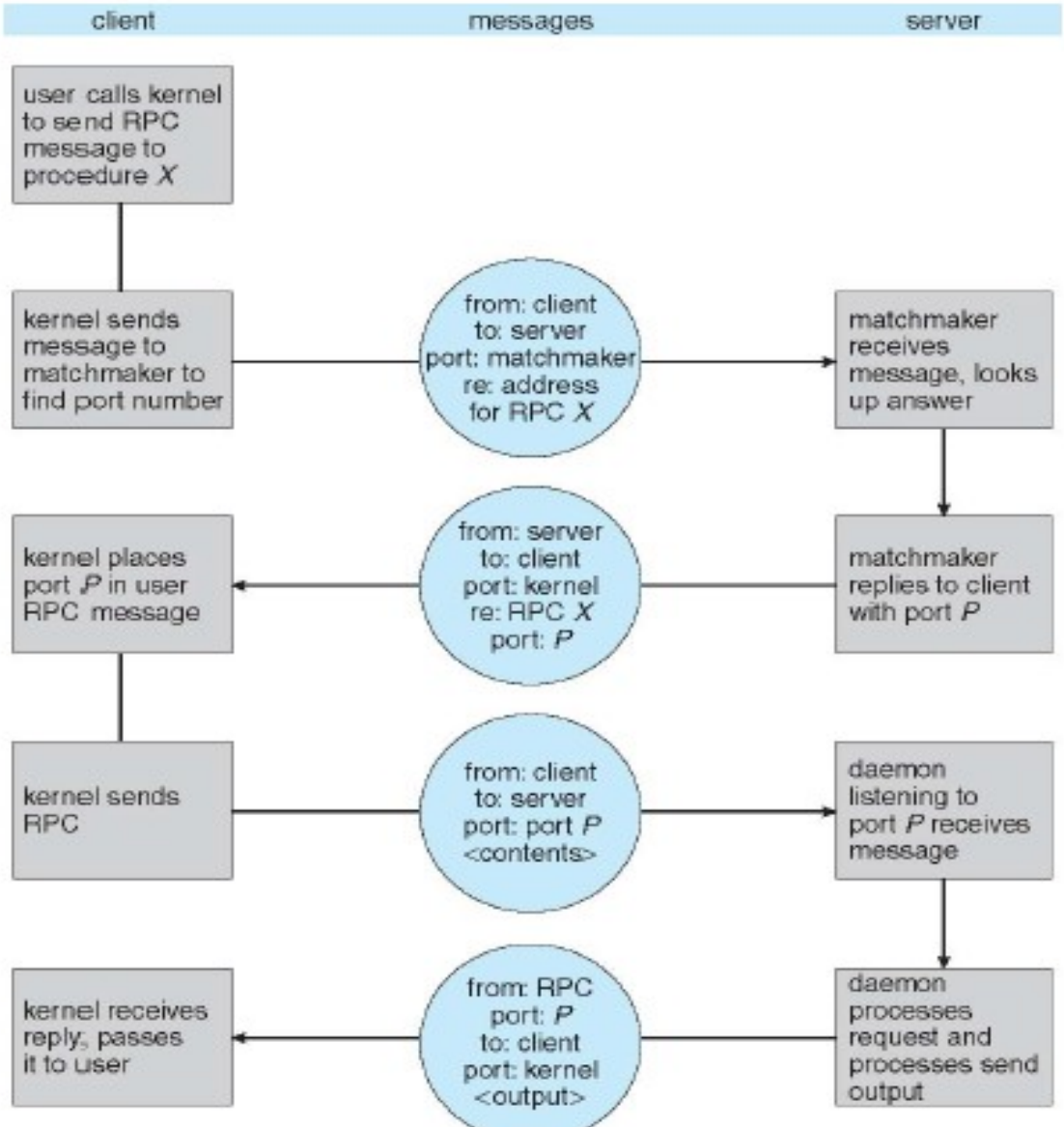
- A socket is defined as an *endpoint for communication*
- Concatenation of IP address and port
- The socket **161.25.19.8:1625** refers to port **1625** on host **161.25.19.8**
- Communication consists between a pair of sockets

Puertos Públicos
Puertos Privados

IPC (Comunicación entre Procesos)



Execution of RPC



Fin clase