Group D

Chapter 3: Processes

Adam Benn

Nitin Pasikanti

Rashed Qazizada

Adam Rashdan

Munish Sharma

Waseem Totonji

Structure

1. Process Concept
2. Process Scheduling
3. Operations on Processes
4. Interprocess Communication
5. Communication in Client–Server Systems

Process Concept

- Program to Process.
- Processes in memory

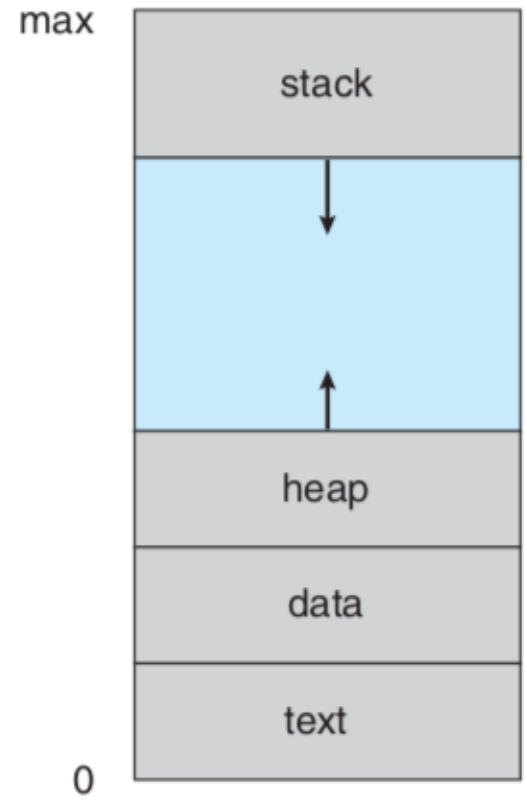


Figure 3.1 Process in memory.

Silberschatz, Abraham, et al. *Operating System Concepts*. 9th ed., Wiley, 2012,p.106

Process State

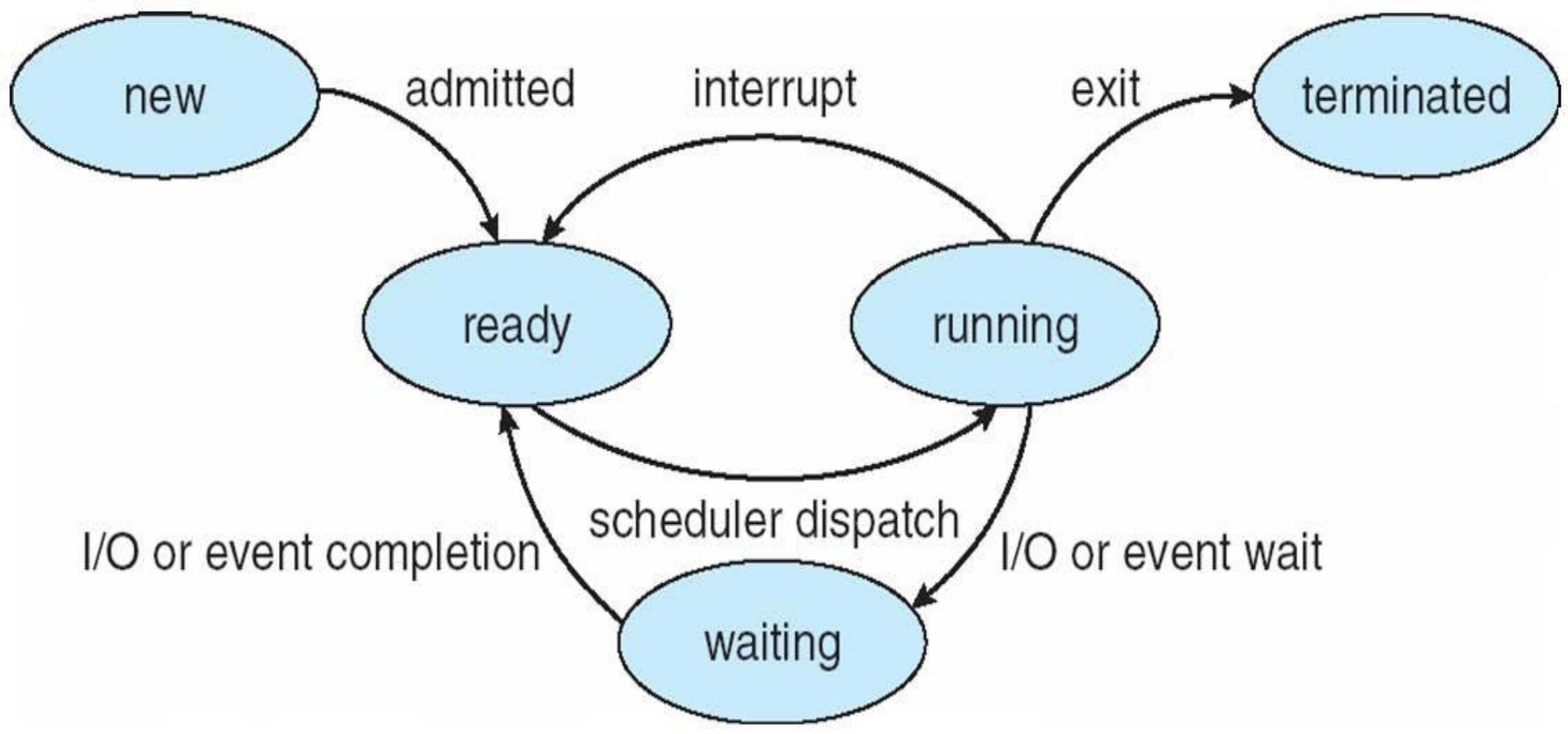


Figure 3.2 Diagram of a process state.

Silberschatz, Abraham, et al. *Operating System Concepts*. 9th ed., Wiley, 2012,p.108

Process Control Block (PCB)

- Process state
- Program counter
- CPU registers
- CPU-scheduling information
- Memory-management information
- Accounting information
- I/O status information

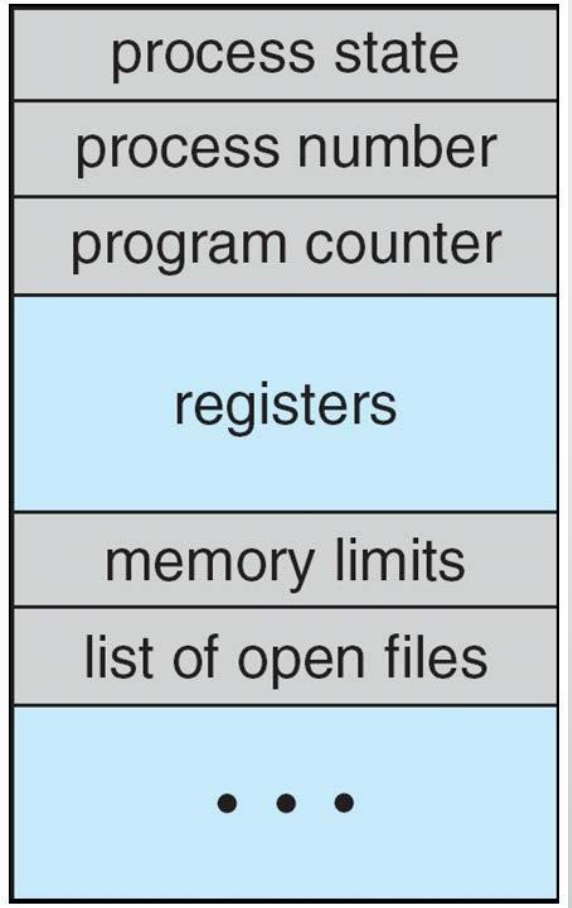


Figure 3.3 Process control block (PCB).

Silberschatz, Abraham, et al. *Operating System Concepts*. 9th ed., Wiley, 2012,p.108

CPU Switch From Process to Process

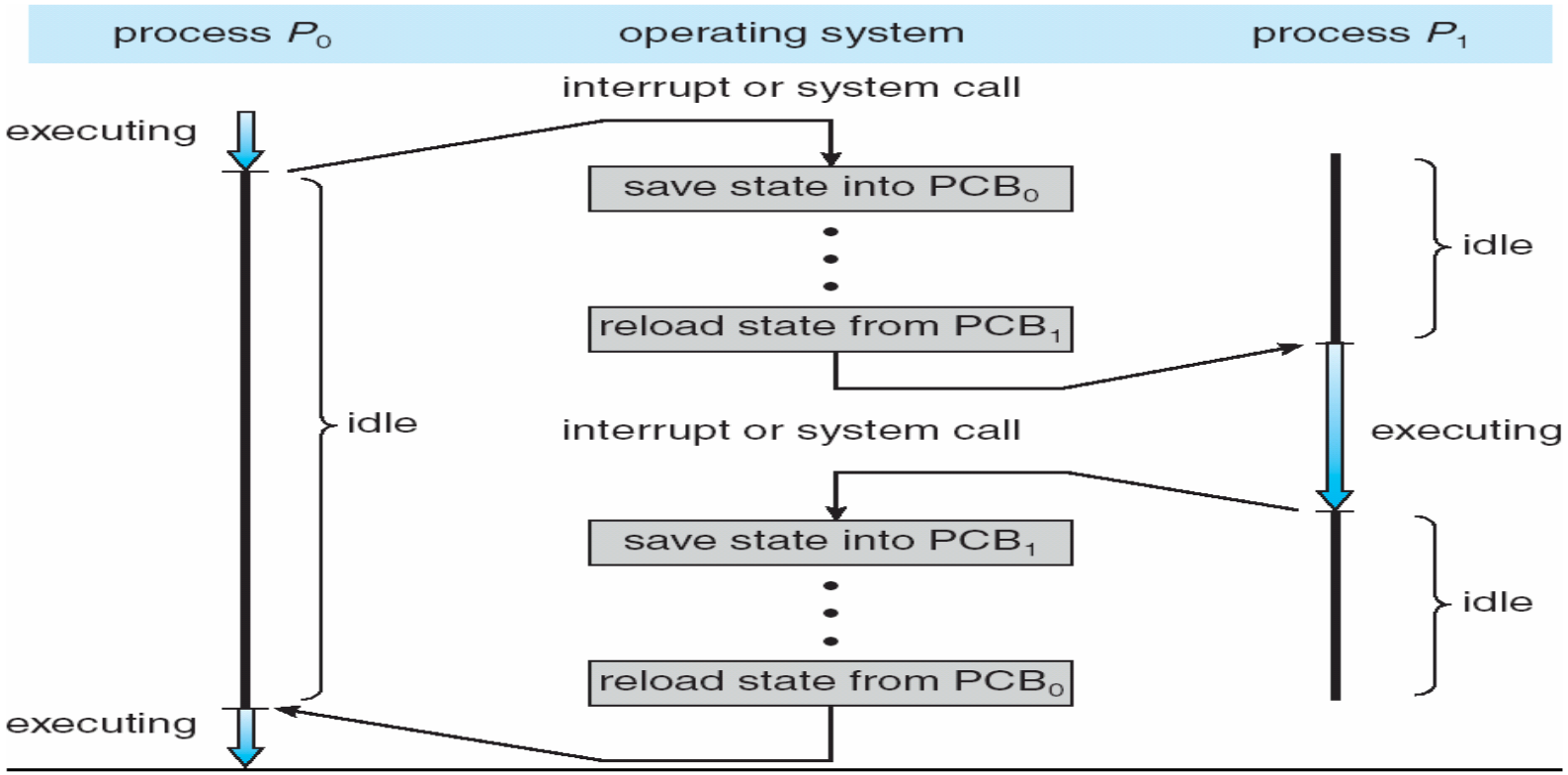


Figure 3.4 Diagram showing CPU switch from process to process.

Silberschatz, Abraham, et al. *Operating System Concepts*. 9th ed., Wiley, 2012,p.109

Process Scheduling

- **Process scheduler**
  - What is the objective of multiprogramming?
  - What is the objective of time sharing?
  - Only one process at a time.

Scheduling queues

- Job Queue
- Ready queue
- Device queues

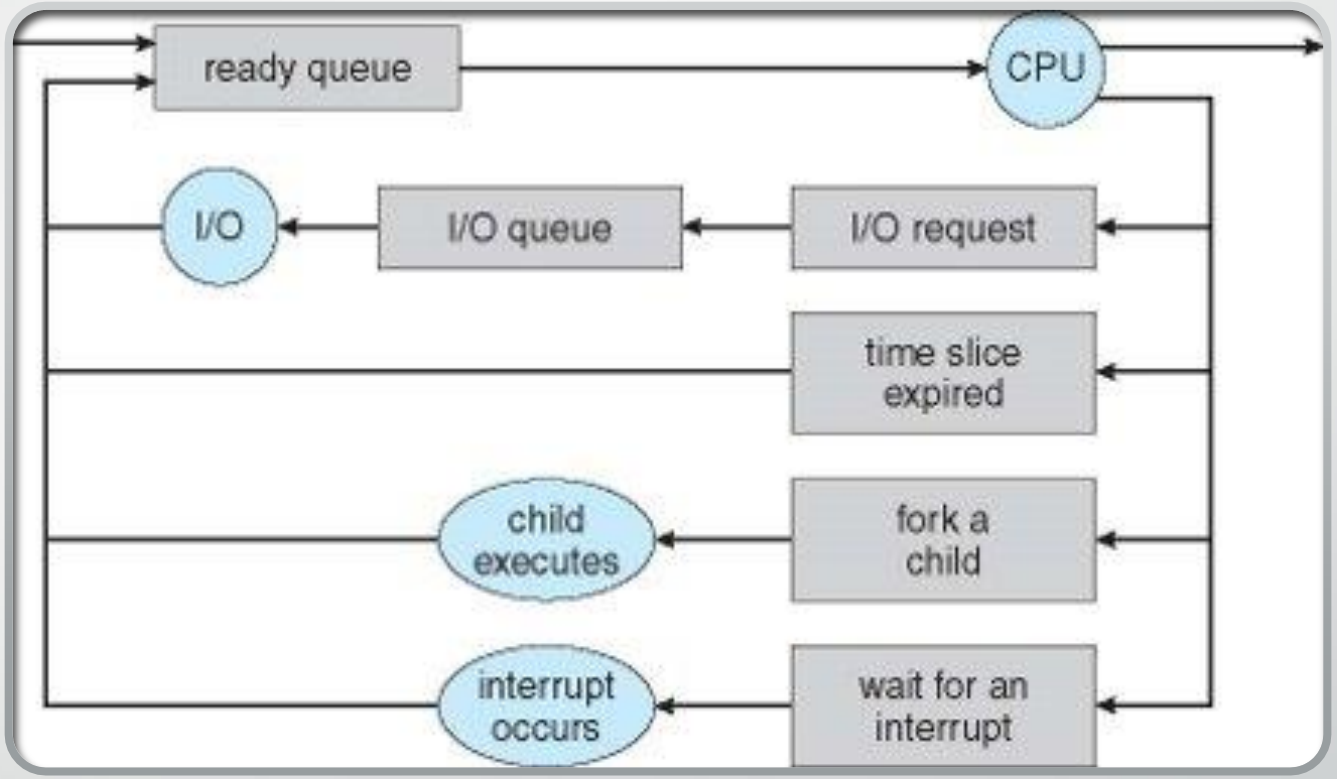


Figure 3.6 Queueing-diagram representation of process scheduling.

Silberschatz, Abraham, et al. *Operating System Concepts*. 9th ed., Wiley, 2012,p.112

Schedulers

- **Short-term scheduler** (or **CPU scheduler**)
- **Long-term scheduler** (or **job scheduler**)
- **Medium-term scheduler**

Context Switch

Saving the state of old process and switching the CPU to another process

Context-Switch time is pure overhead, because the system does no useful work while switching

Operations on Processes

- Process Creation
- Process Termination

Process Creation

A process is a running program

=>Resource sharing

Figure:
A tree of processes on a typical Solaris system.

- When a process creates a new process, two possibilities exists in terms of execution:

- 1. The parent continues to execute concurrently with its children.
- 2. The parent waits until some or all its children have terminated.

- There are also two possibilities in terms of address space of the new process

- 1. The child process is a duplicate of the parent process.
- 2. The child process has a new program loaded into it.

Process Termination

- A process terminates when it finishes executing its final statement and asks the operating system to delete it by using the exit() system call.
- Termination can occur in other circumstances as well.

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

Shared memory

- Process shared commen buffer pool
- The code written by applictaion programmer
- Producer and consumer paradigm

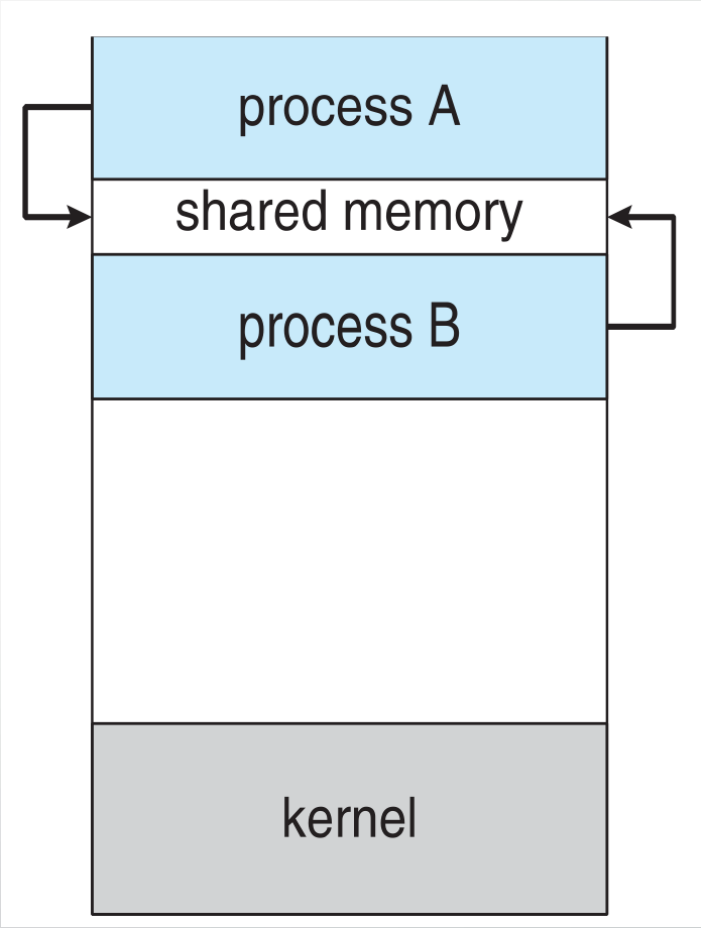


Figure 3.12 Communications models. Shared memory.

Silberschatz, Abraham, et al. *Operating System Concepts*. 9th ed., Wiley, 2012,p.124

Message passing

- Processes communicate without sharing data
- Processes can be fixed / variable size
- Communication links must exist
- There are methods for logically implementing this
  - Direct or indirect communication
  - Symmetric or asymmetric communication
  - Automatic orexplicit buffering

Message passing

- **Direct Communication**
  - Processes must name each other explicitly
  - Properties of communication link
- **Indirect Communication**
  - Messages are directed and received from mailboxes (also referred to as ports)
  - Properties of communication link

Synchronization

- **Blocking** is considered **synchronous**
  - **Blocking send** -- the sender is blocked until the message is received
  - **Blocking receive** -- the receiver is blocked until a message is available
- **Non-blocking** is considered **asynchronous**
  - **Non-blocking send** -- the sender sends the message and continue
  - **Non-blocking receive** -- the receiver receives:
    - A valid message
    - Null message

Buffering

- Queue of messages attached to the link.
- implemented in one of three ways
  - Zero capacity – no messages are queued on a link. Sender must wait for receiver (rendezvous)
  - Bounded capacity – finite length of *n* messages Sender must wait if link full
  - Unbounded capacity – infinite length Sender never waits

Local Procedure Calls (OS example) (LPCs)

- System call sequence to copy the contents of one file to another file

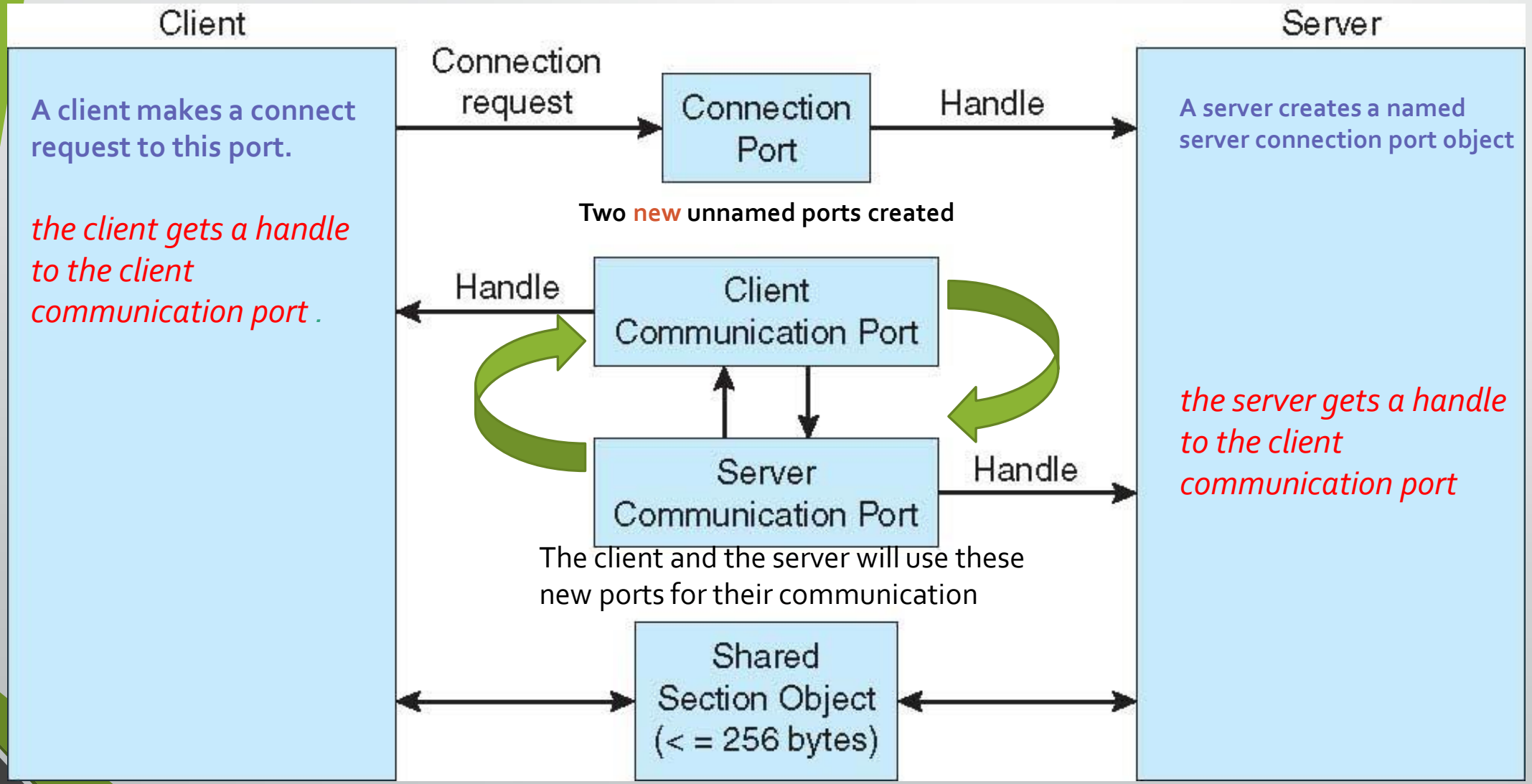


Figure 3.19 Advanced local procedure calls in Windows.

Silberschatz, Abraham, et al. *Operating System Concepts*. 9th ed., Wiley, 2012,p.136

Sockets

One of the strategies for communication in client–server systems

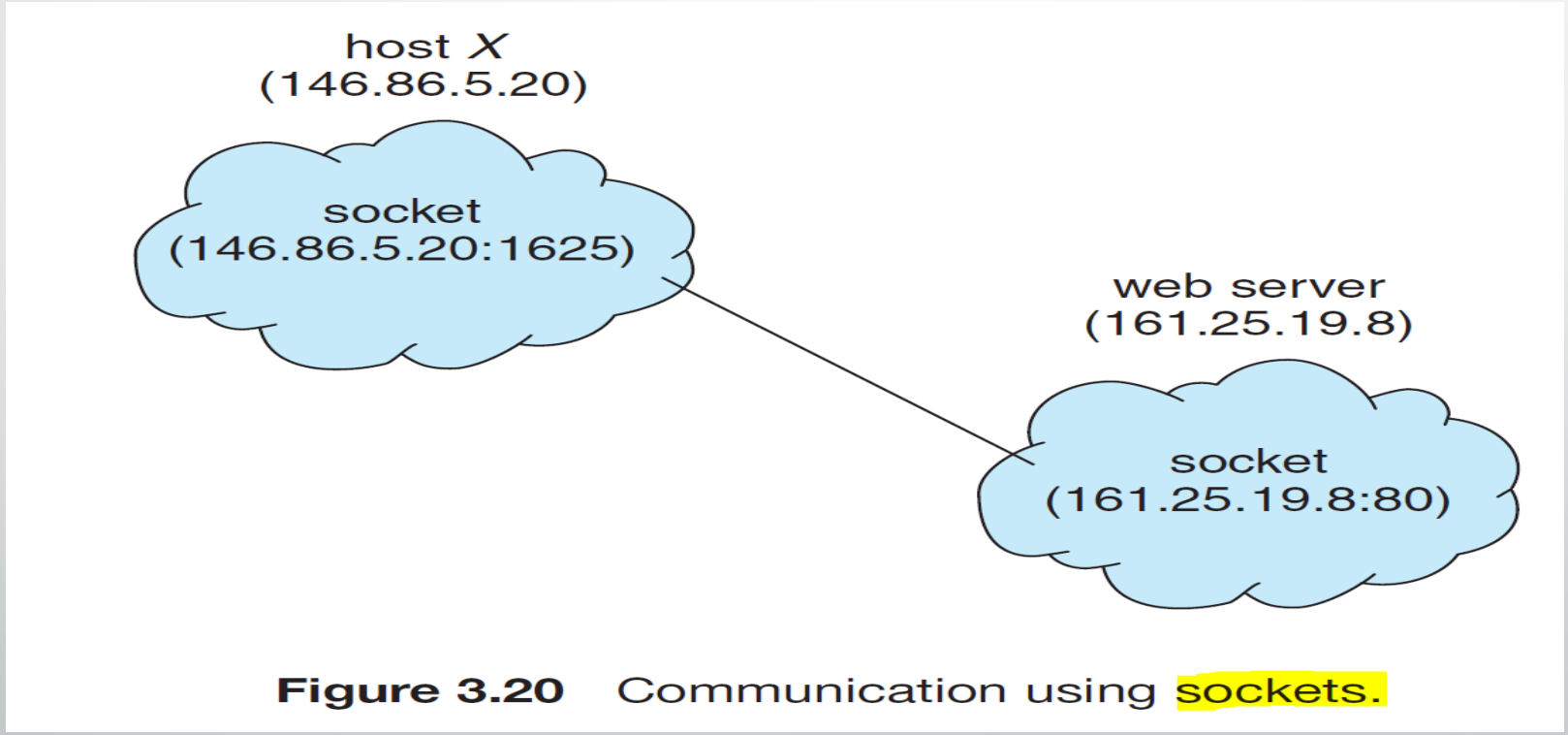
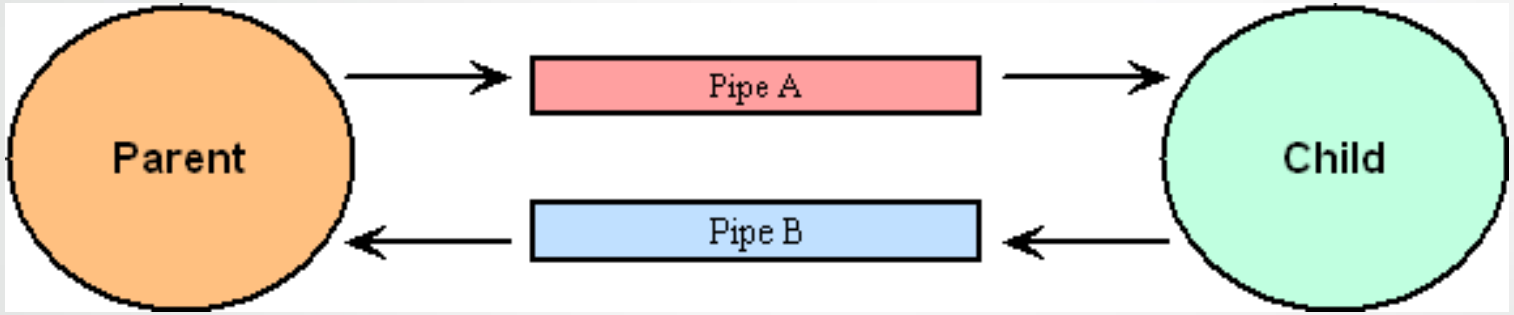


Figure 3.20 Communication using sockets.

Another methods of communication between two processes called **Pipe**

Pipe mechanism : Information stream



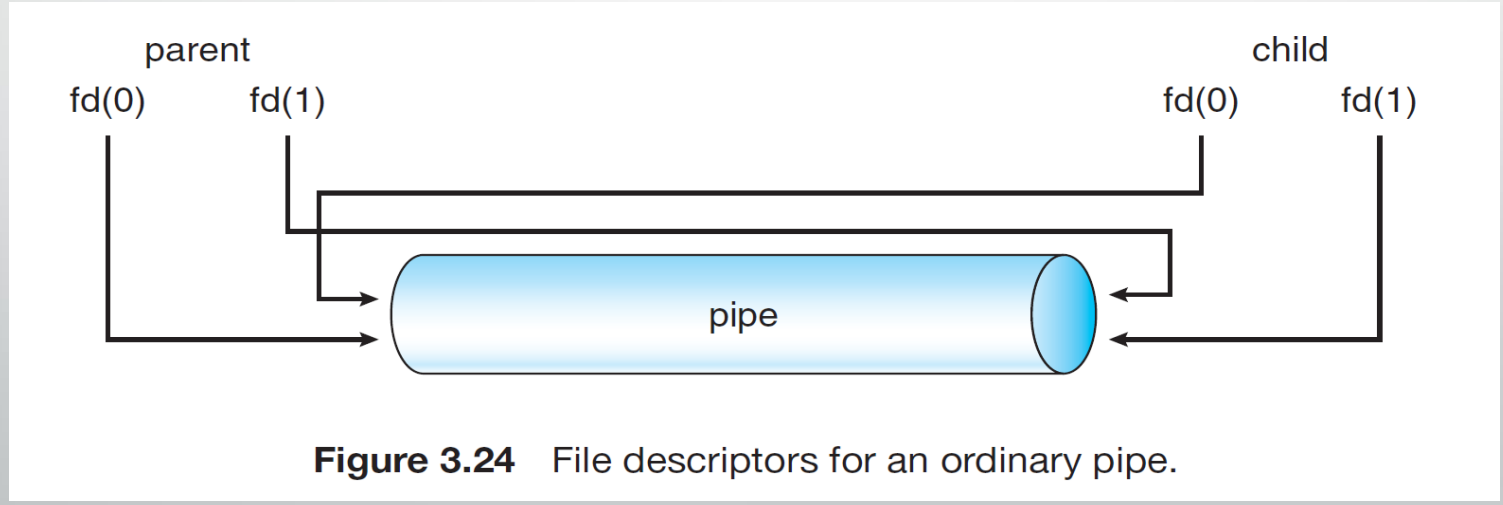


Figure 3.24 File descriptors for an ordinary pipe.

Thank you!