Principles of Operating Systems

CS 446/646

1. Introduction to Operating Systems

a. Role of an O/S

b. O/S History and Features

c. **Types of O/S**

- ✓ Mainframe systems
- ✓ Desktop & laptop systems
- ✓ Parallel systems
- ✓ Real-time systems

d. **Major O/S Components**

e. **System Calls**

f. **O/S Software Architecture**

g. **Examples of O/S**

1.c Types of Operating Systems

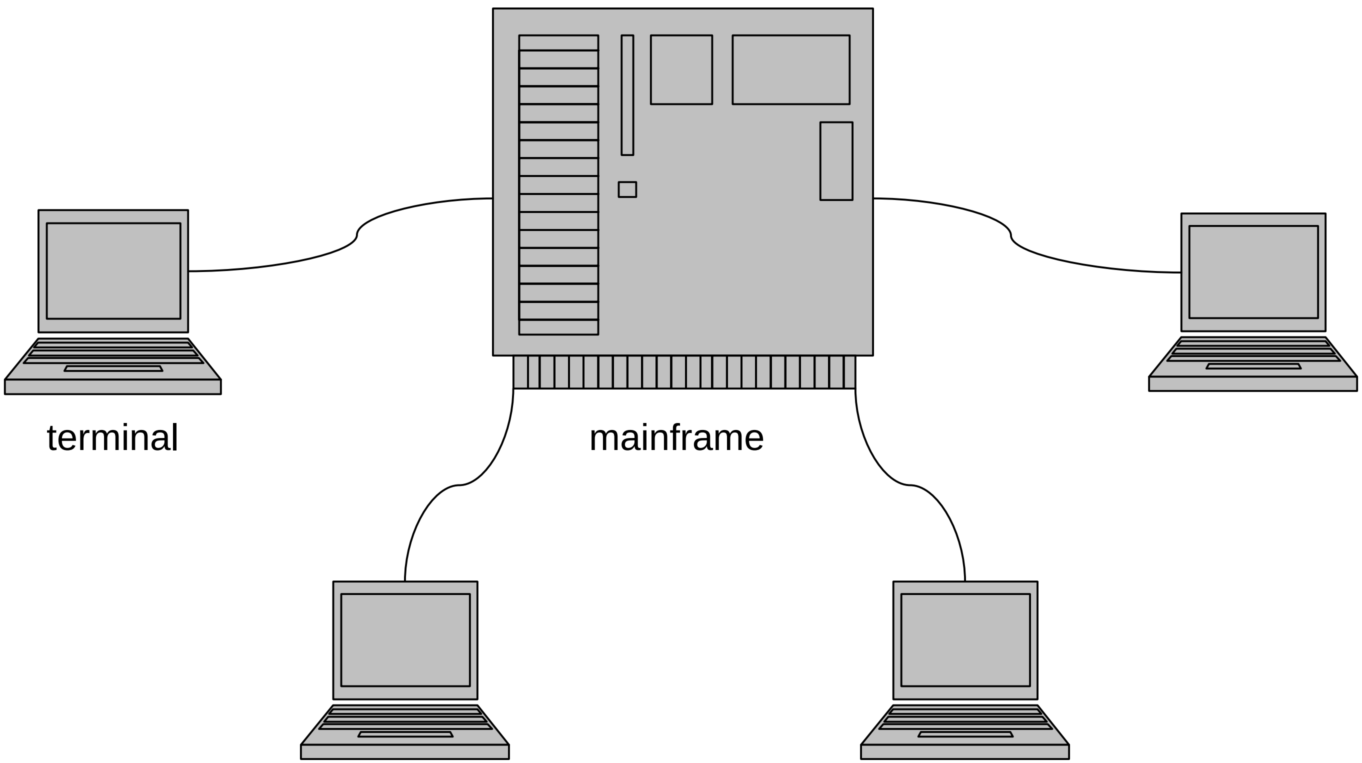
Note: The historical evolution of computers have left us with a great variety of O/S types, not all of which widely known.

In this section, we briefly touch upon a few of them. Some aspects of these systems will be addressed more specifically later in the course.

1.c Types of Operating Systems

Mainframe systems

- Mainframes have a centralized architecture
  - ✓ one big computer connected to many terminals, generally pure I/O keyboard-display devices without CPU ("dumb terminals")



1.c Types of Operating Systems
Mainframe systems

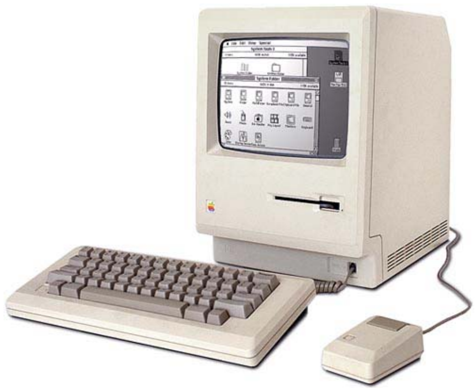
- **Characteristics of mainframe systems**
  - ✓ the first computers used to tackle various applications and still found today in corporate data centers
  - ✓ room-sized, high I/O capacity, reliability, security, tech support
  - ✓ mainframes focus on I/O-bound business data applications ("supercomputers" focus on CPU-bound scientific calculations: see parallel systems)
- **Mainframes provide three main functions**
  - ✓ batch processing: insurance claims, store sales reporting, etc.
  - ✓ transaction processing: credit card, bank account, etc.
  - ✓ time-sharing (sessions): multiple users querying a database

1.c Types of Operating Systems

Desktop & laptop systems

- Personal computers (see 1.b)

IBM PC XT (1981)



Apple Macintosh (1984)

Dell Dimension XPS (2005)

Dell Latitude D410 (2005)

1.c Types of Operating Systems

Desktop & laptop systems

- A very brief overview of the PC operating system zoo

Linux Distributions	BSD Family	Apple & Mac History	Windows History
Debian	4.3BSD Reno	Apple DOS	MS-DOS
Fedora Core (Red Hat)	4.4BSD	ProDOS	Windows 3.0
Gentoo	BSD/OS	Lisa OS	Windows 95 (4.0)
Knoppix	FreeBSD	Mac OS	Windows 98 (4.1)
Linspire	DragonFlyBSD	Darwin	Windows Me (4.2)
Mandrakelinux	NetBSD	Mac OS X	Windows NT 4.0
Slackware	OpenBSD		Windows 2000 (NT 5.0)
SUSE			Windows XP (NT 5.1)
Yellow Dog			Windows Vista (2006)

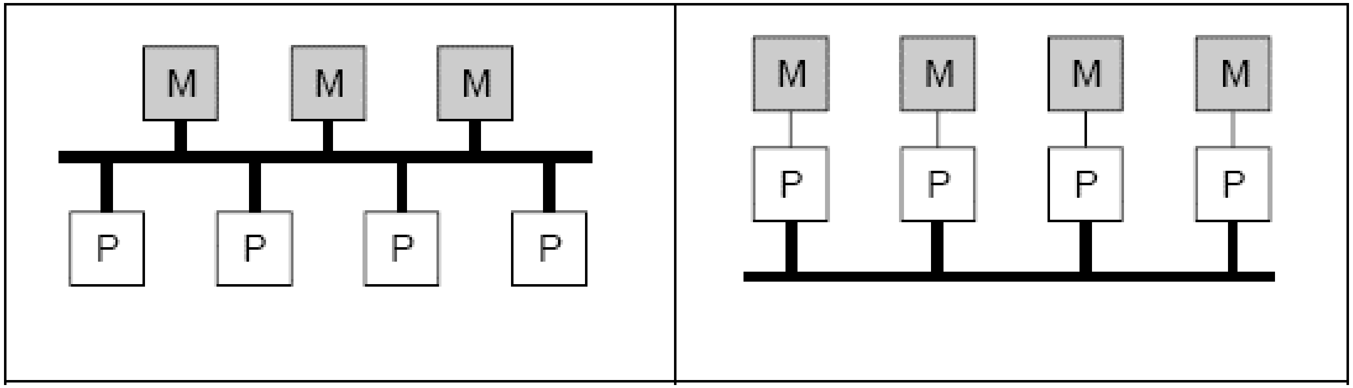
1.c Types of Operating Systems

Parallel systems

- Flynn's 1972 taxonomy (crude, but often used)
  - ✓ SISD = Single Instruction stream, Single Data stream
    - classical sequential "von Neumann" machine
    - your regular PC with a single (pipelined) scalar processor
  - ✓ SIMD = Single Instruction stream, Multiple Data streams
    - a single control unit issues 1 instruction at a time; multiple ALUs carry it out on multiple data sets simultaneously
    - characterizes the vector supercomputers or array processors used in scientific computing
  - ✓ MIMD = Multiple Instruction streams, Multiple Data streams
    - multiple independent CPUs operating within larger system

1.c Types of Operating Systems
Parallel systems

- Subclassification of MIMD architectures
  - “multiprocessors” = shared memory
    - all CPUs access the same physical address space
  - “multicomputers” = private individual memory
    - each CPU has a direct connection to its own local memory
    - homogeneous (clusters) vs. heterogeneous (LAN, WAN)

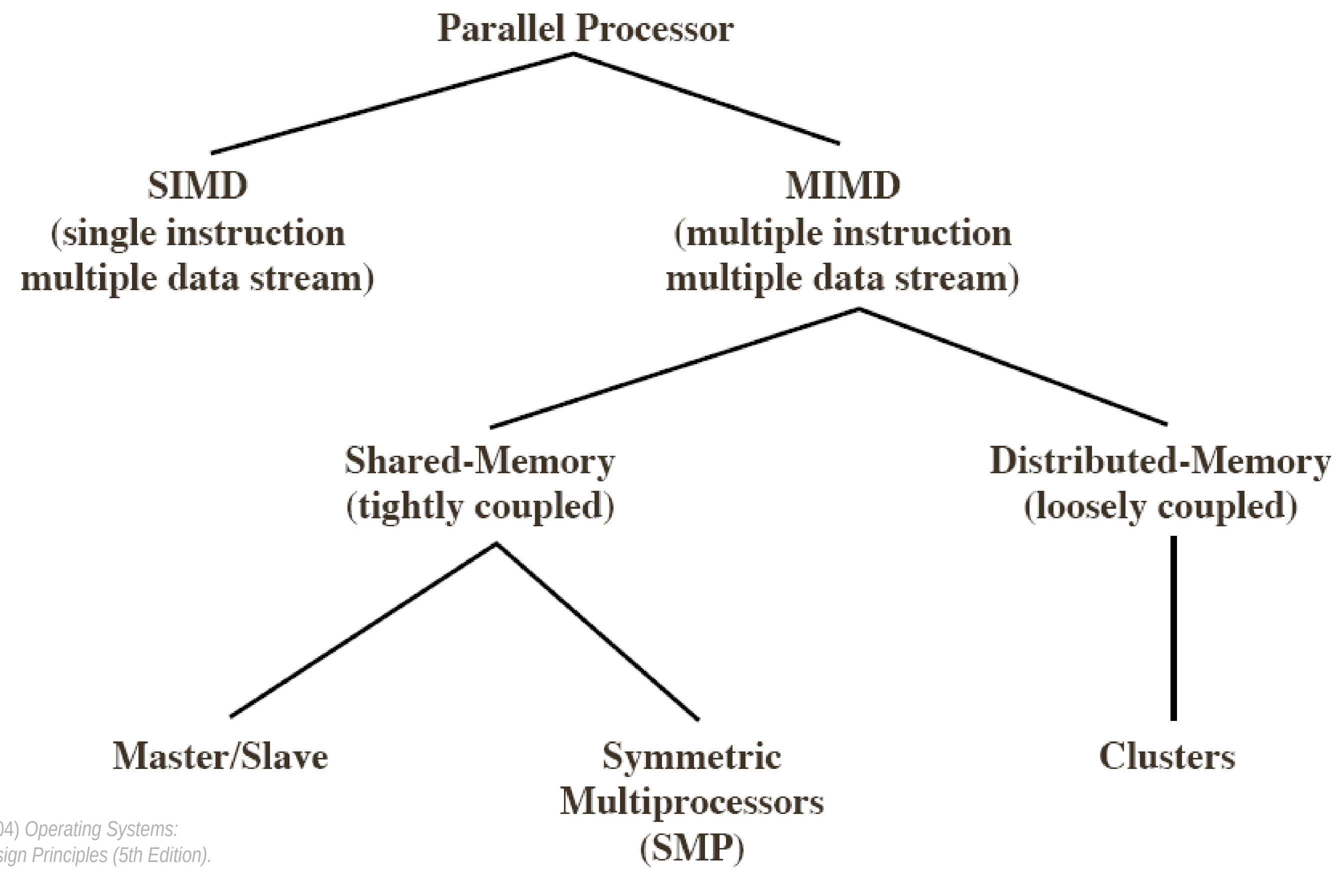


Tanenbaum, A. S. (2002) *Distributed Systems.*

Shared memory and private memory organization in distributed computer systems

1.c Types of Operating Systems
Parallel systems

- Parallel architectures form a diverse and complex field

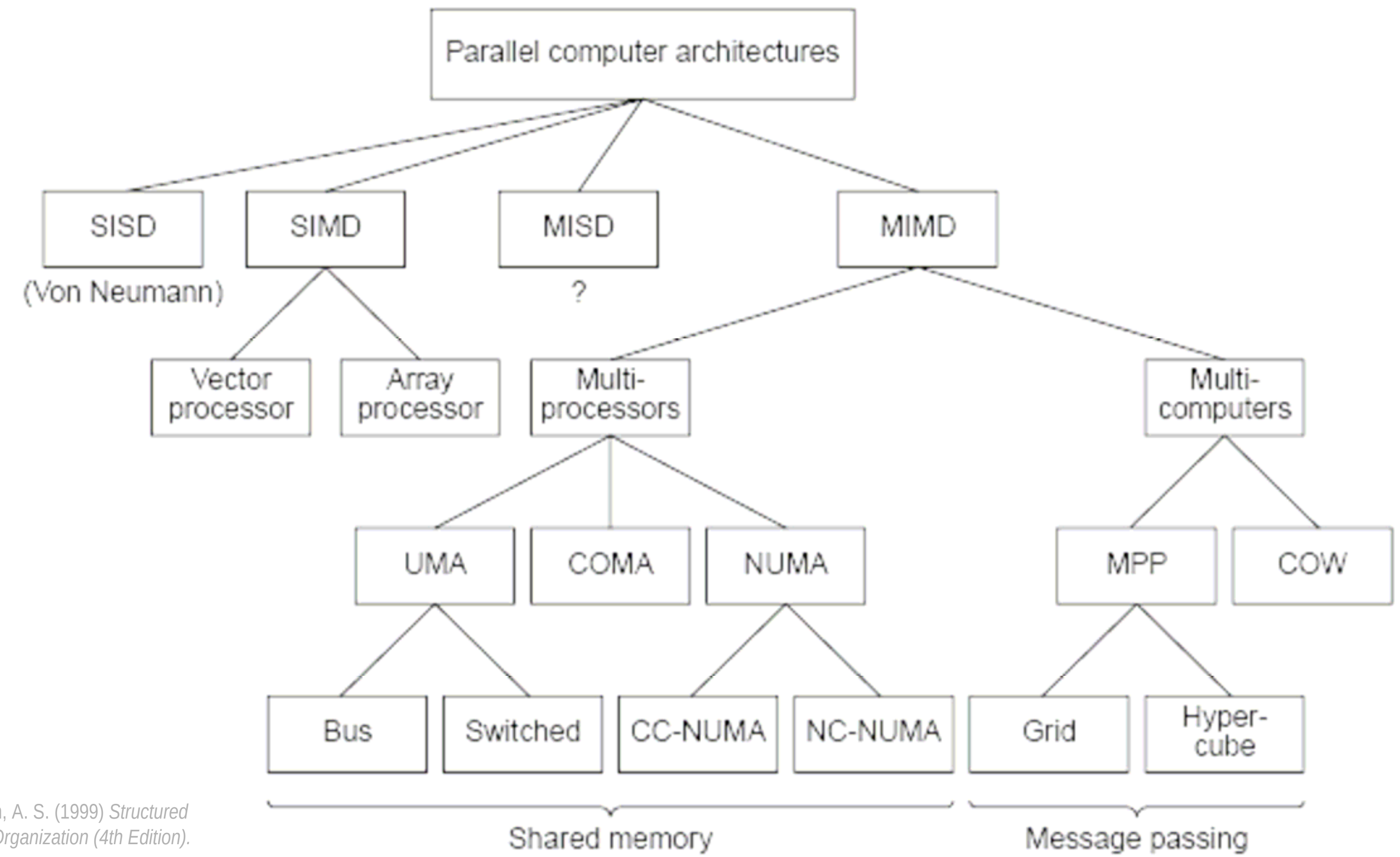


Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

A possible (simplified) taxonomy of parallel computers

1.c Types of Operating Systems
Parallel systems

- Parallel architectures form a diverse and complex field



Tanenbaum, A. S. (1999) *Structured Computer Organization (4th Edition).*

A possible (refined) taxonomy of parallel computers

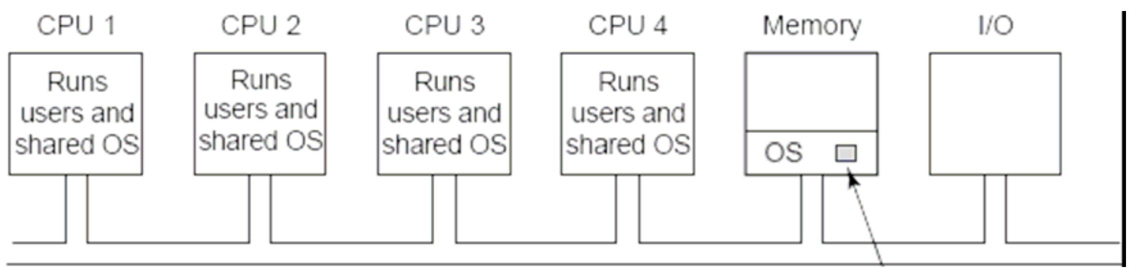
1.c Types of Operating Systems

Parallel systems

- Software level: distributed O/S much like traditional O/S
  - ✓ O/S acts as a resource manager for the underlying hardware
  - ✓ O/S attempts to hide the underlying intricacies and heterogeneity by providing a virtual machine
- Two types of O/S for parallel architectures
  - ✓ tightly-coupled O/S a.k.a. Distributed Operating System (DOS)
    - in multiprocessors and homogeneous multicomputers: tries to maintain a single, global view of the resources
  - ✓ loosely-coupled O/S a.k.a. Network Operating System (DOS)
    - in heterogeneous multicomputers: collection of independent operating systems working together

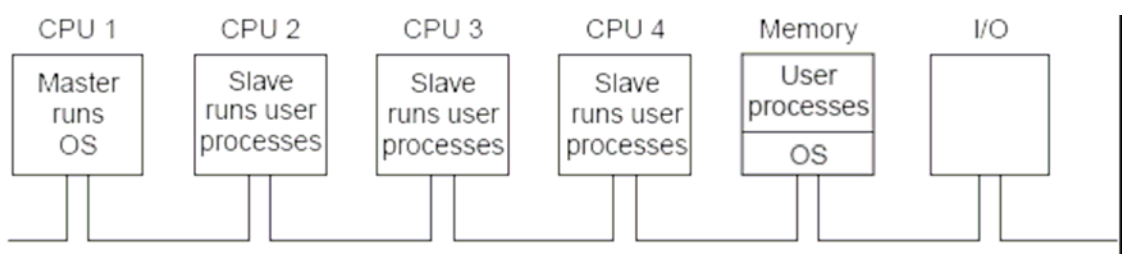
1.c Types of Operating Systems
Parallel systems

- Subtypes of tightly coupled operating systems
  - ✓ SMP = Symmetric MultiProcessing
    - all CPUs are peers and concurrently run the same copy of O/S in memory



Tanenbaum, A. S. (2001) *Modern Operating Systems (2nd Edition).*

  - ✓ asymmetric or "master-slave" multiprocessing
    - one CPU runs the O/S, the others ask for tasks to do



1.c Types of Operating Systems

Real-time systems

- Real-time systems satisfy specific time requirements
  - ✓ systems controlling scientific experiments, medical imaging systems, industrial control systems, some display systems
- "Hard" real-time: critical tasks are guaranteed on time
  - ✓ secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
  - ✓ conflicts with time-sharing systems and virtual memory delays
- "Soft" real-time: critical tasks just get higher priority
  - ✓ ok with time-sharing; not used in industrial robotics
  - ✓ more useful in applications requiring tight but not strict response times (multimedia, virtual reality, robotic exploration)

Principles of Operating Systems
CS 446/646

1. Introduction to Operating Systems

a. Role of an O/S

b. O/S History and Features

c. **Types of O/S**
   - ✓ Mainframe systems
   - ✓ Desktop & laptop systems
   - ✓ Parallel systems
   - ✓ Real-time systems

d. **Major O/S Components**

e. **System Calls**

f. **O/S Software Architecture**

g. **Examples of O/S**

Principles of Operating Systems
CS 446/646

1. Introduction to Operating Systems

a. Role of an O/S

b. O/S History and Features

c. Types of O/S

d. **Major O/S Components**
   - ✓ Processes
   - ✓ Memory management
   - ✓ CPU scheduling
   - ✓ Input/output
   - ✓ File system

e. **System Calls**

f. **O/S Software Architecture**

g. **Examples of O/S**

1.d Major Operating System Components

Note: There is no definitive list of the "components" of an O/S. We give here only a brief overview of the most common subdivisions of an O/S and the services they are responsible for. Also, the components' functions often intersect so there is no one-to-one match between components and O/S software modules.

1.d Major Operating System Components

Processes

- A process is the activity of executing a program

Pasta for six

- boil 1 quart salty water
- stir in the pasta
- cook on medium until "al dente"
- serve

thread of execution

Program

Process

1.d Major Operating System Components

Processes

- It can be interrupted to let the CPU execute a higher-priority process

Pa

- boil wat
- stir
- coo unt
- serv

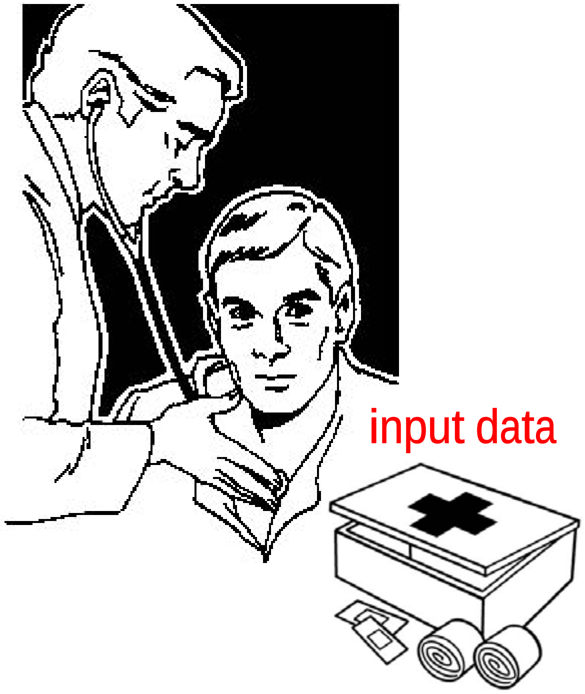
First aid

- Get the first aid kit
- Check pulse
- Clean wound with alcohol
- Apply band aid

thread of execution

Program

CPU (changes hat to "doctor")



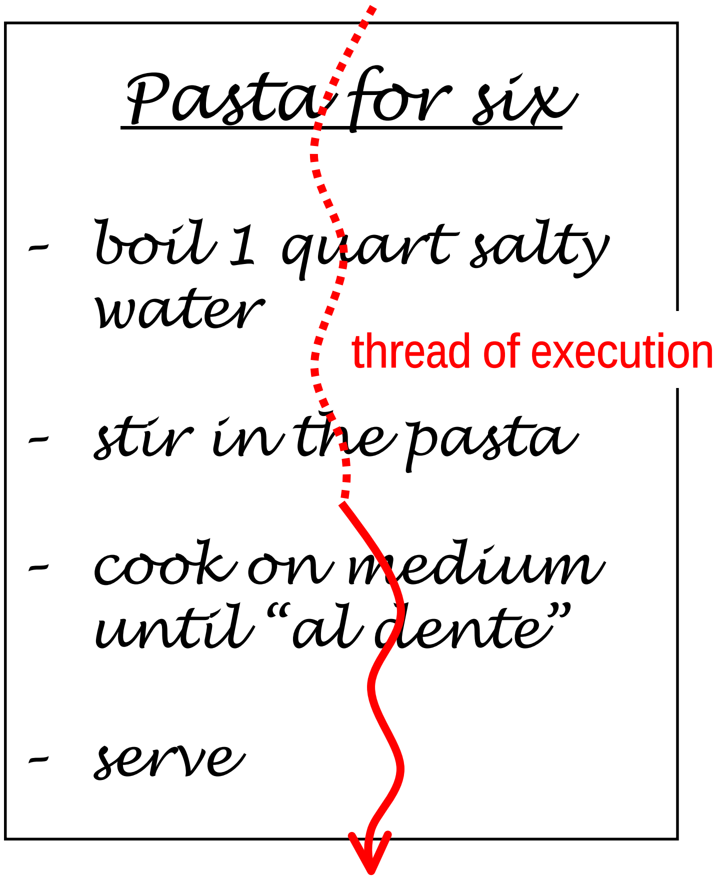
input data

Process

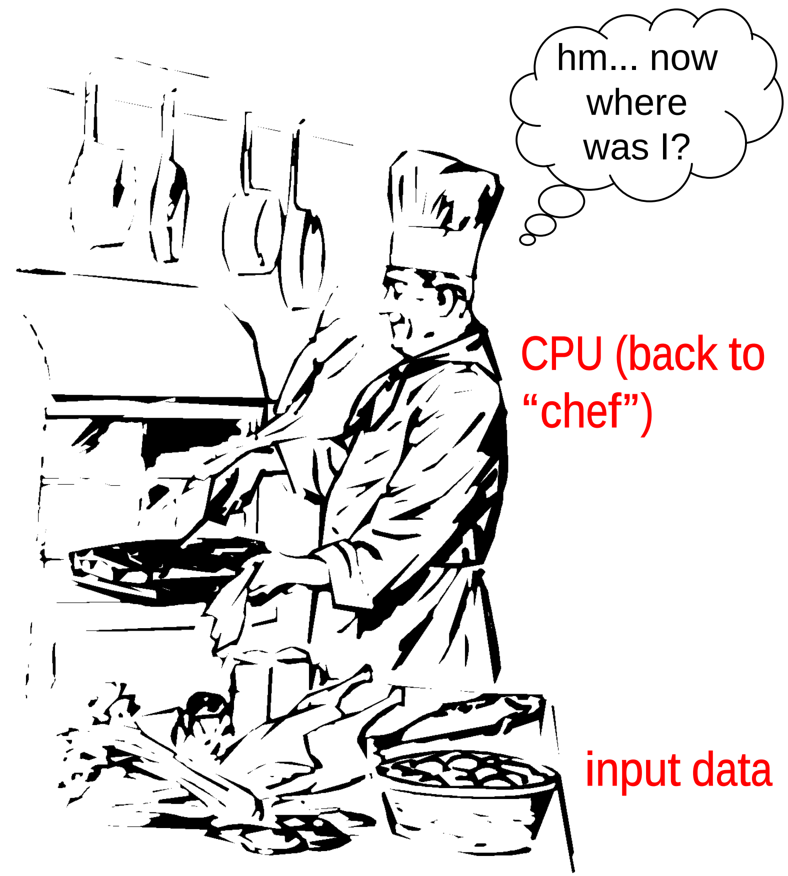
1.d Major Operating System Components

Processes

- . . . and then resumed exactly where the CPU left off



Program

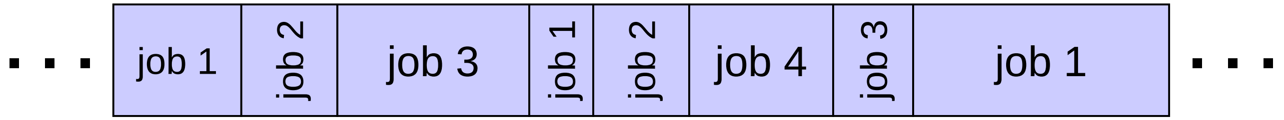


Process

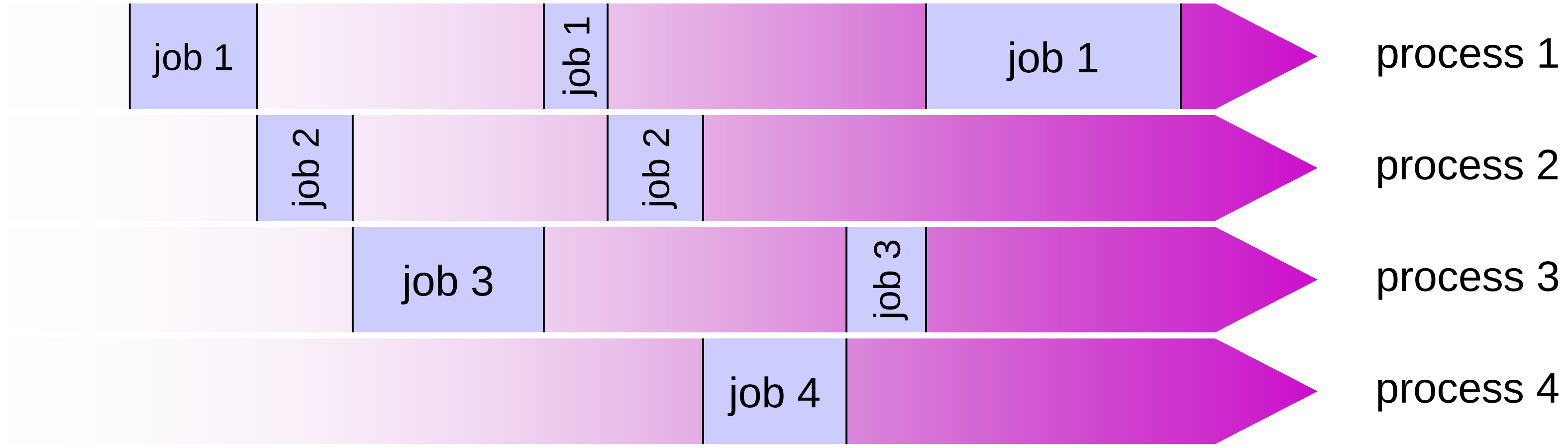
1.d Major Operating System Components

Processes

- Multitasking gives the illusion of parallel processing (independent virtual program counters) on one CPU



(a) Multitasking from the CPU's viewpoint



(b) Multitasking from the processes' viewpoint = 4 virtual program counters

Pseudoparallelism in multitasking

1.d Major Operating System Components
Processes

- A process is an encapsulated unit of activity containing
  - ✓ a sequential thread of execution
  - ✓ a current state
  - ✓ an associated set of system resources (CPU time, memory, files, I/O devices), needed to accomplish its task
- The concept of “process” is a unifying abstraction
  - ✓ multiprogramming, resource-sharing, time-sharing and real-time systems raised complex coordination problems
  - ✓ thus, a higher-level model than jobs and interrupts was needed
  - → the concept of “process” was first introduced in MULTICS in the 1960’s and is now the most important component of modern O/S

1.d Major Operating System Components

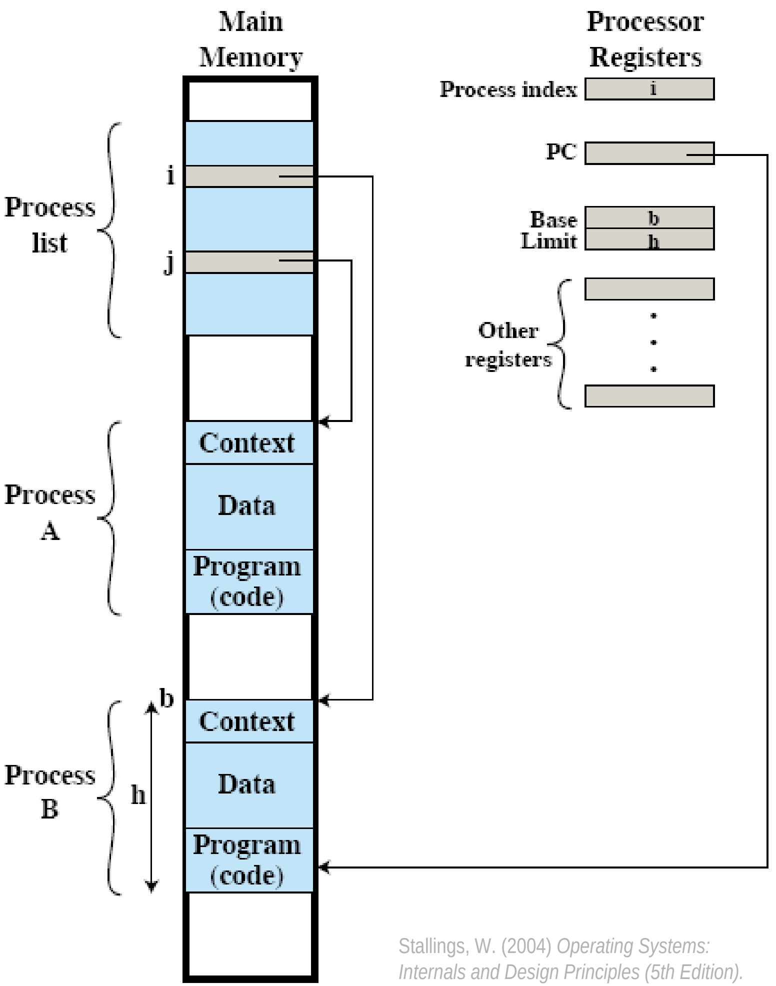
Processes

- Quick preview: a process image consists of three components

user address space:

1. an executable <u>program</u>
2. the associated <u>data</u> needed by the program

3. the execution <u>context</u> of the process, which contains all information the O/S needs to manage the process (id, state, CPU registers, stack, etc.)



Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

Typical process implementation

1.d Major Operating System Components
Processes

- Chart of Operating System Responsibilities

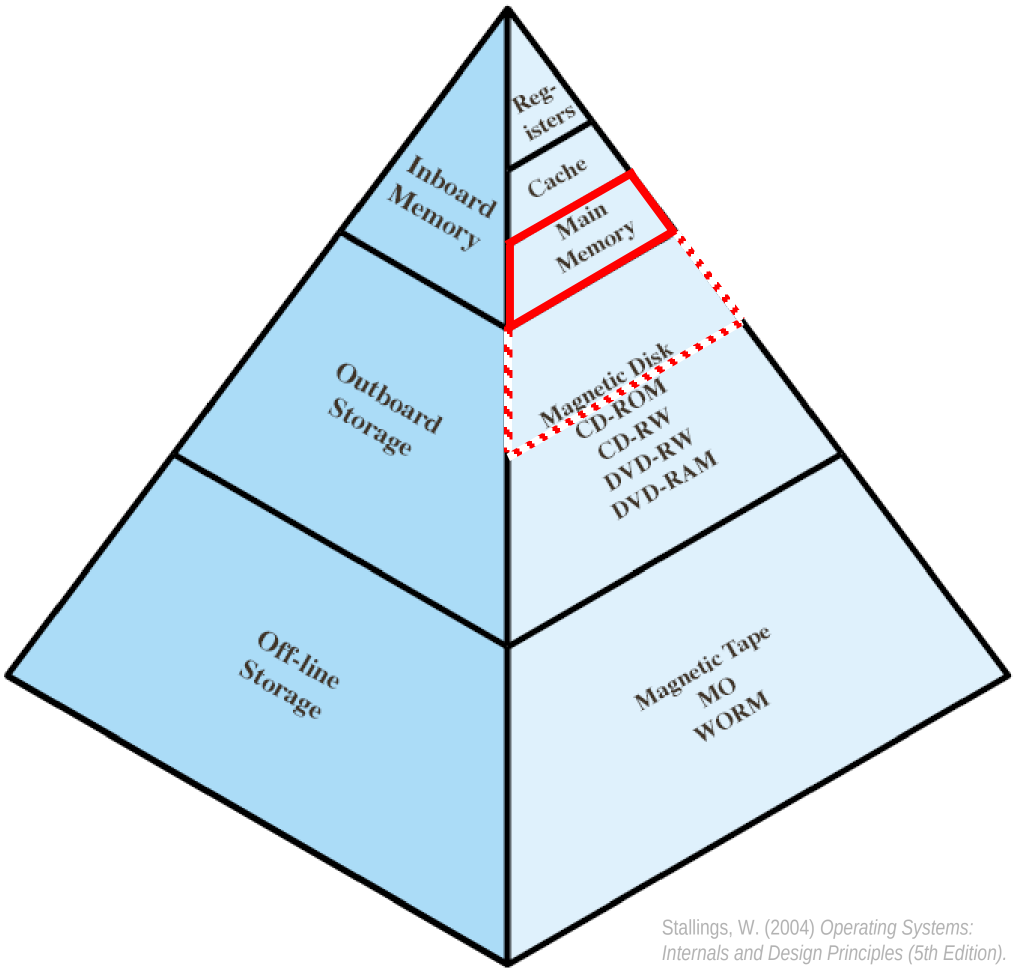
§A – The O/S is responsible for managing processes

- ✓ the O/S creates & deletes processes
- ✓ the O/S suspends & resumes processes
- ✓ the O/S provides mechanisms for process synchronization
- ✓ the O/S provides mechanisms for interprocess communication
- ✓ the O/S provides mechanisms for deadlock handling

1.d Major Operating System Components

Memory management

- **Main memory**
  - ✓ large array of words or bytes, each with its own address
  - ✓ repository of quickly accessible data shared by the CPU and I/O devices
  - ✓ volatile storage that loses its contents in case of system failure



Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

The memory hierarchy

1.d Major Operating System Components

Memory management

- Chart of Operating System Responsibilities

§B – The O/S is responsible for an efficient and orderly control of storage allocation

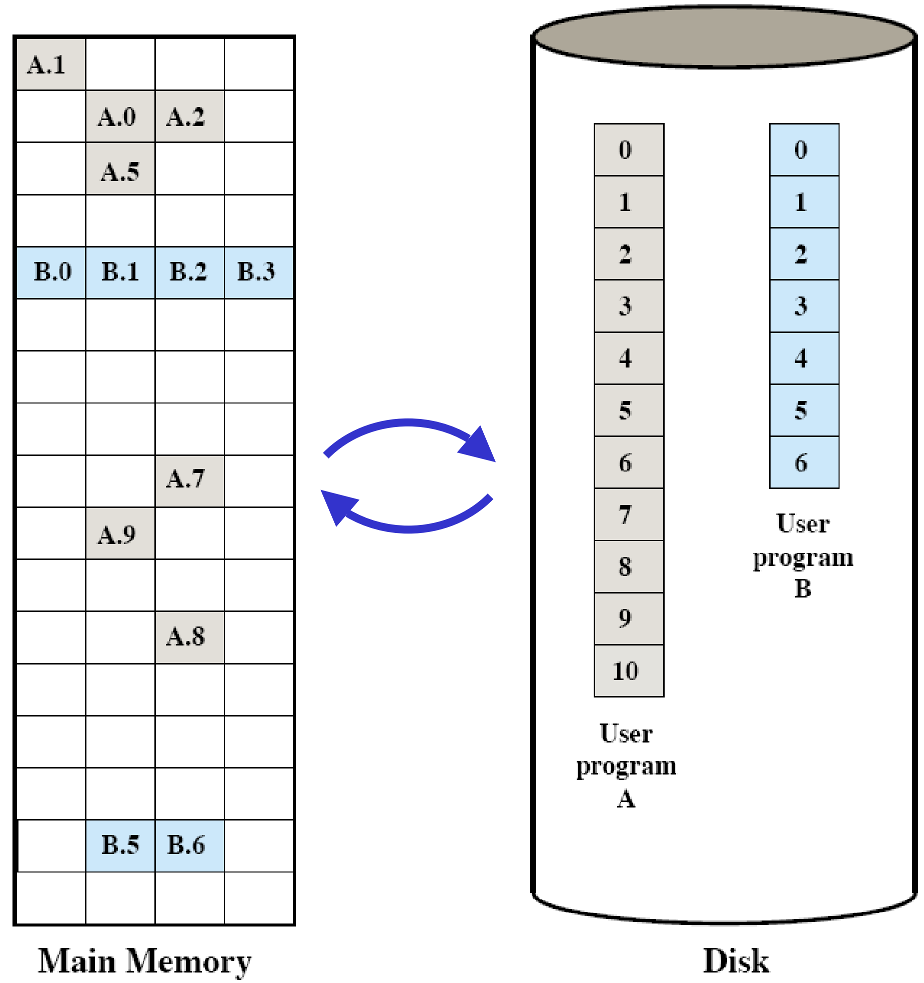
- ✓ the O/S ensures process isolation: it keeps track of which parts of memory are currently being used and by whom
- ✓ the O/S allocates and deallocates memory space as needed: it decides which processes to load or swap out
- ✓ the O/S regulates how different processes and users can sometimes share the same portions of memory
- ✓ the O/S transfers data between main memory and disk and ensures long-term storage

1.d Major Operating System Components

Memory management

Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

- Quick preview: virtual memory is the way
  - ✓ another abstraction
  - ✓ allows memory access by *logical* instead of physical addressing
  - ✓ programs are divided into **pages** that are swapped in and out when needed
  - ✓ not all program pages are loaded at all times

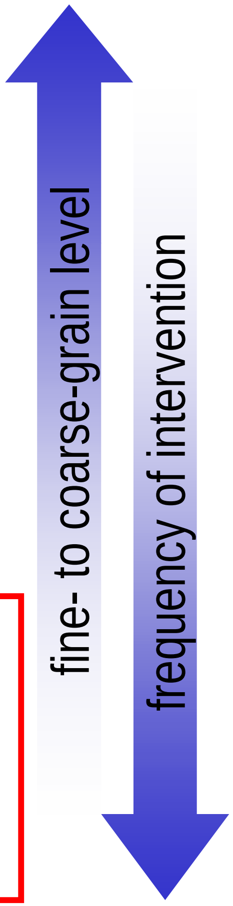


Virtual memory concepts

1.d Major Operating System Components

CPU scheduling

- **Long-term scheduling**
  - ✓ the decision to add a program to the pool of processes to be executed
- **Medium-term scheduling**
  - ✓ the decision to add to the number of processes that are partially or fully in main memory ("swapping")
- **Short-term scheduling = CPU scheduling**
  - ✓ the decision as to which available processes in memory are to be executed by the processor ("dispatching")
- **I/O scheduling**
  - ✓ the decision to handle a process's pending I/O request



1.d Major Operating System Components

CPU scheduling

- Chart of Operating System Responsibilities

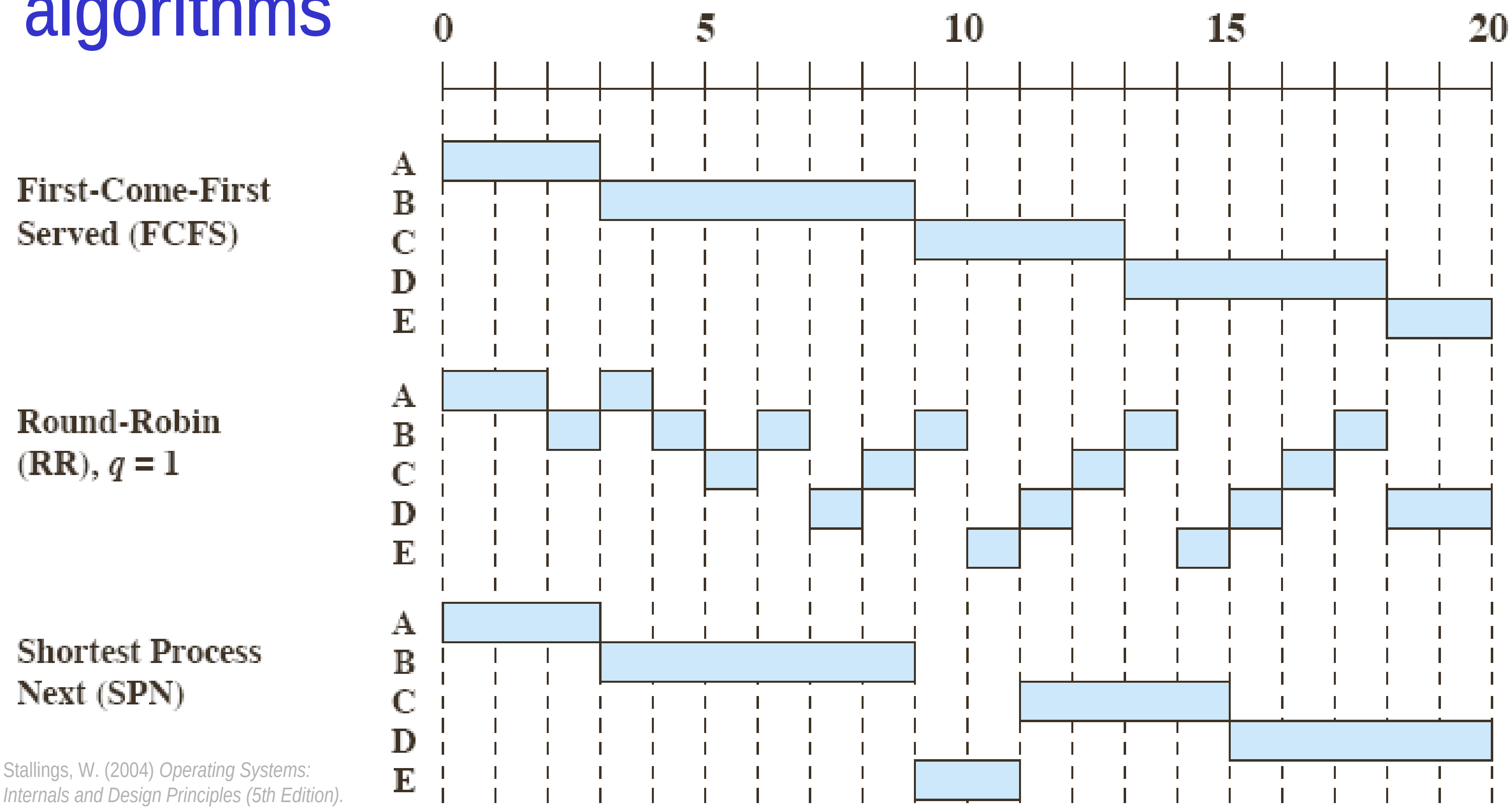
§C – The O/S is responsible for efficiently using the CPU and providing the user with short response times

- ✓ the O/S decides which available processes in memory are to be executed by the processor
- ✓ the O/S decides what process is executed when and for how long, also reacting to external events such as I/O interrupts
- ✓ the O/S relies on a scheduling algorithm that attempts to optimize CPU utilization, throughput, latency, and/or response time, depending on the system requirements

1.d Major Operating System Components

CPU scheduling

- Quick preview: A sample of CPU scheduling algorithms

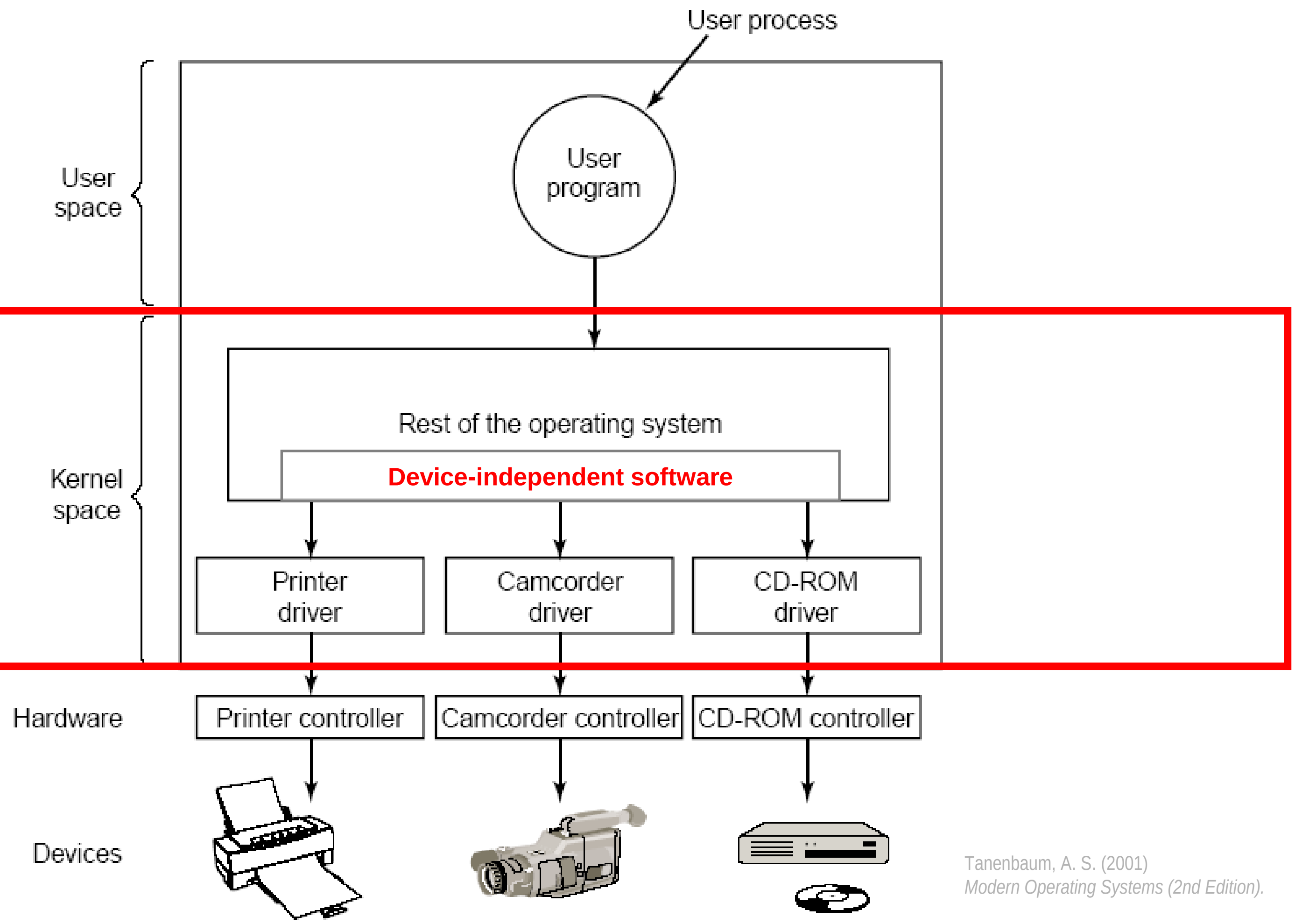


Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

A comparison of scheduling policies

1.d Major Operating System Components

Input/output



Tanenbaum, A. S. (2001)
Modern Operating Systems (2nd Edition).

Layers of the I/O subsystem

1.d Major Operating System Components
Input/output

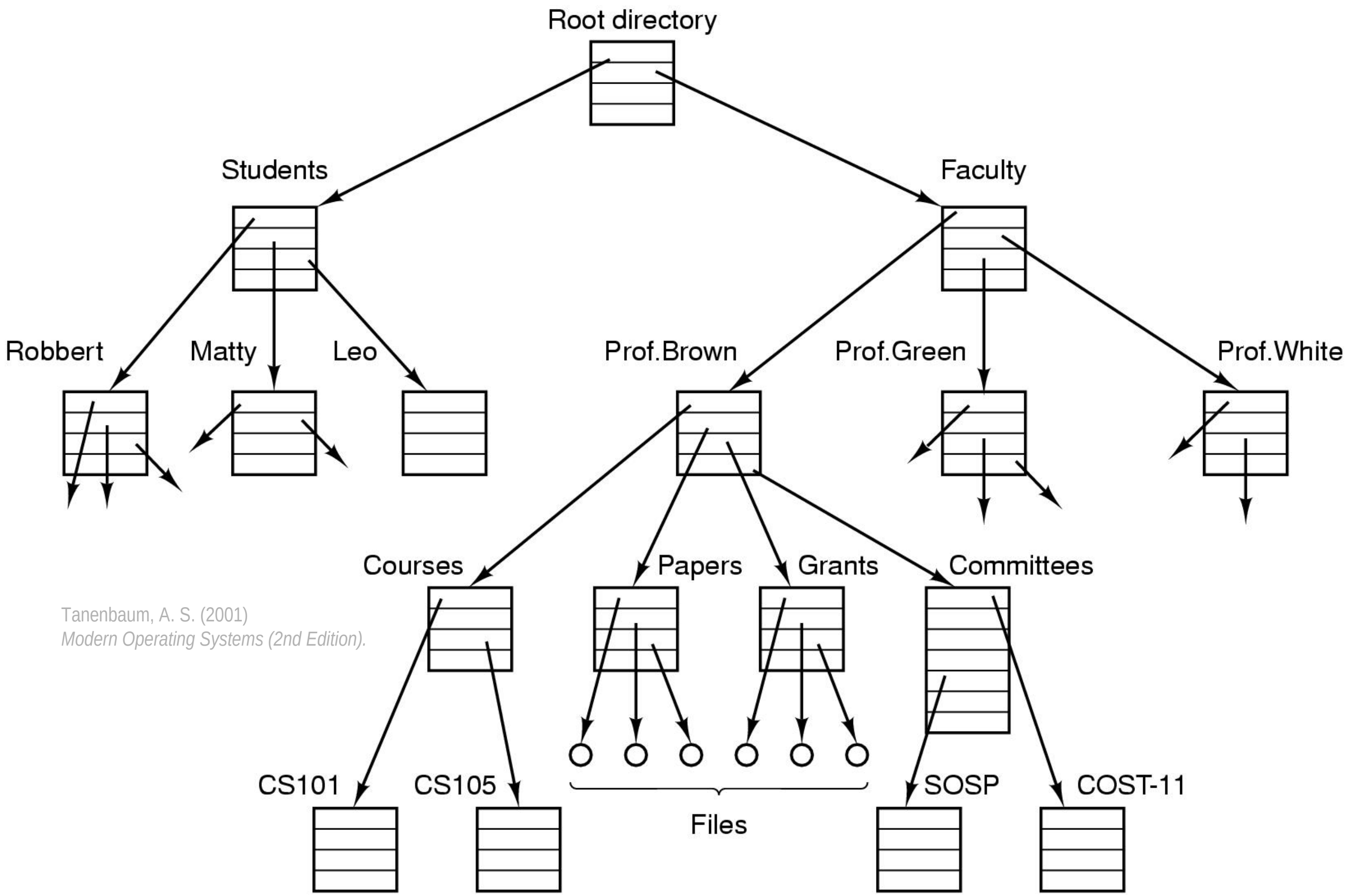
- Chart of Operating System Responsibilities

§D – The O/S is responsible for controlling access to all the I/O devices

- ✓ the O/S hides the peculiarities of specific hardware devices from the user
- ✓ the O/S issues the low-level commands to the devices, catches interrupts and handles errors
- ✓ the O/S relies on software modules called “device drivers”
- ✓ the O/S provides a device-independent API to the user programs, which includes buffering

1.d Major Operating System Components

File system



Tanenbaum, A. S. (2001)
Modern Operating Systems (2nd Edition).

A file system for a university department

1.d Major Operating System Components

File system

- Chart of Operating System Responsibilities

§E – The O/S is responsible for providing a uniform logical view of information storage

- ✓ the O/S defines a logical unit of storage, the “file”, and groups files in a hierarchy of “directories”
- ✓ the O/S supports primitives for manipulating files and directories (create, delete, rename, read, write, etc.)
- ✓ the O/S ensures data confidentiality and integrity
- ✓ the O/S keeps a mapping of the logical files onto the physical secondary storage
- ✓ the O/S backs up files on stable (nonvolatile) storage media

Principles of Operating Systems

CS 446/646

1. Introduction to Operating Systems

a. Role of an O/S

b. O/S History and Features

c. Types of O/S

d. **Major O/S Components**
   - ✓ Processes
   - ✓ Memory management
   - ✓ CPU scheduling
   - ✓ Input/output
   - ✓ File system

e. **System Calls**

f. **O/S Software Architecture**

g. **Examples of O/S**