



# Operating System Concepts

Che-Wei Chang

[chewei@mail.cgu.edu.tw](mailto:chewei@mail.cgu.edu.tw)

Department of Computer Science and Information Engineering, Chang Gung University

# Syllabus

- ▶ **Lecturer:** Che-Wei Chang (張哲維)
- ▶ **Lecture Hours:** Wednesday 9:10 am – 12:00 pm
- ▶ **Office Hours:** Thursday 4:00 pm – 5:30 pm
- ▶ **Classroom:** C0102R
- ▶ **Textbook:** Silberschatz, Galvin, and Gagne, “Operating System Principles,” 9th Edition, John Wiley & Sons
- ▶ **Website:** <https://icechewei.github.io/webpage/teaching.html>
- ▶ **Grading:**
  - Quiz and Attendance: 20%
  - Homework and Project: 30%
  - Midterm: 25%
  - Final: 25%



# Teaching Assistants

- ▶ 何承育 <alex1357823@yahoo.com.tw>
- ▶ 黃聖文 <david960023@gmail.com>



# Rules

- ▶ Closed Book Examinations
  - All books and papers should be collected into your backpacks
- ▶ Some Quizzes
  - The announcement is not always provided
- ▶ No Grade Adjustment
  - Some bonus might be provided in the project and exams

# Contents



1. Introduction
2. System Structures
3. Process Concept
4. Multithreaded Programming
5. Process Scheduling
6. Synchronization
7. Deadlocks
8. Memory-Management Strategies
9. Virtual-Memory Management
10. File System
11. Implementing File Systems
12. Secondary-Storage Systems

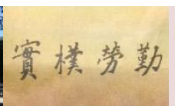




# Chapter 1 . Introduction

# Objectives

- ▶ To describe the basic organization of computer systems
- ▶ To provide a grand tour of the major components of operating systems
- ▶ To give an overview of the many types of computing environments
- ▶ To explore several open-source operating systems





# Overview

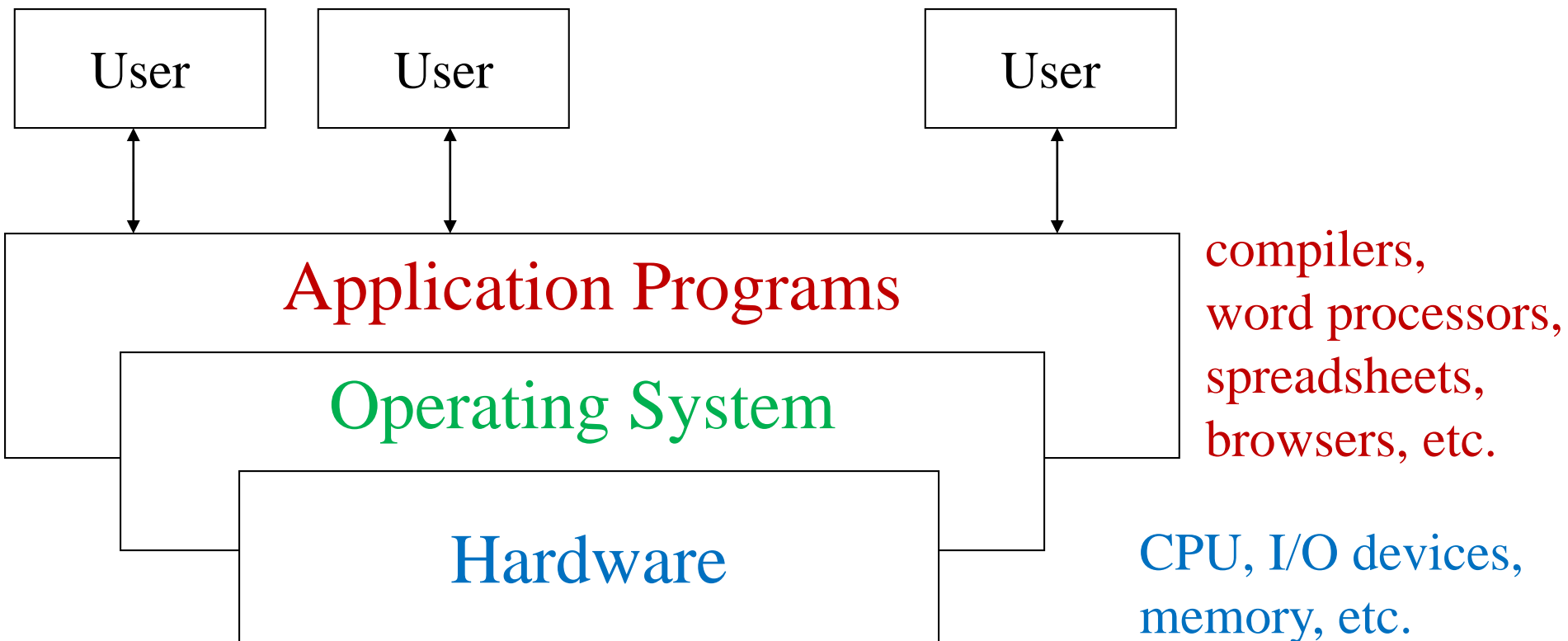


# Introduction

- ▶ What is an Operating System?
  - A basis for application programs
  - An intermediary between users and hardware
- ▶ Amazing Variety
  - Super computers, enterprise servers, personal computers (PCs), handheld computers, embedded systems, wearable devices
  - Convenient vs Efficient



# Computer System Components



- ▶ OS – a government/environment provider



# User View

- ▶ The user view of the computer varies by the interface being used
- ▶ Examples:
  - Personal computer → Ease of use
  - Mainframe or minicomputer → maximization of resource utilization
    - Efficiency and fair share
  - Handheld computer → individual usability
  - Embedded computer without user view → run without user intervention



# System View

- ▶ A Resource Allocator
  - CPU time, Memory Space, File Storage, I/O Devices, Shared Code, Data Structures, and more
- ▶ A Control Program
  - Control execution of user programs
  - Prevent errors and misuse
- ▶ OS Definition – US Department of Justice against Microsoft in 1998 (Netscape Navigator)
  - The stuff shipped by vendors as an OS
    - Internet Explorer → No
    - Microsoft Windows → Yes
  - Run at all time



# System Goals

- ▶ Two Conflicting Goals:
  - Convenient for the user
  - Efficient operation of the computer system
- ▶ We should
  - recognize the influences of operating systems and computer architecture on each other
  - and learn why and how OS's are by tracing their evolution and predicting what they will become
    - Cray-2: a super computer in 1985 with 3.9 GFLOPS
      - GFLOPS: Giga FLoating-point Operations Per Second
    - Ryzen 9 3950X : an AMD desktop processor in 2019 (Q4) with 170.56 GFLOPS

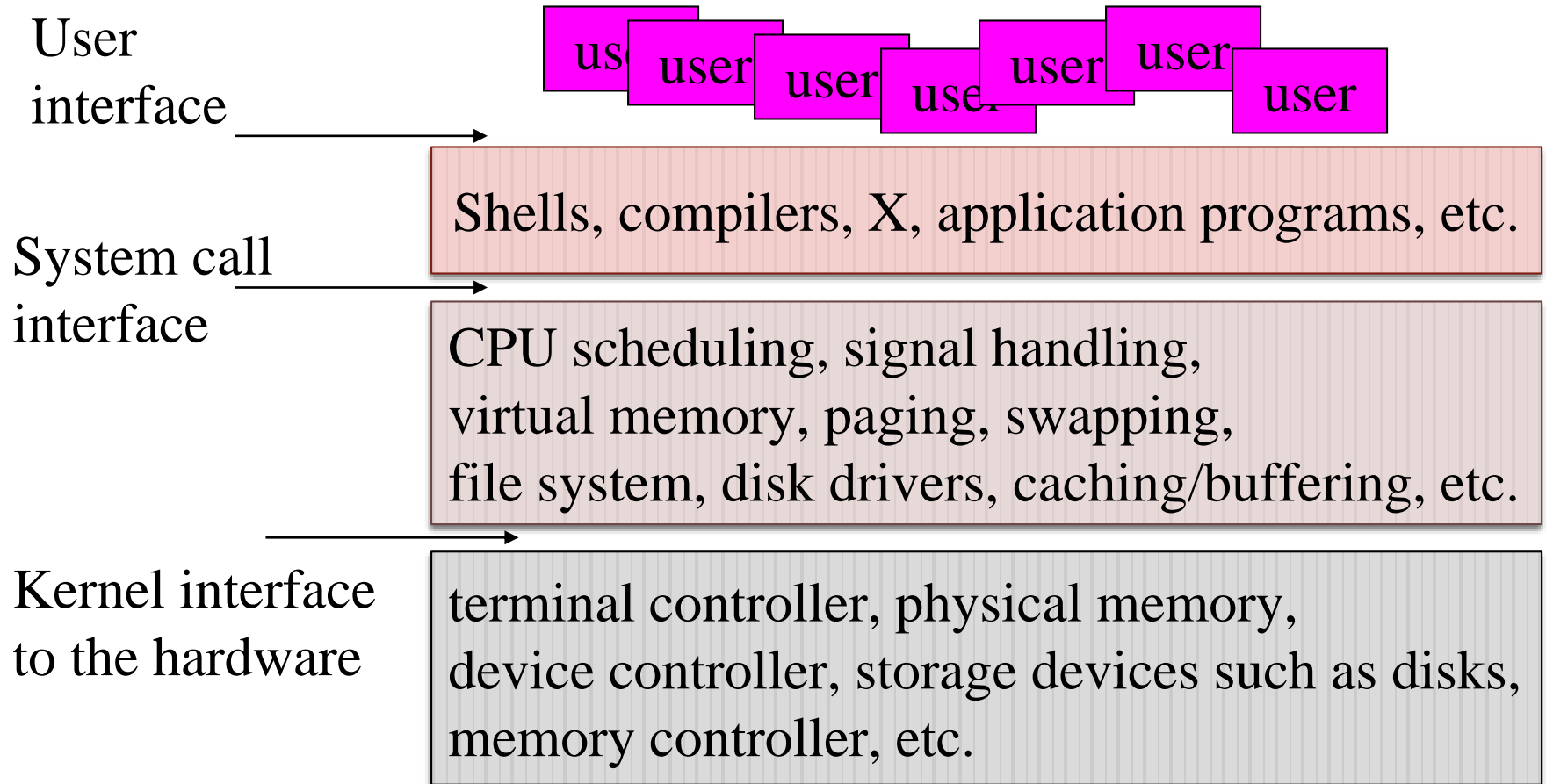
Source:

<https://en.wikipedia.org/wiki/Supercomputer>

[https://setiathome.berkeley.edu/cpu\\_list.php](https://setiathome.berkeley.edu/cpu_list.php)



# UNIX Architecture



UNIX



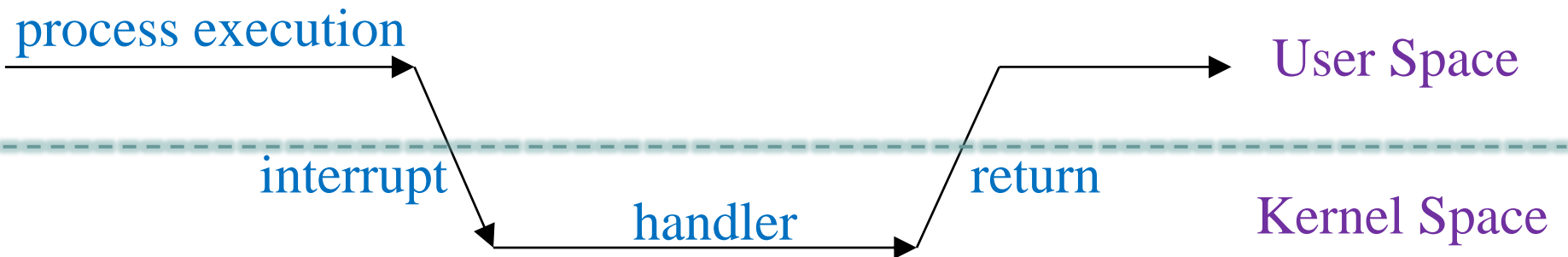
# Computer Startup

- ▶ **Bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as firmware
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution
  - BIOS: basic input output system
  - UEFI: unified extensible firmware interface
- ▶ **Operating system** runs initial program to initialize system processes, e.g., various daemons, login processes, after the kernel has been bootstrapped



# Interrupt

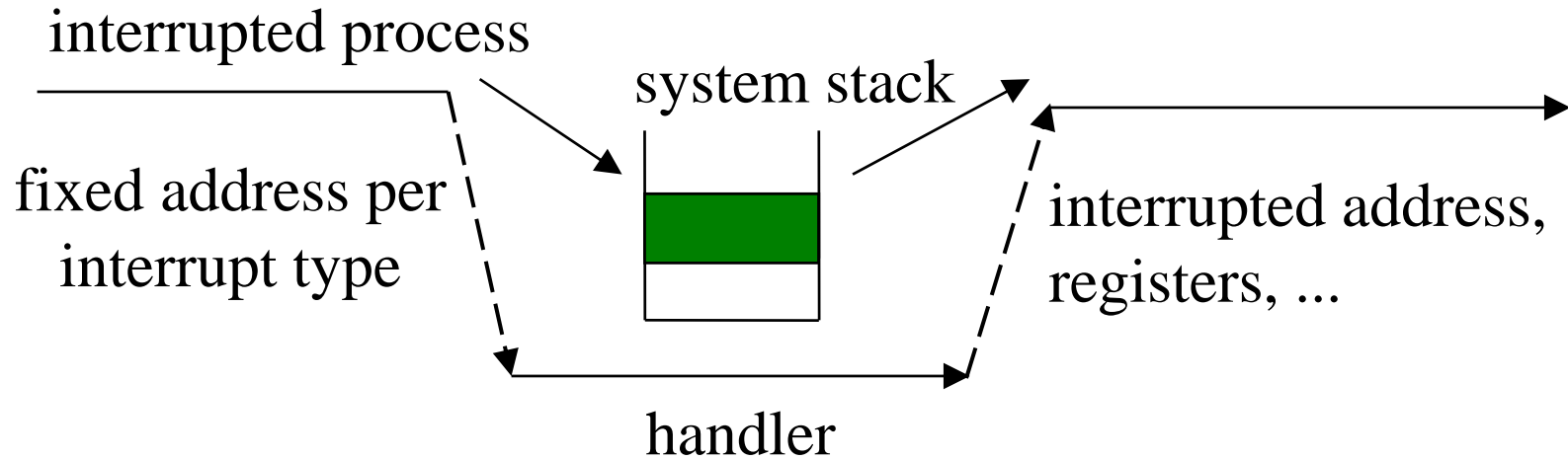
- ▶ **Hardware interrupts**: services requests of I/O devices
- ▶ **Software interrupts**: signals, invalid memory access, division by zero, **system calls**, etc



- ▶ Procedures: generic handler or interrupt vector (MS-DOS, UNIX)



# Interrupt Handling Procedure (1 / 2)



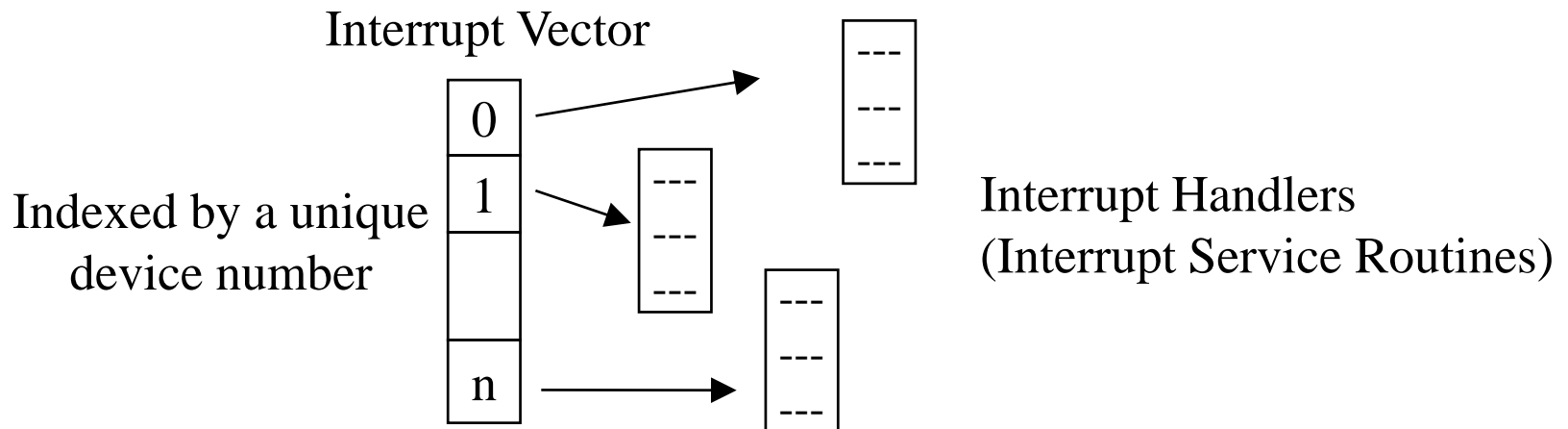
- ▶ Saving of the address of the interrupted instruction
  - By fixed locations or stacks
- ▶ Interrupt disabling or enabling issues
  - Might lose some interrupts?
    - ➔ prioritized interrupts masking



# Interrupt Handling Procedure (2/2)

## ► Interrupt Handling

- Save interrupt information
- OS determine the interrupt type
- Call the corresponding handlers
- Return to the interrupted job by the restoring important information (e.g., saved return address and program counter)

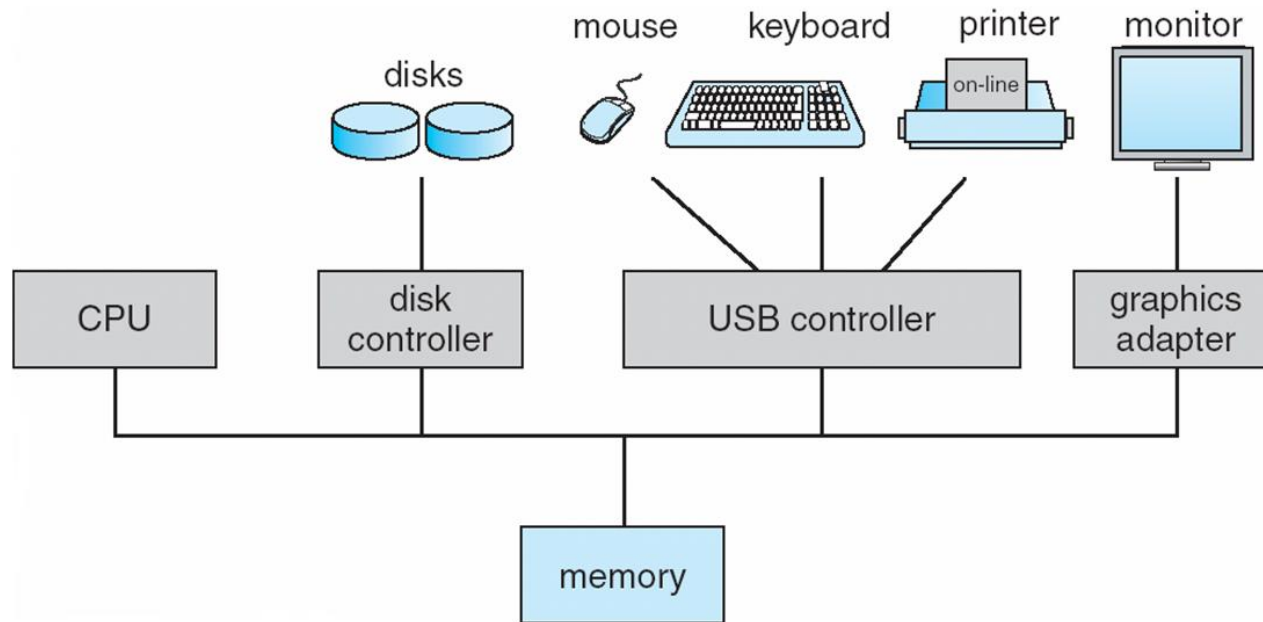




# Computer Systems

# Computer System Organization

- ▶ One or more CPUs for computing
- ▶ Memory and storage devices for keeping data
- ▶ Peripheral devices for I/O operations



# Computer Processors

- ▶ Some systems use a single general-purpose processor
  - Most systems have special-purpose processors as well
- ▶ Multiprocessors systems are growing in use and importance
  - Tightly coupled: have more than one processor in close communication sharing computer bus, clock, and sometimes memory and peripheral devices
  - Loosely coupled: otherwise



# Multiprocessor Systems

- ▶ Symmetric Multiprocessing
  - Each processor runs an identical copy of the OS
  - All processors are the same in user and system views
- ▶ Asymmetric Multiprocessing
  - Master-and-slave framework
  - Commonly seen in extremely large systems
  - Hardware and software make a difference



# Parallel Systems

- ▶ Tightly-coupled multiprocessor systems are also known as parallel systems
- ▶ Advantages:
  - Increased throughput
  - Economy of scale
  - Increased reliability– graceful degradation and fault tolerance
- ▶ Trends
  - Multiple cores over single chip
    - Cores in a chip can even share cache
  - Hyper-threading processors
    - More than one programs can be executed on a core



# Clustered Systems

- ▶ Loosely-coupled multiprocessor systems are also known as clustered systems
  - Computers which share storage and are closely linked via LAN networking
  - Processors do not share memory or a clock
- ▶ Advantages:
  - High availability
  - Performance improvement
- ▶ Some clusters are for high-performance computing
  - Applications must be written to use parallelization
- ▶ Some clusters have distributed lock manager
  - Conflicting operations must be avoided





# Memory Management

- ▶ Memory: a large array of words or bytes, where each has its own address
- ▶ OS must keep several programs in memory to improve CPU utilization and user response time
- ▶ Management algorithms depend on the hardware support
- ▶ Services
  - Memory usage and availability
  - Decision of memory assignment
  - Memory allocation and deallocation



# Secondary Storage Management

- ▶ Goal:
  - On-line storage medium for programs & data
    - Backup of main memory
- ▶ Services for Disk Management
  - Free-space management
  - Storage allocation, e.g., continuous allocation
  - Read/write request scheduling, e.g., first-come-first-serve



# Tertiary Storage Management

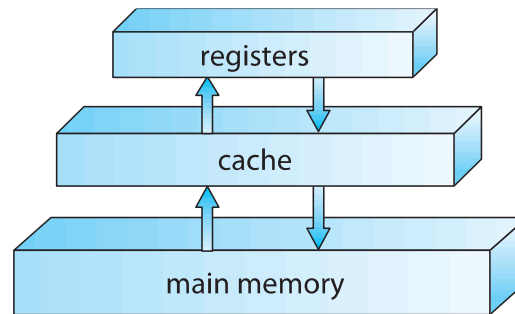
- ▶ Goals:
  - Backups of disk data, seldom-used data, and long-term archival storage
- ▶ Examples:
  - Magnetic tape drives and their tapes, CD & DVD drives and platters
- ▶ Services – OS Supports or Applications' Duty
  - Device mounting and unmounting
  - Exclusive allocation and freeing
  - Data transfers from tertiary devices to secondary storage devices



# Storage-Device Hierarchy

## Primary Storage

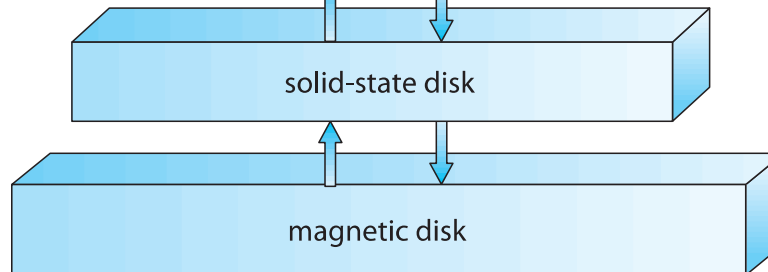
- volatile storage



- Access time: a cycle
- Access time: several cycles
- Access time: many cycles

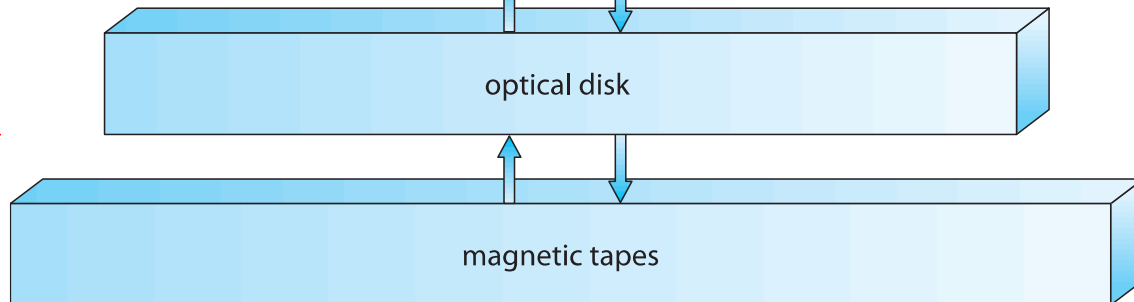
## Secondary Storage

- nonvolatile storage



## Tertiary Storage

- removable media



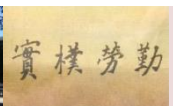
# I/O Devices

- ▶ Device drivers are used by OS to drive the device controllers
- ▶ Character Devices
  - Sequential access
  - Examples might include printers, scanners, sound boards
  - The same device may have both block and character oriented interfaces
- ▶ Block Devices
  - Block size is from 512B to 4KB
  - For example, disks are commonly implemented as block devices

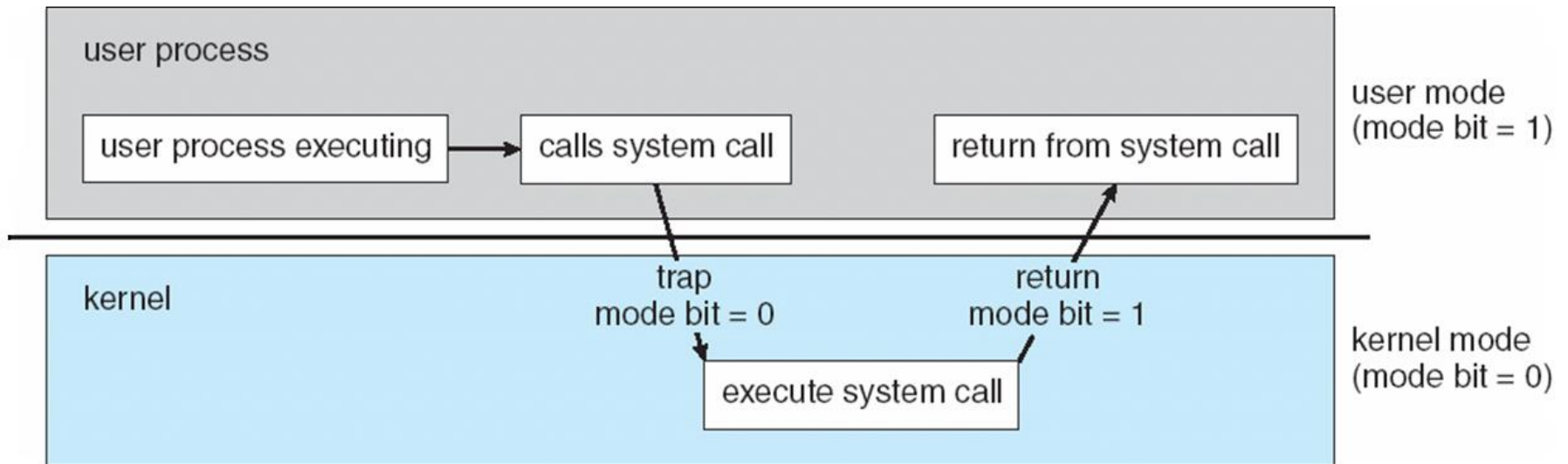


# Hardware Protection

- ▶ Dual-mode operation allows OS to protect itself and other system components
  - User mode and kernel mode
  - Mode bit provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as privileged, only executable in kernel mode
    - System call changes the current mode to kernel mode, return from call resets it to user mode
- ▶ Increasingly CPUs support multi-mode operations
  - For example, virtual machine manager (VMM) mode



# Transition Between User and Kernel Modes





# Operating-System Operations



# Process Management

- ▶ **A process is a program in execution**
  - A program is a passive entity
  - A process is an active entity
- ▶ Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- ▶ Process termination requires to reclaim any reusable resources
- ▶ Typically system has many processes, some users, some operating system running concurrently on one or more CPUs



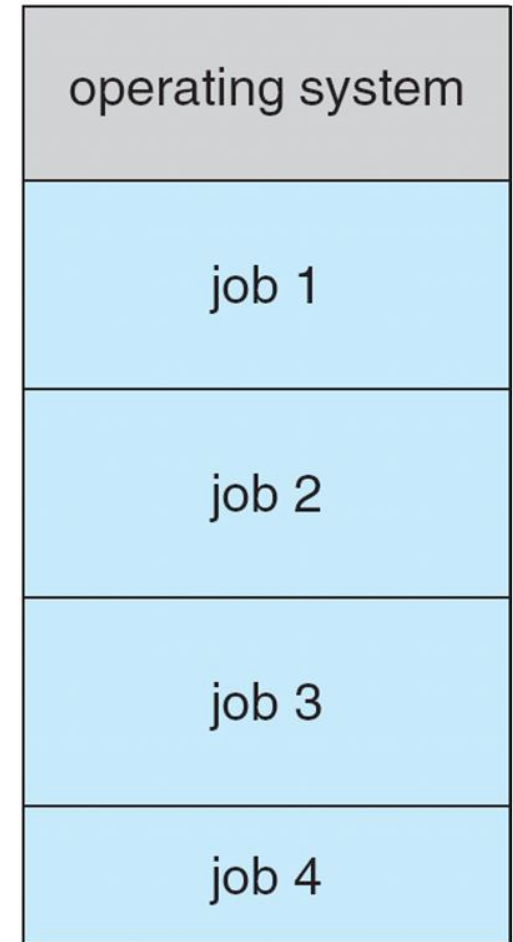
# Process Management Activities

- ▶ Creating and deleting both user and system processes
- ▶ Suspending and resuming processes
- ▶ Providing mechanisms for process synchronization
- ▶ Providing mechanisms for process communication
- ▶ Providing mechanisms for deadlock handling



# Multiprogramming

- ▶ Multiprogramming is needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - A subset of total jobs in system is kept in memory via **job scheduling**
  - One job selected and run via **CPU scheduling**
  - When a job has to wait (for I/O for example), OS switches to another job



# Timesharing

- ▶ Timesharing (multitasking) is a logical extension of multiprogramming
  - CPU switches jobs so frequently
  - Users can interact with each job while it is running
  - Timesharing creates interactive computing
- ▶ Each user has at least one program executing in memory
  - A program executing in memory → a process is created
- ▶ If several processes ready to run at the same time
  - Pick a process to run on CPU → CPU scheduling
- ▶ If processes don't fit in memory,
  - Swapper moves them in and out of memory → job scheduling
- ▶ Virtual memory allows execution of processes not completely in memory



# Caching (1 / 2)

## ▶ Caching

- Information is copied to a faster storage system on a temporary basis
- Assumption: Data or binaries will be used again soon
  - Programmable registers, instruction cache, etc.

## ▶ Cache Management

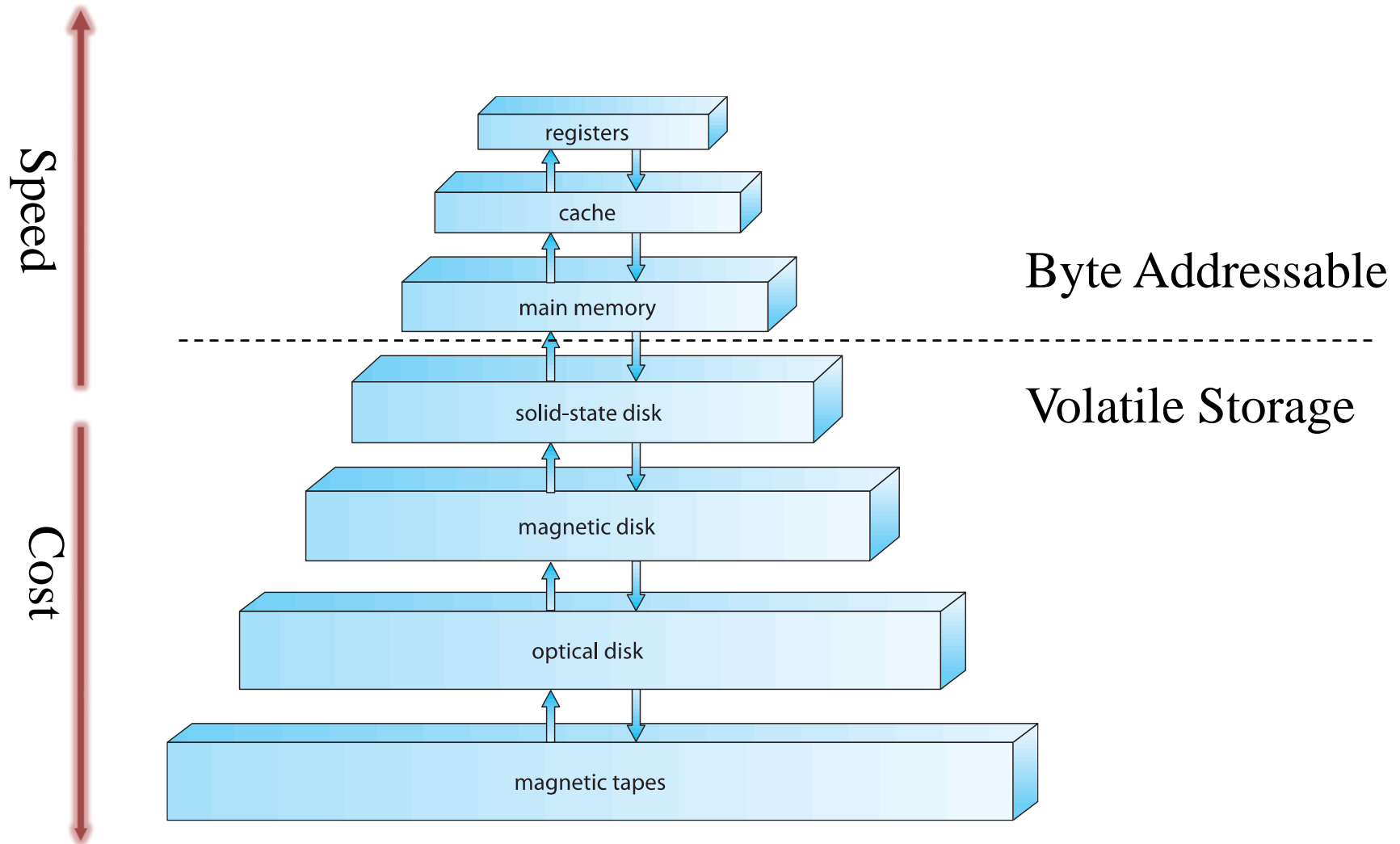
- Cache size and the replacement policy

## ▶ Movement of Information Between Hierarchy

- Hardware design & operating system control

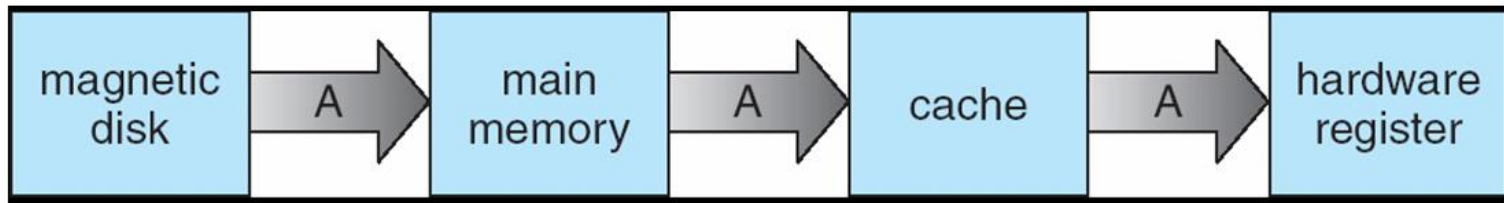


# Caching (2/2)



# Migration of Integer A from Disk to Register

- ▶ Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy



- ▶ Multiprocessor environment must provide **cache coherency** in hardware such that all CPUs have the most recent value in their cache
- ▶ Distributed environment situation even more complex
  - Several copies of a datum can exist

# File-System Management

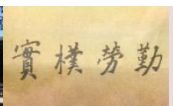
- ▶ Goal:
  - A uniform logical view of information storage
  - Each medium controlled by a device
    - Magnetic tapes, magnetic disks, optical disks, etc.
- ▶ OS provides a logical storage unit: File
  - Formats:
    - Free form or being formatted rigidly
  - General Views:
    - A sequence of bits, bytes, lines, records





# File Management Activities

- ▶ Creating and deleting files and directories
- ▶ Primitives to manipulate files and directories
- ▶ Mapping files onto secondary storage
- ▶ Backup files onto stable (non-volatile) storage media



# I/O System Management

- ▶ Goal:
  - Hide the peculiarities of specific hardware devices from users
- ▶ Components of an I/O System
  - A buffering, caching, and spooling system
  - A general device-driver interface
  - Device drivers



# Protection and Security (1 / 2)

## ▶ Goal

- Resources are only allowed to be accessed by authorized processes

## ▶ Definitions:

- Protection – any mechanism for controlling the access of processes or users to the resources defined by the computer system
- Security – Defense of a system from external and internal attacks, e.g., viruses, denial of services, etc



# Protection and Security (2 / 2)

- ▶ Protected Resources
  - Files, CPU, memory space, etc
- ▶ Protection Services
  - Detection & controlling mechanisms
  - Specification mechanisms
- ▶ Distinguishing of Users
  - User names and ID's
  - Group names and GID's
  - Privilege Escalating, e.g., Setuid in Unix
    - To gain extra permissions for an activity





# Advanced Topics

# Distributed Systems (1 / 2)

- ▶ Definition: Loosely-Coupled Systems – processors do not share memory or a clock
- ▶ Advantages or Reasons
  - Resource sharing: computation power, peripheral devices, specialized hardware
  - Computation speedup: distribute the computation among various sites – load sharing
  - Reliability: redundancy → reliability



# Distributed Systems (2 / 2)

- ▶ Distributed systems depend on networking for their functionality
- ▶ Networks vary by the protocols used
  - TCP/IP, ATM, etc.
- ▶ Types – different distance
  - Local-area network (LAN)
  - Wide-area network (WAN)
  - Metropolitan-area network (MAN)
  - Small-area network – distance of few feet
- ▶ Media – copper wires, fiber strands, ...



# Real-Time Embedded Systems (1 / 2)

- ▶ Embedded Computers— Most Prevalent Form of Computers
  - Have a wide variety ranged from car engines to VCR's
  - Tend to have specific tasks and almost always run real-time operating systems
- ▶ Definition:
  - A real-time system is a computer system where a timely response by the computer to external stimulation is vital!





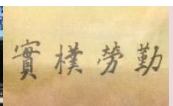
# Real-Time Embedded Systems (2/2)

- ▶ Hard real-time system: The system has failed if a timing constraint, e.g. deadline, is not met
  - All delays in the system must be bounded
  - Many advanced features are absent
- ▶ Soft real-time system: Missing a timing constraint is serious but does not necessarily result in a failure
  - A critical task has a higher priority
  - Supported in most commercial OS
- ▶ Real-time means **on-time** instead of **fast**



# Computing Environments— Mobile Devices

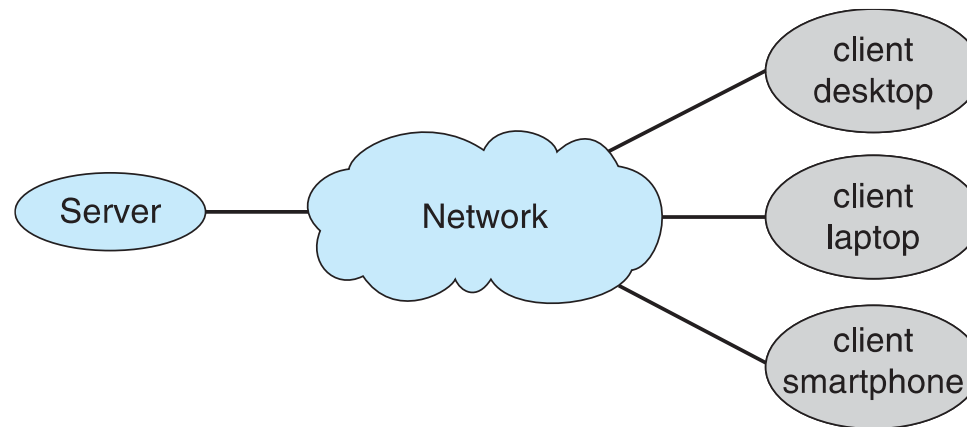
- ▶ Target devices: handheld smartphones, tablets, ...
- ▶ Extra features: GPS, gyroscope, ...
- ▶ New types of Application: augmented reality, ...
- ▶ Use IEEE 802.11 wireless, or cellular data networks for connectivity
- ▶ Leaders are Apple iOS and Google Android



# Computing Environments— Client–Server

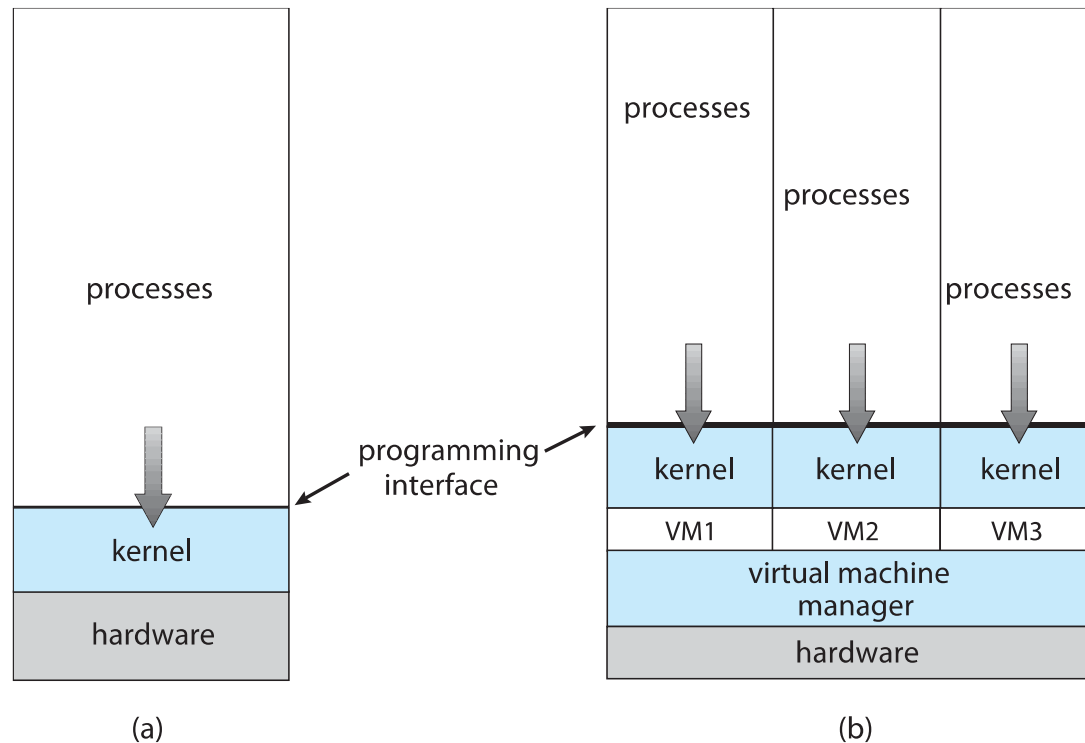
## ► Client-Server Systems

- Trend:
  - The functionality of clients is improved in the past decades
- Categories:
  - Compute-server systems
  - File-server systems



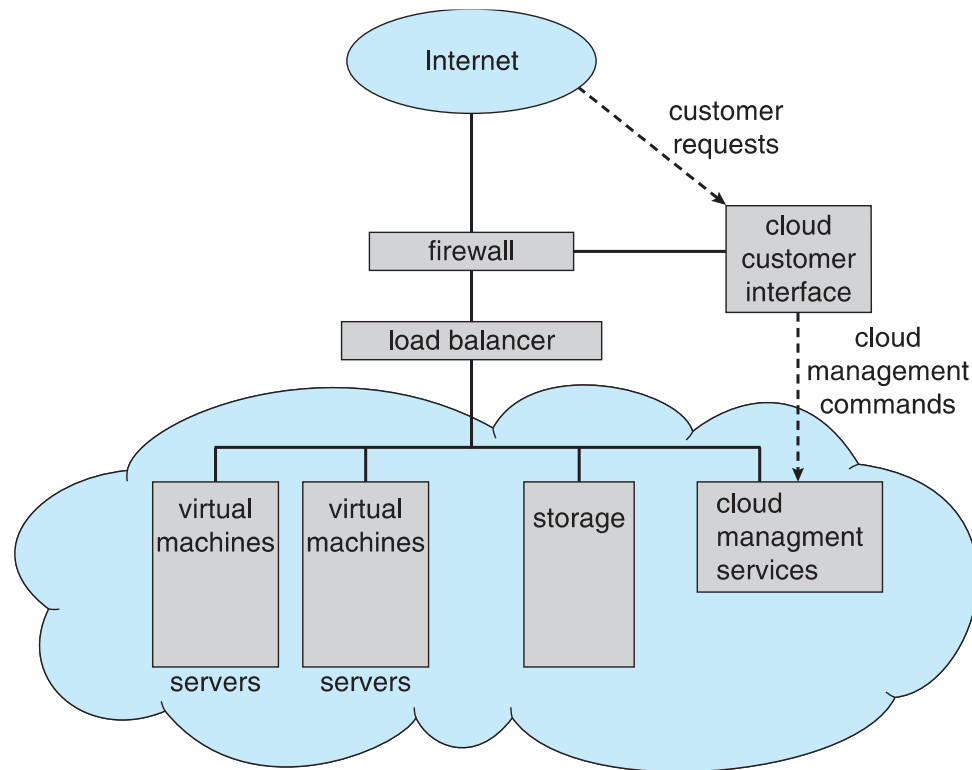
# Computing Environments— Virtualization

- Use cases involve laptops and desktops running multiple OSES for exploration or compatibility



# Computing Environments— Cloud Computing

- ▶ Cloud computing environments are composed of traditional OSeS, plus virtualization tools plus cloud management tools





**We will go through the details in  
the following lectures!**