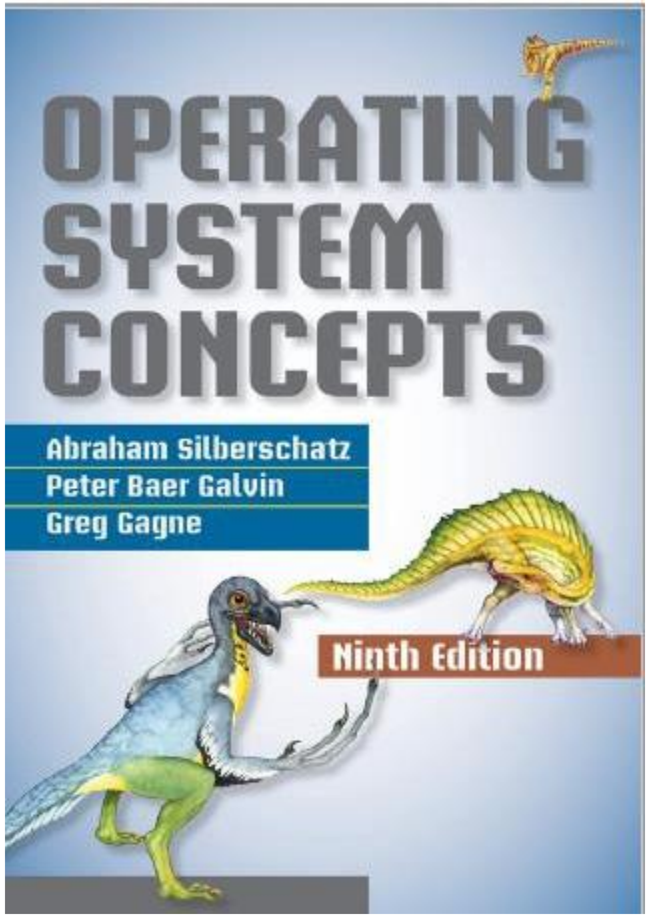
Operating System

LEC1: INTRODUCTION

Reference


OPERATING SYSTEM CONCEPTS
Abraham Silberschatz
Peter Baer Galvin
Greg Gagne
Ninth Edition

Topics to be covered

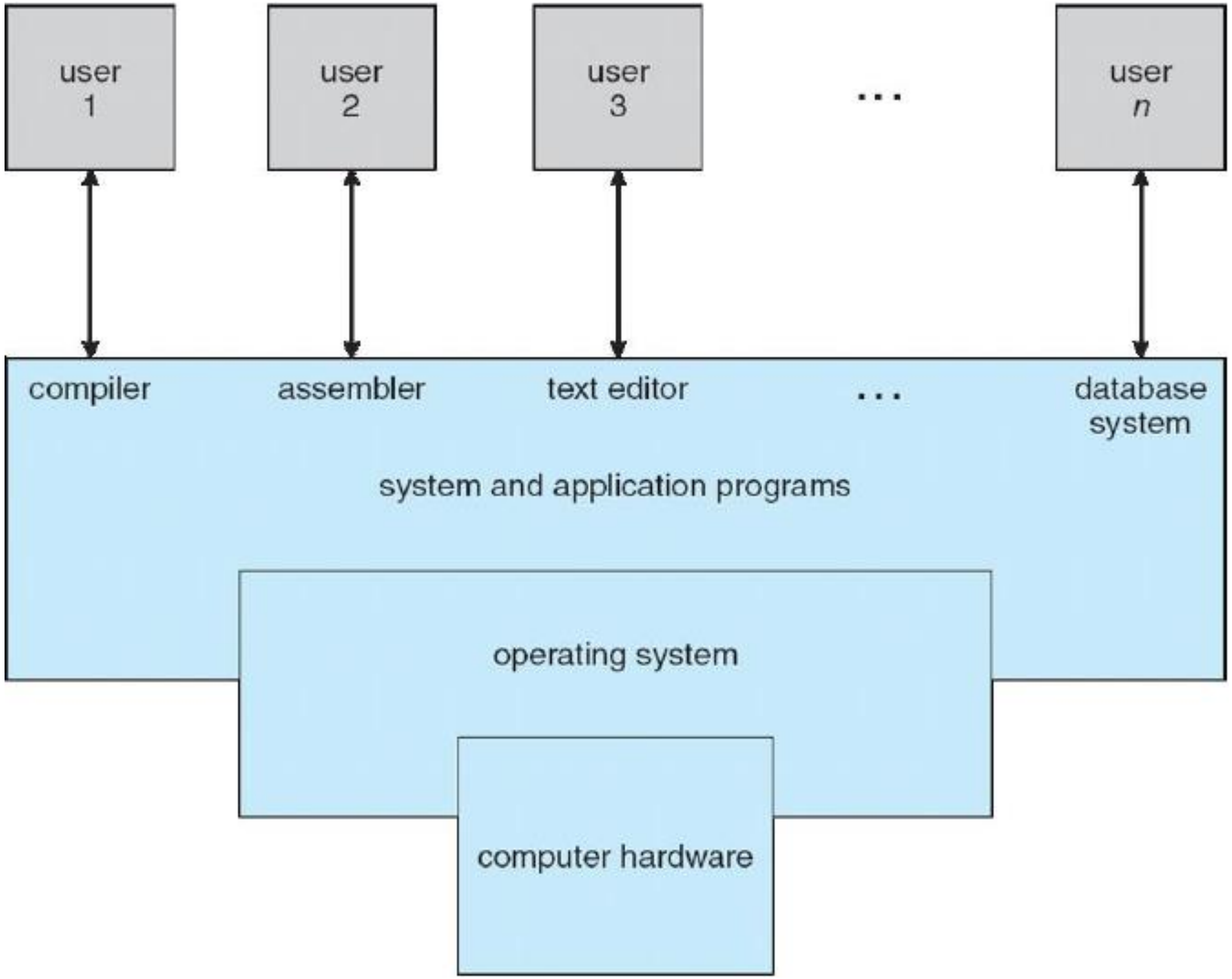
- Introduction to operating system
- Process Management
- CPU scheduling
- Deadlocks
- Memory Management
- I/O Systems
- Protection and security

What is Operating System?

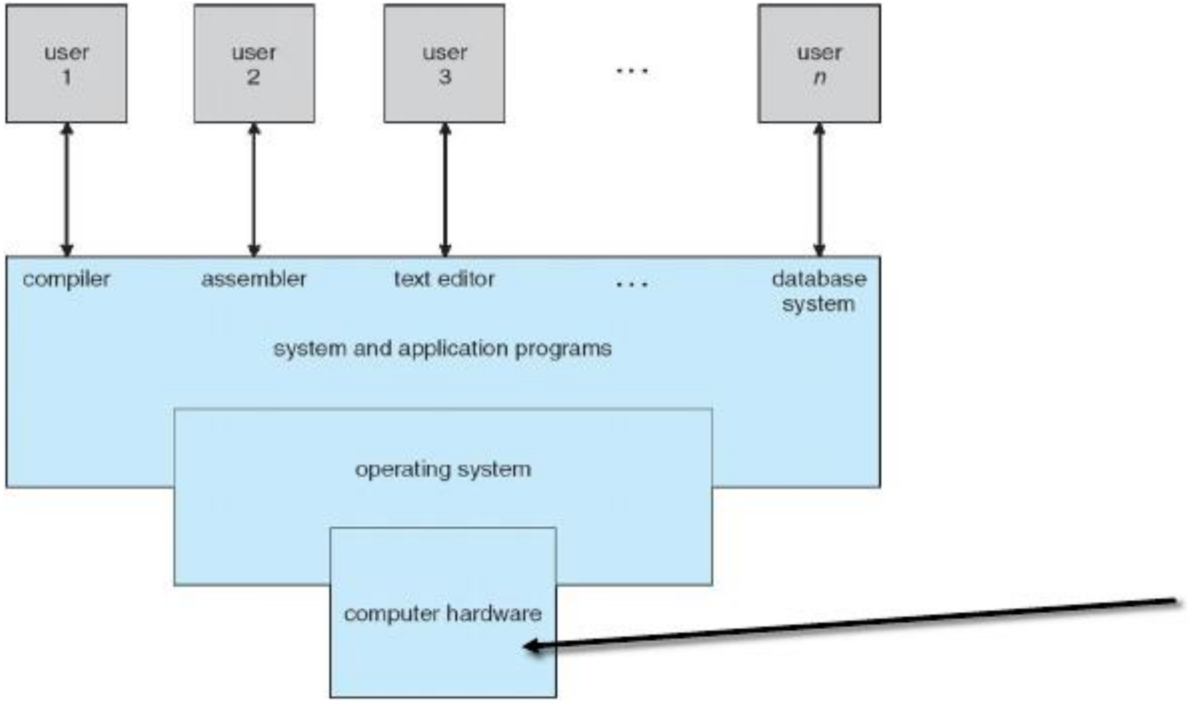
What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware
- **Operating system goals:**
  - **Execute** user programs and make solving user problems easier
  - Make the computer system **convenient** to use
  - Use the computer hardware in an **efficient** manner
- "The one program running at all times on the computer" is the **kernel**.
- Everything else is either
  - a system program (ships with the operating system) , or
  - an application program.

ComputerSystem Structure



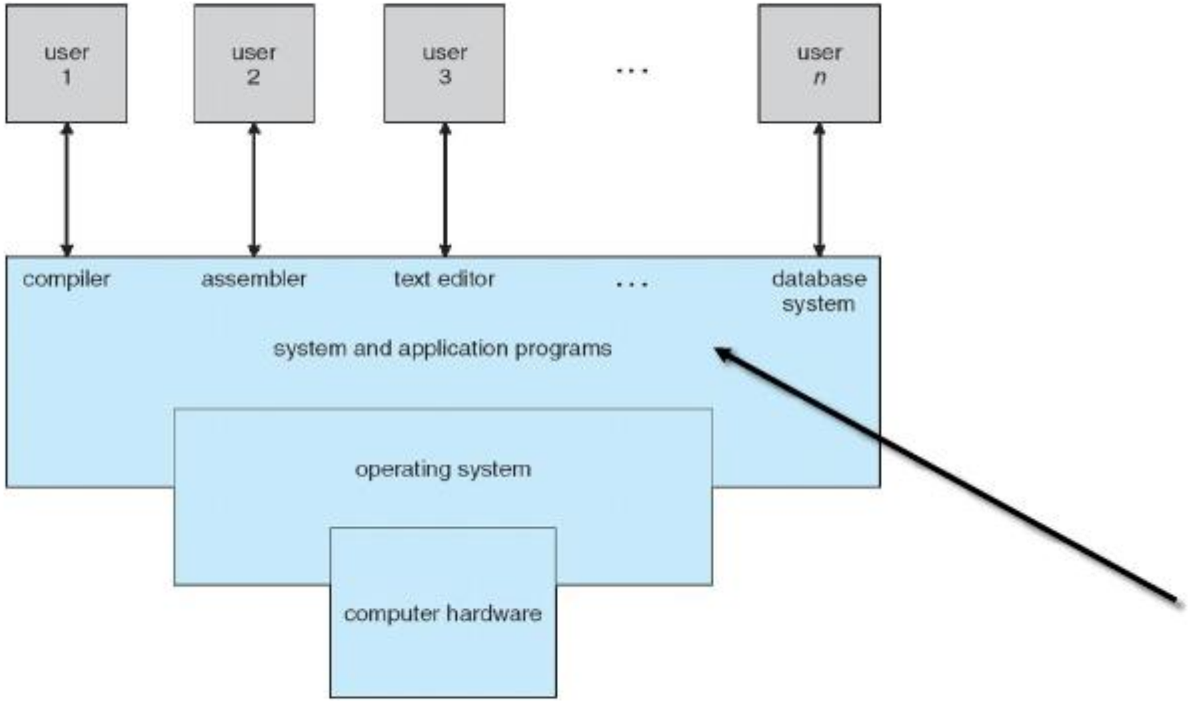
Computer System Structure



provides basic computing resources

CPU, memory, I/O devices

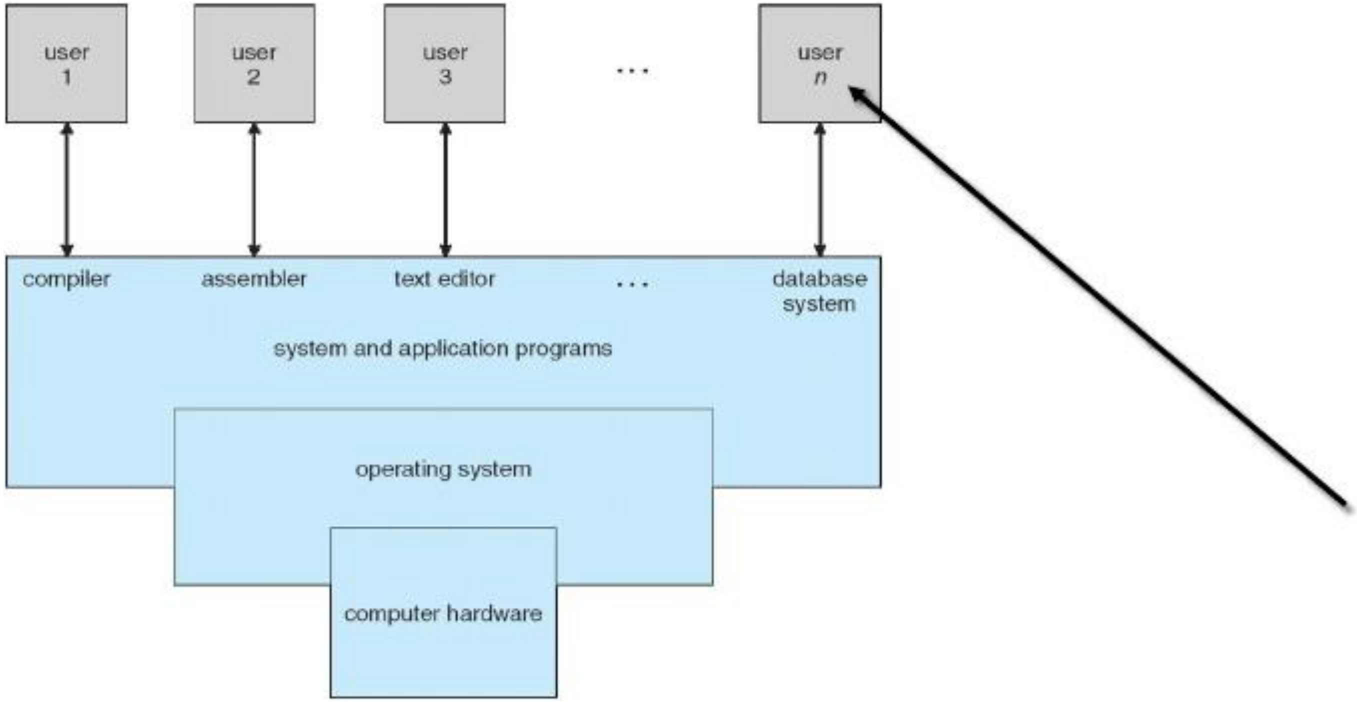
Computer System Structure



define the ways in which the system resources are used to solve the computing problems of the users

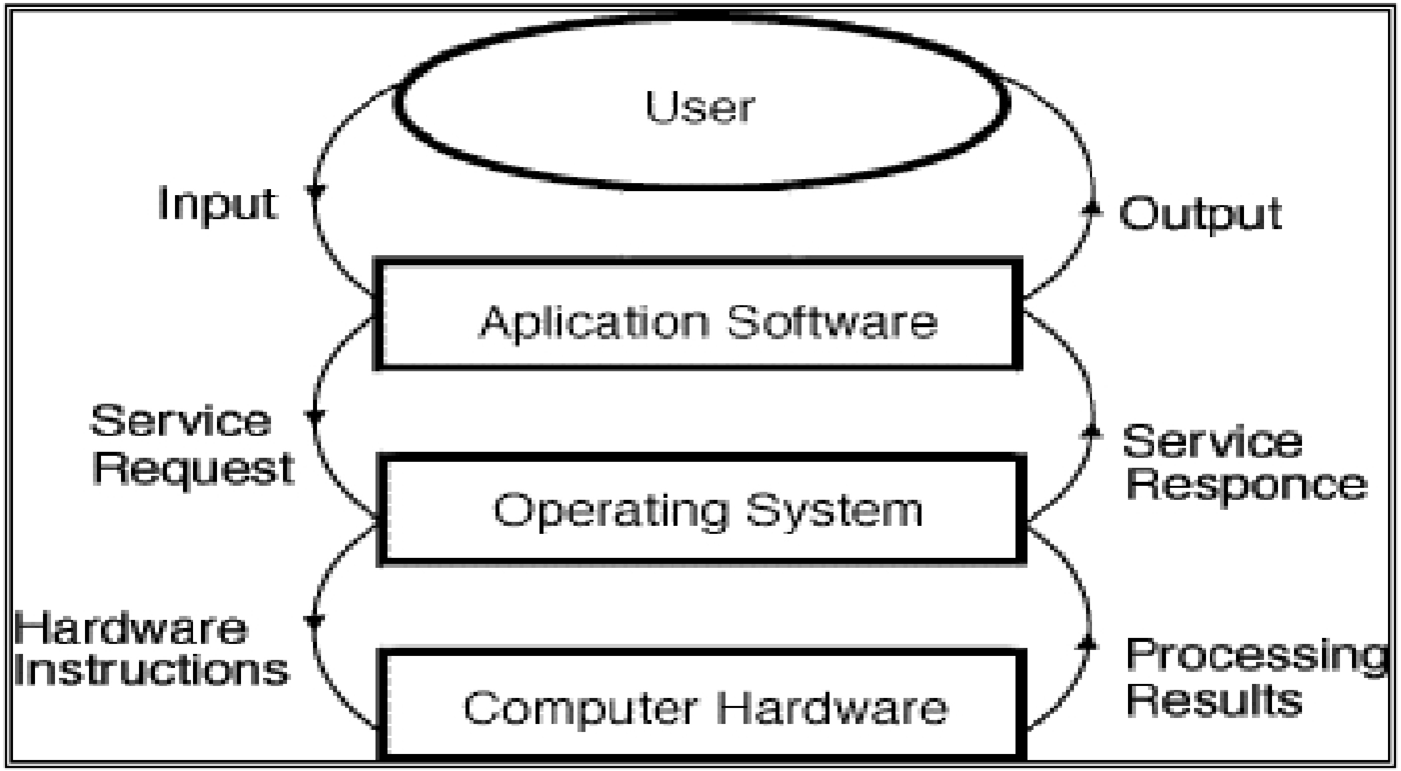
Word processors, compilers, web browsers, database systems, video games

Computer System Structure



People, machines, other computers

Dynamic View of System Components



ServicesProvidedbyOperatingSystems

1. **Program execution**.
2. **I/O operations**.
3. **File-system manipulation**
4. **Communications**.
5. **Error detection**.
6. **Resource allocation**.
7. **Accounting**.
8. **Protection and security**.

ServicesProvidedbyOperatingSystems

1. **Program execution**.
   - The system must be able to load a program into memory and to run that program.
   - The program must be able to end its execution, either normally or abnormally (indicating error).
2. **I/O operations**.
3. **File-system manipulation**
4. **Communications**.
5. **Error detection**.
6. **Resource allocation**.
7. **Accounting**.
8. **Protection and security**.

ServicesProvidedbyOperatingSystems

1. **Program execution**.
2. **I/O operations**.
   - A running program may require I/O, which may involve a file or an I/O device.
   - For efficiency and protection, users usually cannot control I/O devices directly.
   - Therefore, the operating system must provide a means to do I/O.
3. **File-system manipulation**
4. **Communications**.
5. **Error detection**.
6. **Resource allocation**.
7. **Accounting**.
8. **Protection and security**.

ServicesProvidedbyOperatingSystems

1. **Program execution**.
2. **I/O operations**
3. **File-system manipulation**.
   - programs need to read and write files and directories. They also need to create and delete them by name, search for a given file, and list file information.
   - some operating systems include permissions management to allow or deny access to files or directories based on file ownership.
4. **Communications**.
5. **Error detection**.
6. **Resource allocation**.
7. **Accounting**.
8. **Protection and security**.

ServicesProvidedbyOperatingSystems

1. **Program execution**.
2. **I/O operations**.
3. **File-system manipulation**.
4. **Communications**.
   - one process may need to exchange information with another process.
   - Such communication may occur between processes on the same computer or between processes on different computer systems tied together by a computer network.
   - Communications may be implemented via **shared memory**, in which two or more processes read and write to a shared section of memory, or **message passing**, in which packets of information in predefined formats are moved between processes by the operating system.
5. **Error detection**.
6. **Resource allocation**.
7. **Accounting**.
8. **Protection and security**.

ServicesProvidedbyOperatingSystems

1. **Program execution**.
2. **I/O operations**.
3. **File-system manipulation**.
4. **Communications**.
5. **Error detection**.
   - The operating system needs to be detecting and correcting errors constantly.
   - Errors may occur:
     - in the CPU and memory hardware (such as a memory error or a power failure),
     - in I/O devices (such as a parity error on disk, a connection failure on a network, or lack of paper in the printer),
     - in the user program (such as an arithmetic overflow, an attempt to access an illegal memory location, or a too-great use of CPU time).
   - For each type of error, the operating system should take the appropriate action to ensure correct and consistent computing.
6. **Resource allocation**.
7. **Accounting**.
8. **Protection and security**.

ServicesProvidedbyOperatingSystems

1. **Program execution**.
2. **I/O operations**.
3. **File-system manipulation**.
4. **Communications**.
5. **Error detection**.
6. **Resource allocation**.
   - When there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them.
   - The operating system manages many different types of resources.
7. **Accounting**.
8. **Protection and security**.

ServicesProvidedbyOperatingSystems

1. **Program execution**.
2. **I/O operations**.
3. **File-system manipulation**.
4. **Communications**.
5. **Error detection**.
6. **Resource allocation**.
7. **Accounting**.
   - We want to keep track of which users use how much and what kinds of computer resources.
8. **Protection and security**.

ServicesProvidedbyOperatingSystems

1. **Program execution**.
2. **I/O operations**.
3. **File-system manipulation**.
4. **Communications**.
5. **Error detection**.
6. **Resource allocation**.
7. **Accounting**.
8. **Protection and security**.

- Protection involves ensuring that all access to system resources is controlled.
- Security of the system from outsiders is also important.
  - each user to authenticatehimself or herself to the system to gain access to system resources. It extends to
  - defending external I/O devices, including network adapters, from invalid access attempts and to recording all such connections for detection of break-ins.

Historical overview

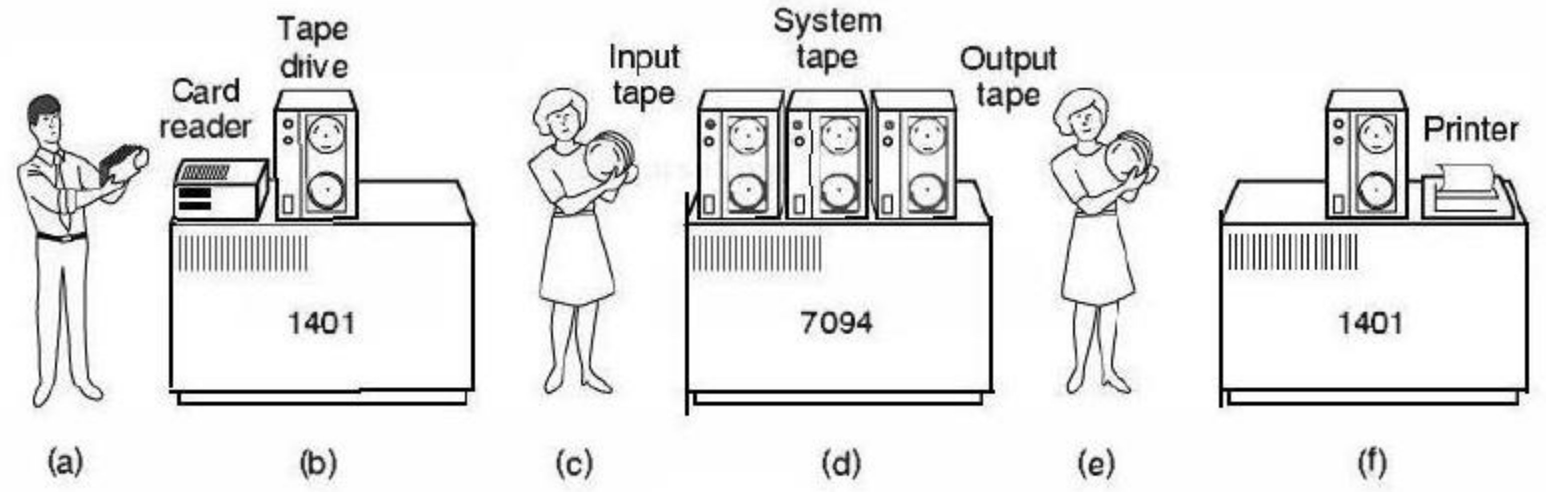
- Mainframe system: the first computer used to tackle many commercial and scientific applications
  - Batch systems
  - Multiprogrammed systems
  - Time-Sharing Systems
- Personal-Computer Systems
- Multiprocessor Systems (parallel system)
- Distributed System

Mainframe system

Historical overview

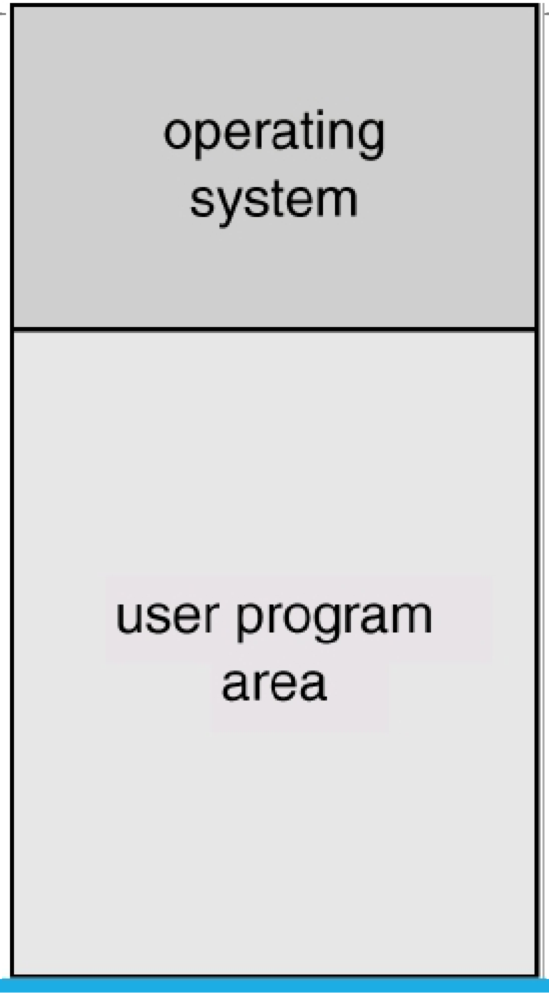
Mainframe system:

- Batch systems:
  - Physically enormous machines run from a console
  - The common input devices were card readers and tape drivers
  - The common output devices were line printers, tape drivers and card punches
  - The user didn't interact directly with the computer system
  - The user prepared a job in the form of punch cards (program, data control information) and submit it to the computer operator
  - The output consists of results of the program, dump of the final memory and the register content for debugging
  - The operating system was simple, it transfer control automatically from one job to another.
  - To speed up processing operators batched together jobs with similar needs and run them through computers as a group



An early batch system. (a) Programmers bring cards to 1401. (b) 1401 reads batch of jobs onto tape. (c) Operator carries input tape to 7094. (d) 7094 does computing. (e) Operator carries output tape to 1401. (f) 1401 prints output.

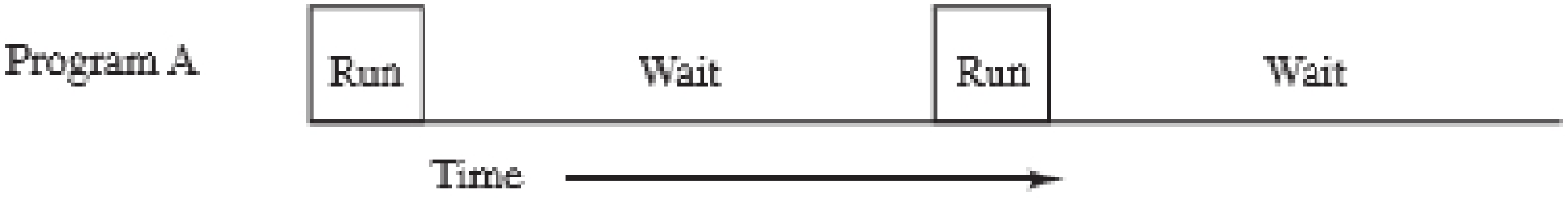
MemoryLayout for a Simple Batch System



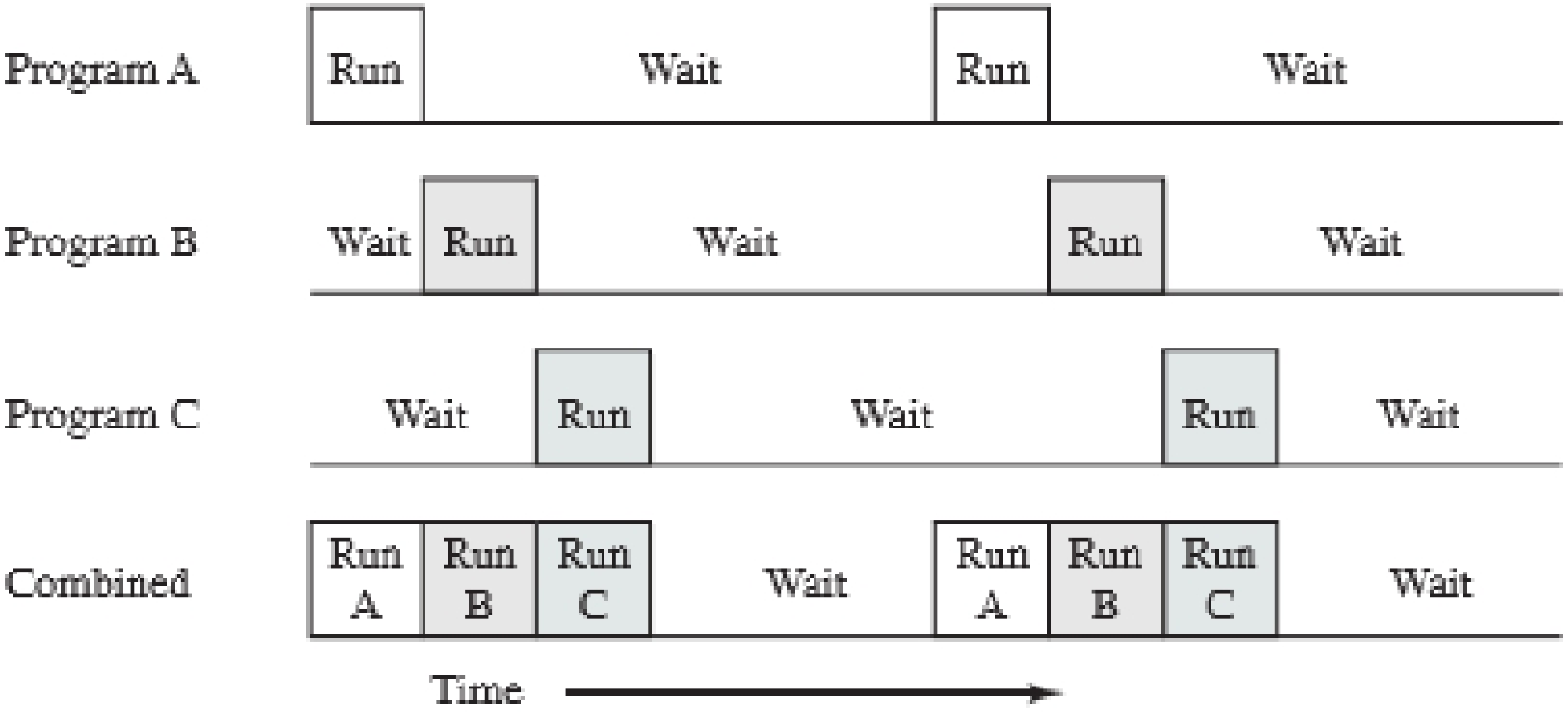
Historical overview

Mainframe system:

- Multiprogrammed systems:
  - Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them.
  - The operating system picks and begins to execute one of the jobs in the memory, if the job waits for some task (ex: I/O operations) the operating system switches to another job.

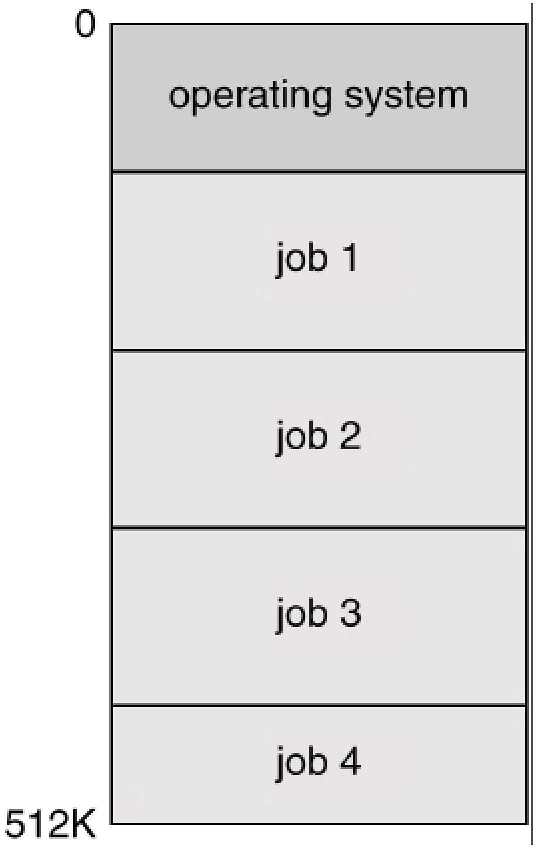


(a) Uniprogramming



(c) Multiprogramming with three programs

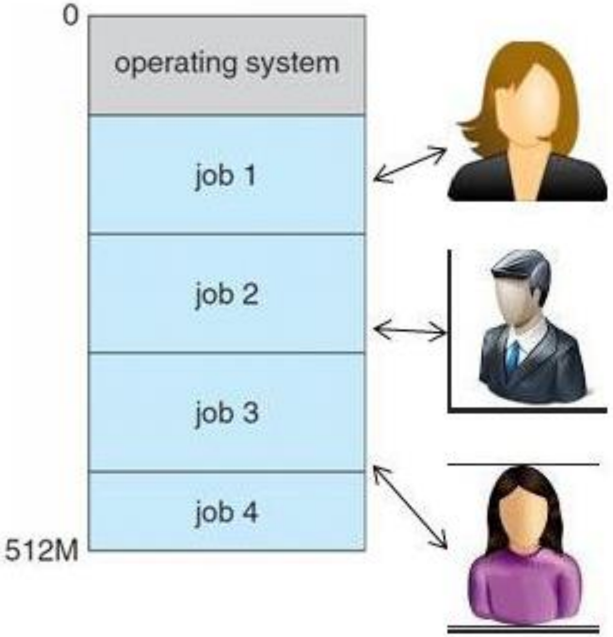
M emory Layout for Multiprogrammed Systems

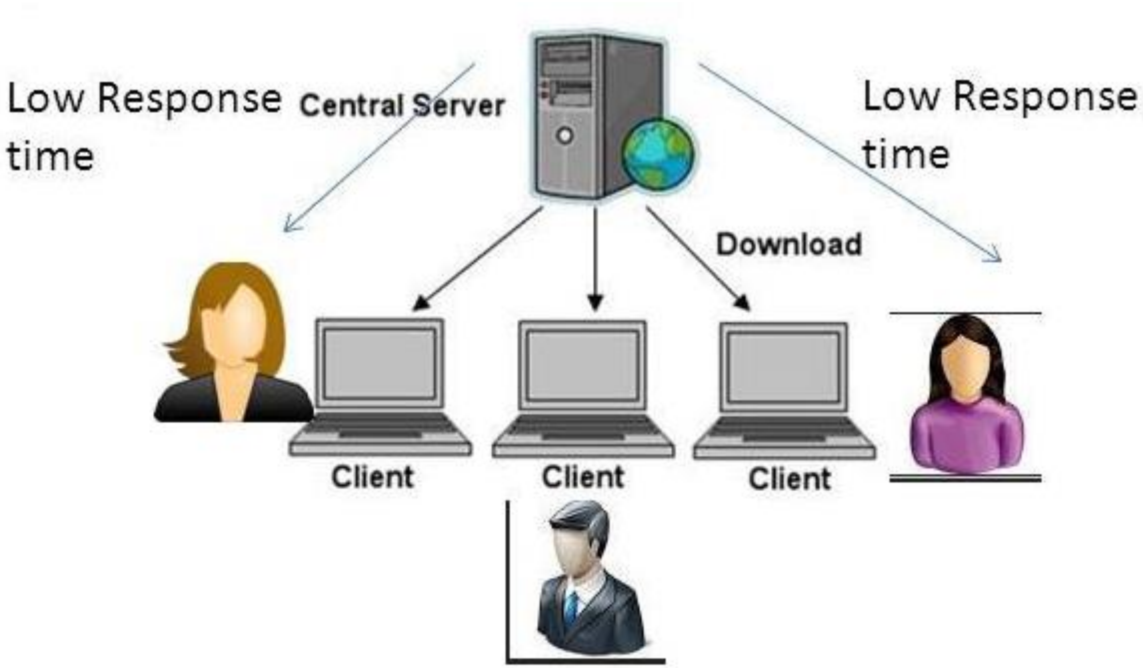


Historical overview

Mainframe system:

- Time-Sharing Systems:
  - A logical extension of multiprogramming
  - The CPU executes multiple jobs by switching among them, but switch occur so frequently that the user can interact with each program while it is running
  - It provides direct communication between the user and the system using a mouse or a keyboard
  - It allows more than one user to share the computer simultaneously


0
operating system
job 1
job 2
job 3
job 4
512M


Low Response time
Central Server
Low Response time
Download
Client
Client
Client

Historical overview

Personal computers – computer system

- computer system dedicated to a single user.
- I/O devices – keyboards, mice, display screens, small printers.
- User convenience and responsiveness.
- Can adopt technology developed for larger operating system’ often individuals have sole use of computer and do not need advanced CPU utilization of protection features.

Historical overview

Multiprocessor Systems (parallel system)

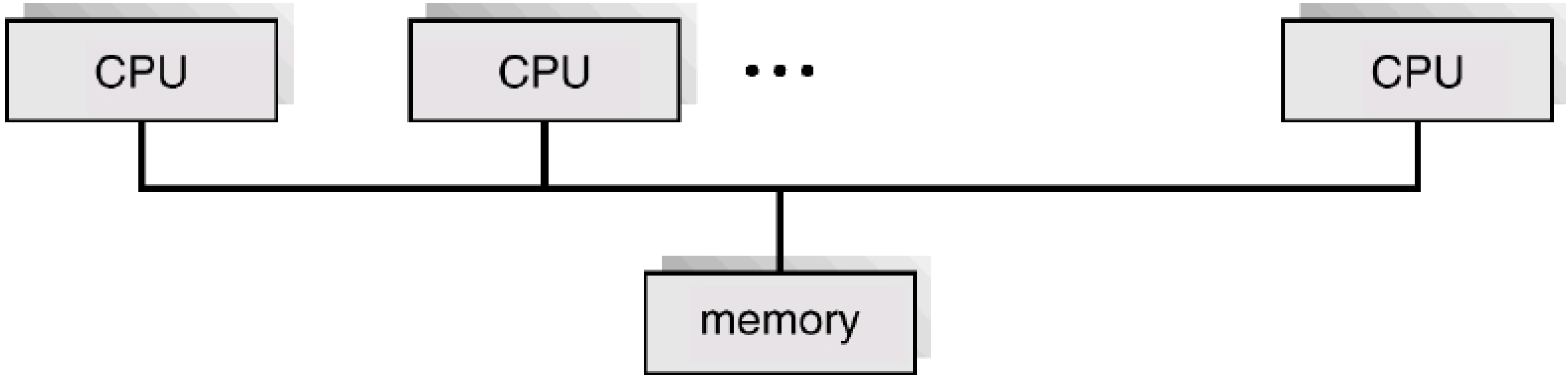
- Multiprocessor systems with more than one CPU in close communication.
- *Tightly coupled system* – processors share memory and a clock; communication usually takes place through the shared memory.
- Advantages of parallel system:
  - Increased *throughput: more work is done in less time*
  - Economical*: save money than multiple single processor system as they share peripherals, mass storage and power supplies*
  - Increased reliability: failure of one processor will not halt the system

Historical overview

Multiprocessor Systems (parallel system)

- *Symmetric multiprocessing (SMP)*
  - Each processor runs an identical copy of the operating system.
  - Many processes can run at once without performance deterioration.
  - Most modern operating systems support SMP
- *Asymmetric multiprocessing*
  - Each processor is assigned a specific task; master processor schedules and allocates work to slave processors.
  - More common in extremely large systems

Symmetric Multiprocessing Architecture



Historical overview

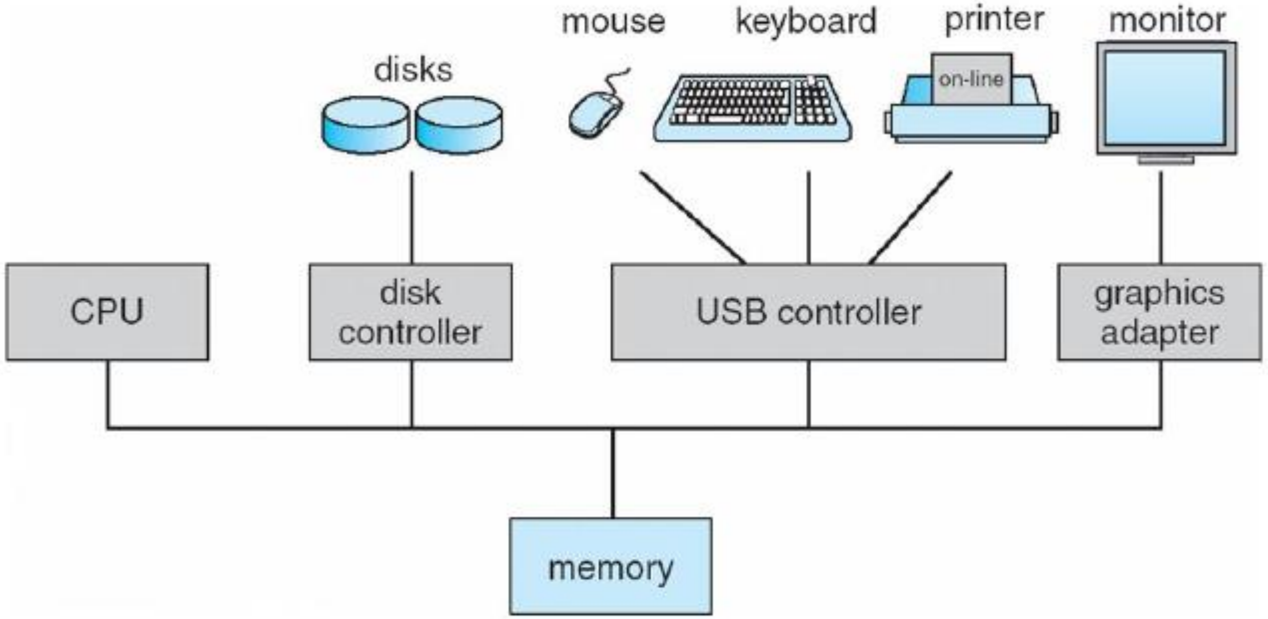
Distributed System

- Distribute the computation among several physical processors.
- *Loosely coupled system* – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines.
- Advantages of distributed systems.
  - Resources Sharing
  - Computation speed up – load sharing
  - Reliability
  - Communications

Computer System Organization

Computer-system operation

- One or more CPUs, device controllers connect through common bus providing access to shared memory
- Concurrent execution of CPUs and devices competing for memory cycles



Computer-System Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an interrupt

Common Functions of Interrupts

- Interrupt transfers control to the interrupt service routine generally, through the **interrupt vector**, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction
- Hardware may trigger an interrupt at any time sending a signal to the CPU.
- Software may trigger an interrupt executing a special operation called a **system call.**

Digital Data Representation

- **Bit**: The smallest unit of data that a binary computer can recognize (a single 1 or 0)
- **Byte** = 8 bits
- Byte terminology used to express the size of documents and other files, programs, etc.

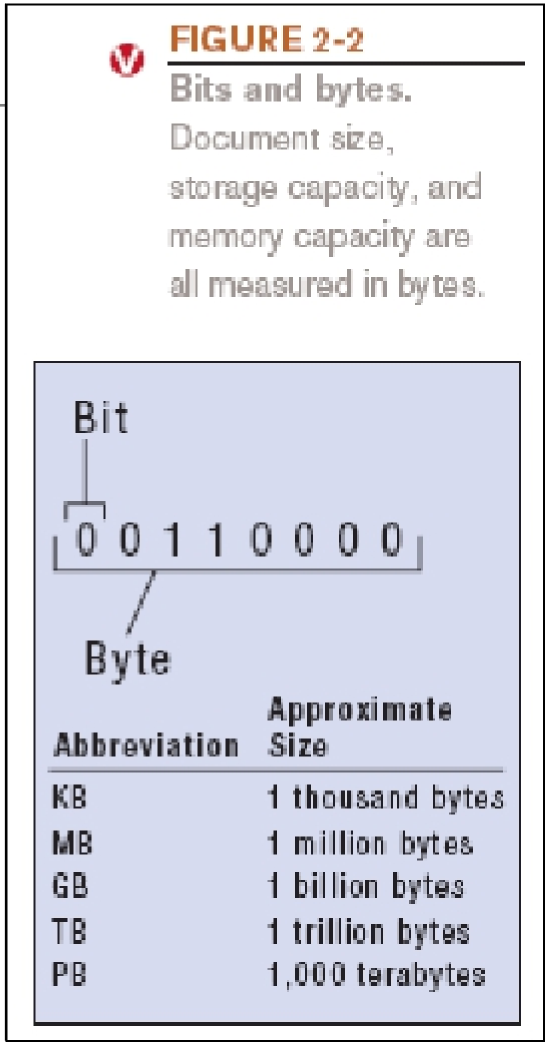


FIGURE 2-2

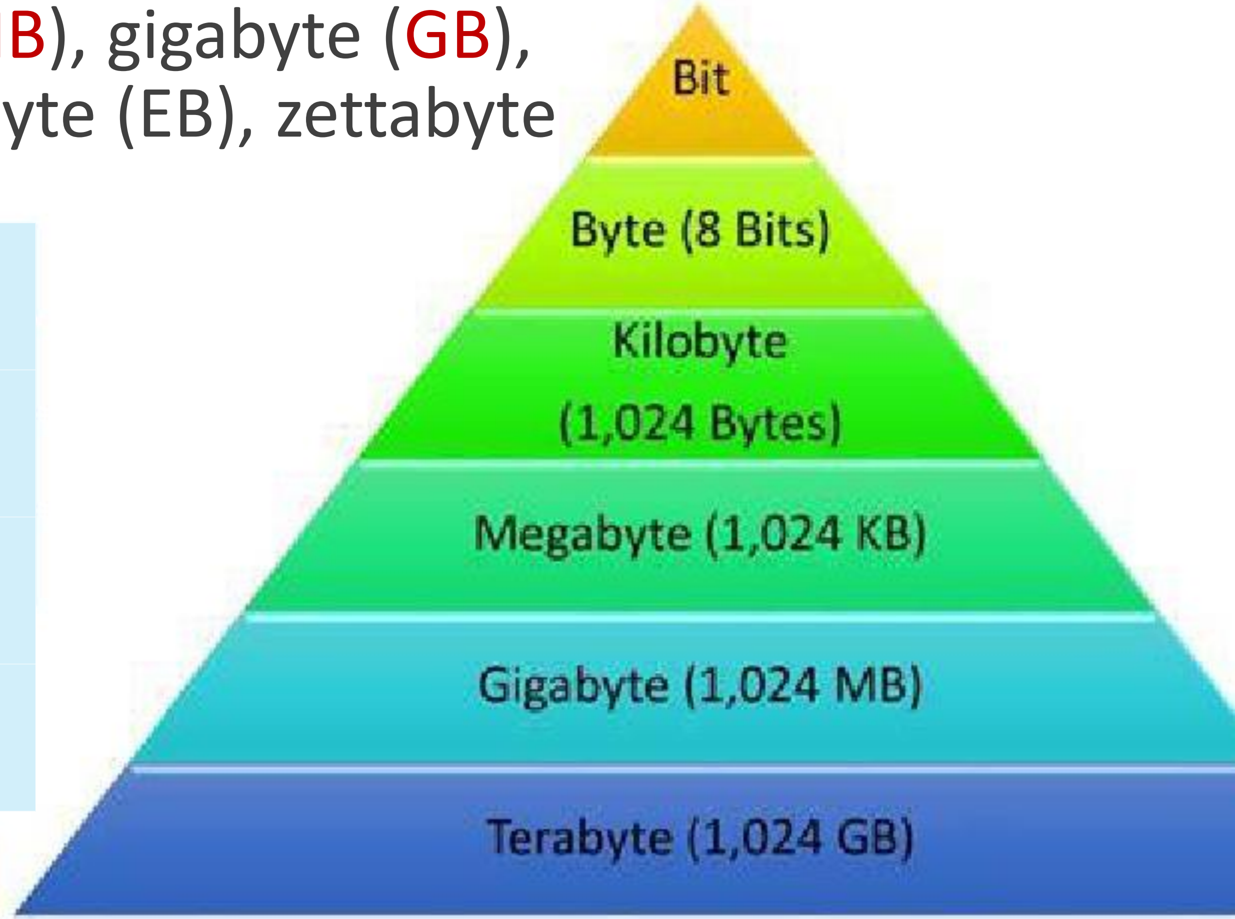
Bits and bytes. Document size, storage capacity, and memory capacity are all measured in bytes.

Abbreviation	Approximate Size
KB	1 thousand bytes
MB	1 million bytes
GB	1 billion bytes
TB	1 trillion bytes
PB	1,000 terabytes

Digital Data Representation

- Prefixes are often used to express larger quantities of bytes: kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB), petabyte (PB), exabyte (EB), zettabyte (ZB), yottabyte (YB).

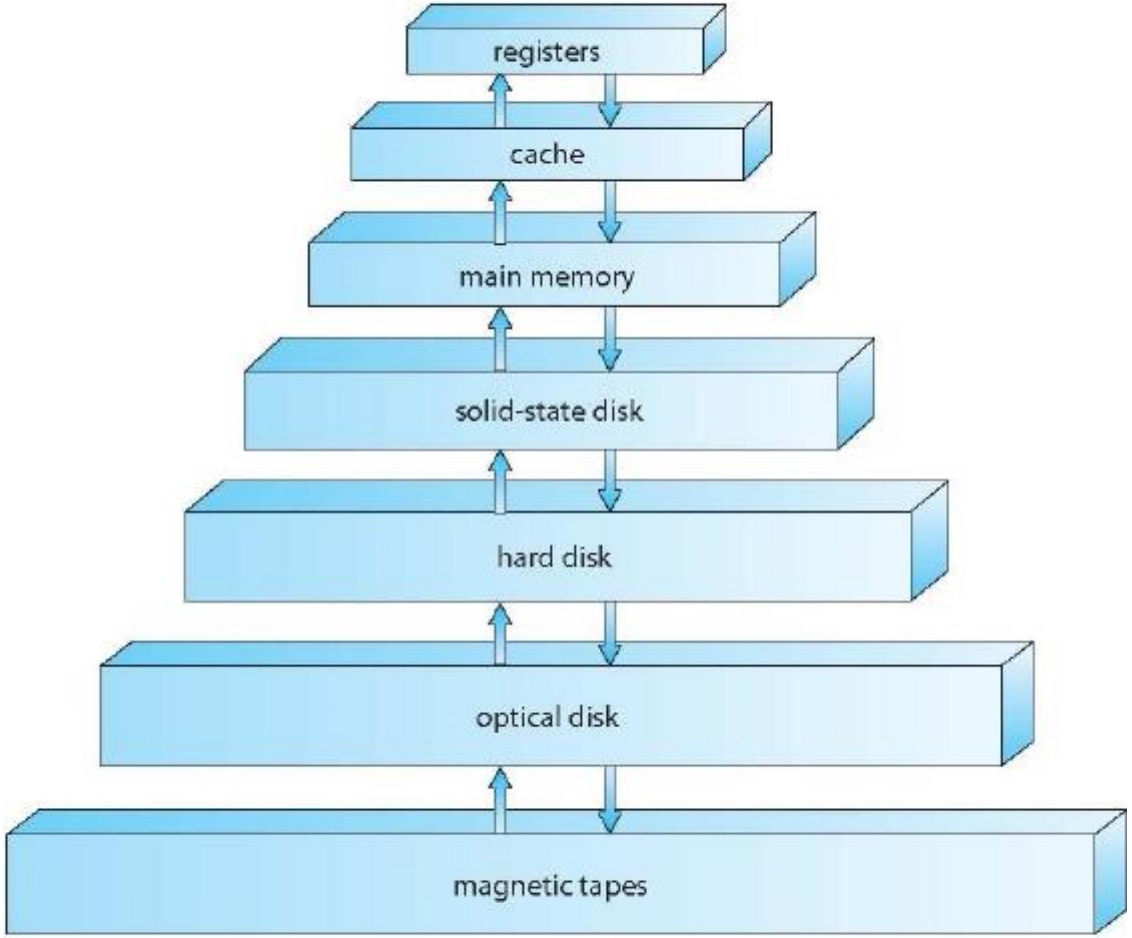
1Kilobytes(KB)	2^{10}bytes	1024bytes
1Megabytes(MG)	2^{20}bytes	1024KB
1Gegabytes(GB)	2^{30}bytes	1024MB
1Terabytes(TB)	2^{40}bytes	1024GB



Storage Hierarchy

- Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility
- **Caching** – copying information into faster storage system; main memory can be viewed as a cache for secondary storage
- **Device Driver** for each device controller to manage I/O
  - Provides uniform interface between controller and kernel

Storage-DeviceHierarchy



Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - **Random access**
  - Typically **volatile**
- Secondary storage – extension of main memory that provides large**nonvolatile** storage capacity
- Hard disks – rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into **tracks**, which are subdivided into **sectors**
  - The **disk controller** determines the logical interaction between the device and the computer
- **Solid-state disks** – faster than hard disks, nonvolatile
  - Various technologies
  - Becoming more popular

Caching

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache management important design problem
  - Cache size and replacement policy