

# Module 1: Introduction

- What is an operating system?
- Simple Batch Systems
- Multiprogramming Batched Systems
- Time-Sharing Systems
- Personal-Computer Systems
- Parallel Systems
- Distributed Systems
- Real-Time Systems

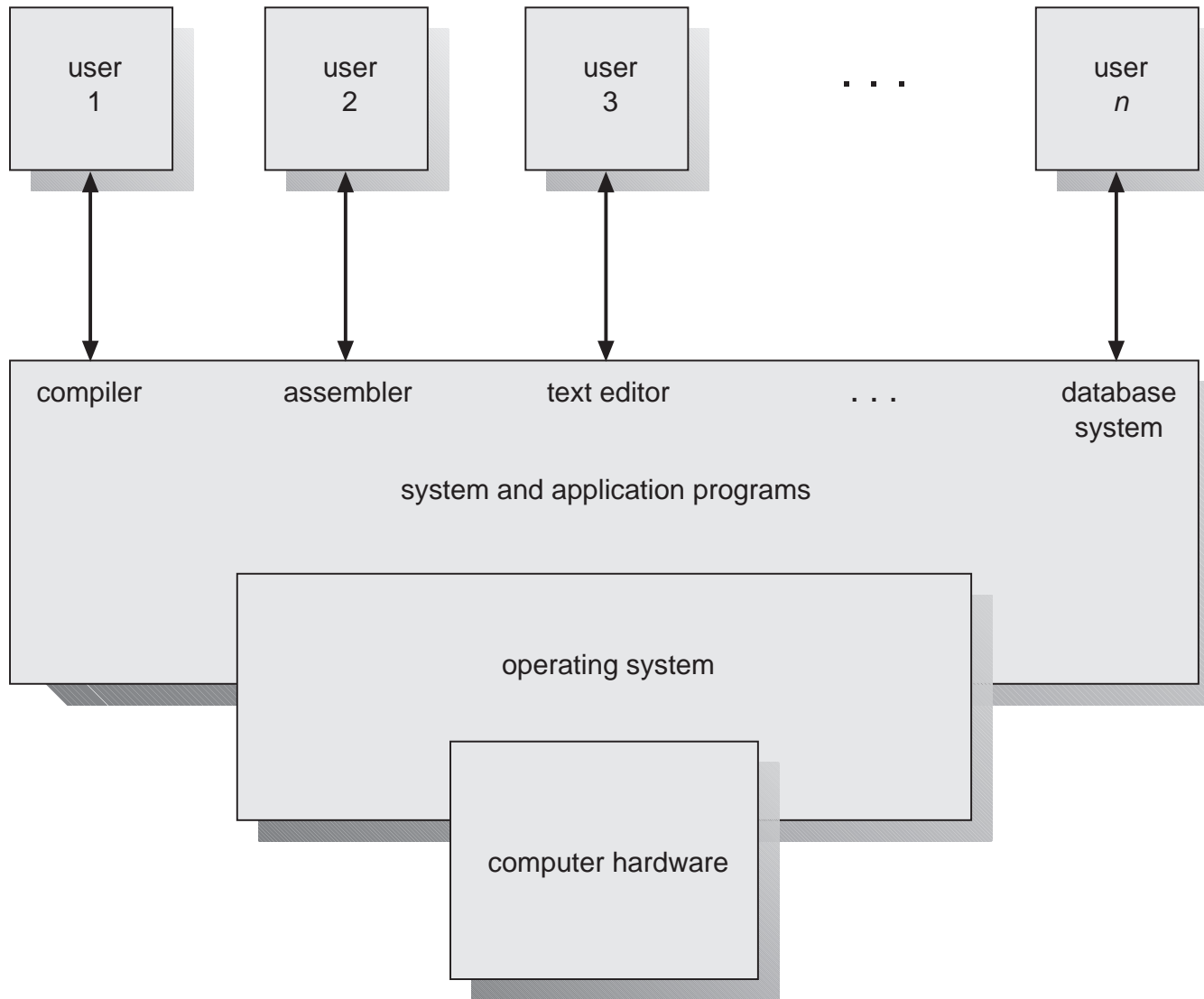
# What is an Operating System?

- 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.

# Computer System Components

1. Hardware – provides basic computing resources (CPU, memory, I/O devices).
2. Operating system – controls and coordinates the use of the hardware among the various application programs for the various users.
3. Applications programs – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs).
4. Users (people, machines, other computers).

# Abstract View of System Components



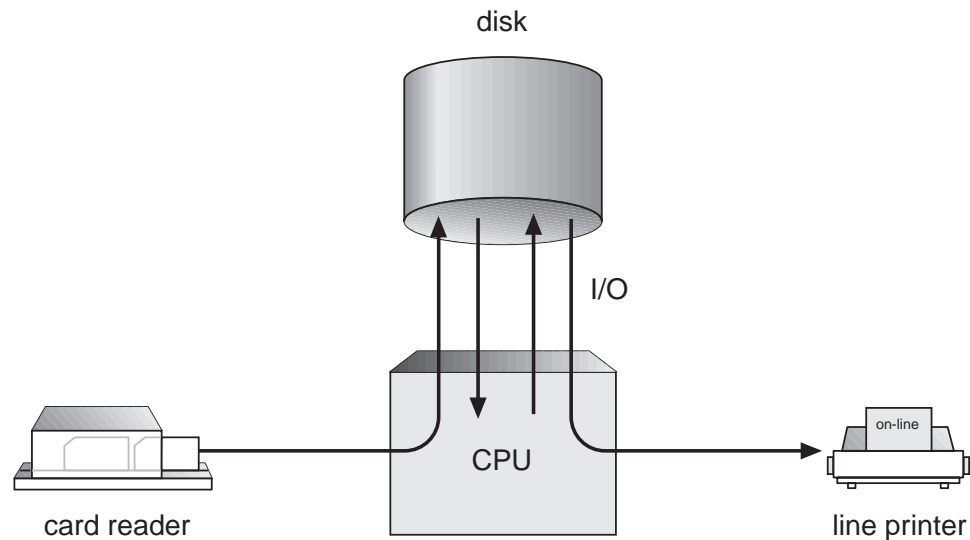
## Operating System Definitions

- Resource allocator – manages and allocates resources.
- Control program – controls the execution of user programs and operation of I/O devices.
- Kernel – the one program running at all times (all else being application programs).

## Early Systems – Bare machine (early 1950s)

- Structure
  - Large machines run from console
  - Single user system
  - Programmer/User as operator
  - Paper tape or punched cards
- Early Software
  - Assemblers, compilers
  - Linkers, loaders
  - Libraries of common subroutines
  - Device drivers
- Secure
- Inefficient use of expensive resources
  - Low CPU utilization
  - Significant amount of setup time

# Spooling



- Overlap I/O of one job with computation of another job. While executing one job, the OS:
  - reads next job from card reader into a storage area on the disk (job queue).
  - outputs printout of previous job from disk to printer.
- *Job pool* – data structure that allows the OS to select which job to run next in order to increase CPU utilization.

## OS Features Needed for Multiprogramming

- I/O routine supplied by the system.
- Memory management – the system must allocate the memory to several jobs.
- CPU scheduling – the system must choose among several jobs ready to run.
- Allocation of devices.

## Time-Sharing Systems– Interactive Computing

- The CPU is multiplexed among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory).
- A job is swapped in and out of memory to the disk.
- On-line communication between the user and the system is provided; when the operating system finishes the execution of one command, it seeks the next “control statement” not from a card reader, but rather from the user’s keyboard.
- On-line file system must be available for users to access data and code.

## Personal-Computer Systems

- *Personal computers* – 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 systems; often individuals have sole use of computer and do not need advanced CPU utilization or protection features.

# Parallel Systems

- 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 systems:
  - Increased *throughput*
  - Economical
  - Increased reliability
    - \* *graceful degradation*
    - \* *fail-soft systems*

## Parallel Systems (Cont.)

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

# Distributed Systems

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

# Real-Time Systems

- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems.
- Well-defined fixed-time constraints.
- *Hard real-time system*.
  - Secondary storage limited or absent; data stored in short-term memory, or read-only memory (ROM).
  - Conflicts with time-sharing systems; not supported by general-purpose operating systems.
- *Soft real-time system*.
  - Limited utility in industrial control or robotics.
  - Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.