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

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:

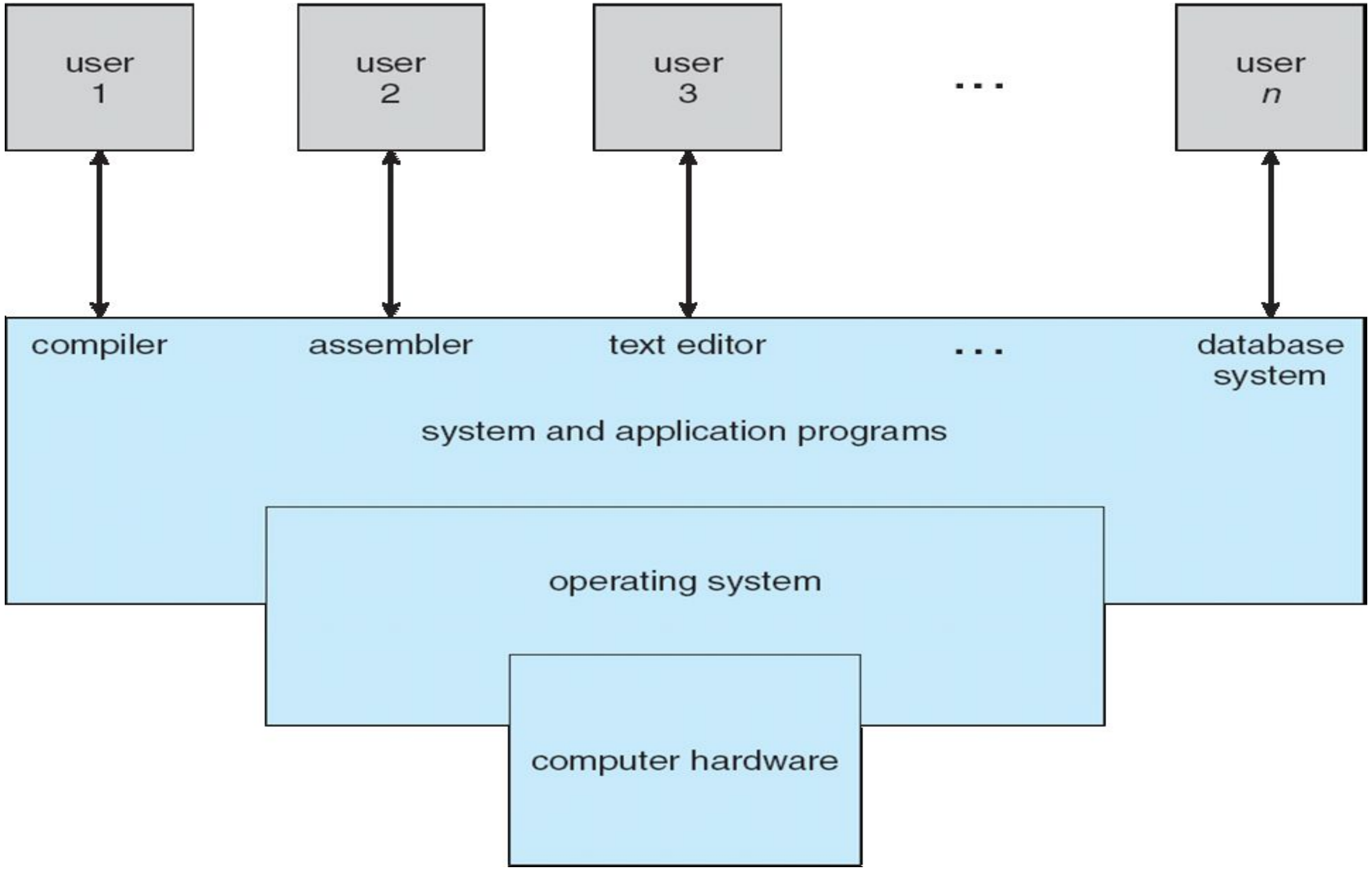
- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

Computer System Structure

Computer system can be divided into four components:

- **Hardware** – provides basic computing resources
  - CPU, memory, I/O devices
- **Operating system**
  - Controls and coordinates use of hardware among various applications and users
- **Application programs** – 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
- **Users**
  - People, machines, other computers

Four Components of a Computer System



What Operating Systems Do – Users View

Users want convenience, **ease of use** and **good performance**

- Don’t care about **resource utilization**

But shared computer (**mainframe** or **minicomputer)** must keep all users happy

Users of dedicate systems like **workstations** have dedicated resources but frequently use shared resources from **servers**

Handheld computers are resource poor, optimized for usability and battery life

Some computers have little or no user interface, such as embedded computers in devices and automobiles

What Operating Systems Do – System View

OS is a **resource allocator**

- Manages all resources
- Decides between conflicting requests for efficient and fair resource use

OS is a **control program**

- Controls execution of programs to prevent errors and improper use of the computer

Operating System Definition (Cont.)

No universally accepted definition

"Everything a vendor ships when you order an operating system" is a good approximation

- But varies wildly

"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.

Computer-System Architecture

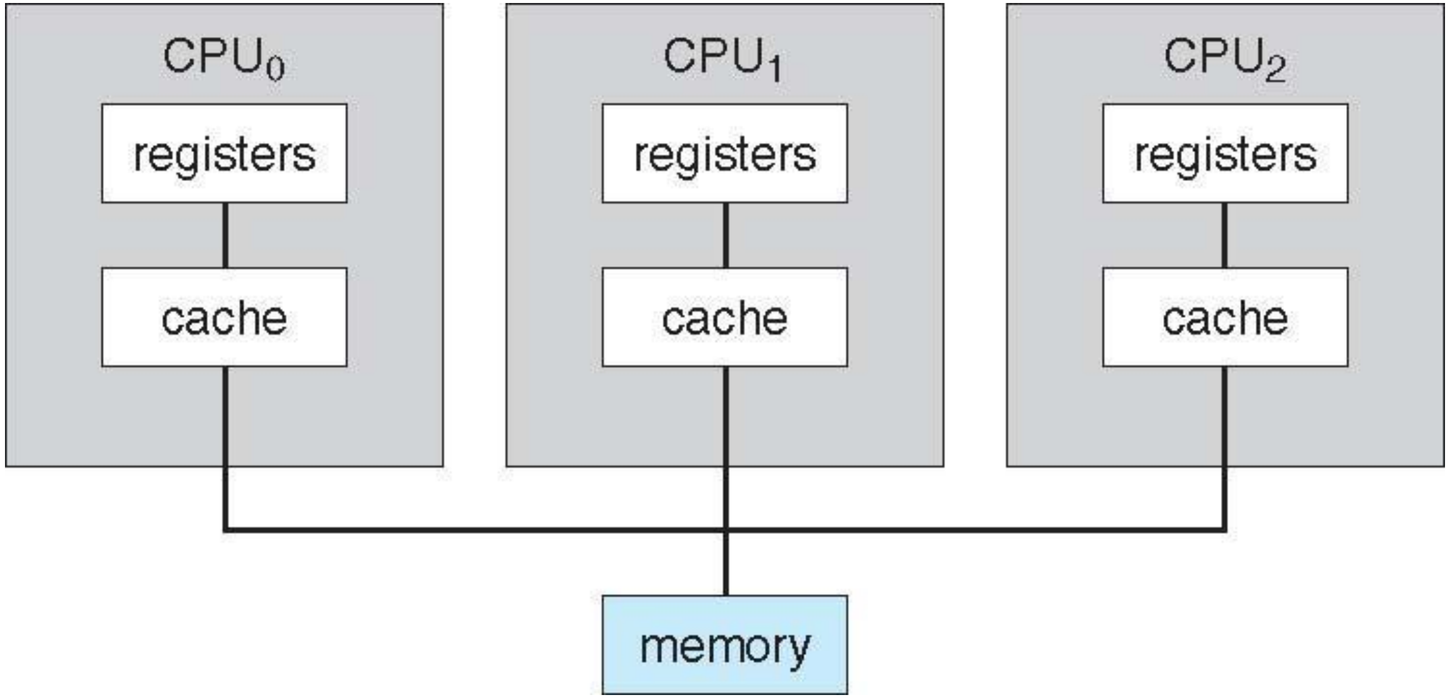
1.Most systems use a **single general-purpose processor**

- Most systems have special-purpose processors as well

2.Multiprocessors systems growing in use and importance

- Advantages include:
  1. **Increased throughput-** The speed up ratio with N processors is not N,but less than N
  2. **Economy of scale-** They can share peripherals,storage and power supplies
  3. **Increased reliability** – graceful degradation or fault tolerance
- Two types:
  1. **Symmetric Multiprocessing** – each processor performs all tasks
  2. **Asymmetric Multiprocessing** – each processor is assigned a specie task.

Symmetric Multiprocessing Architecture

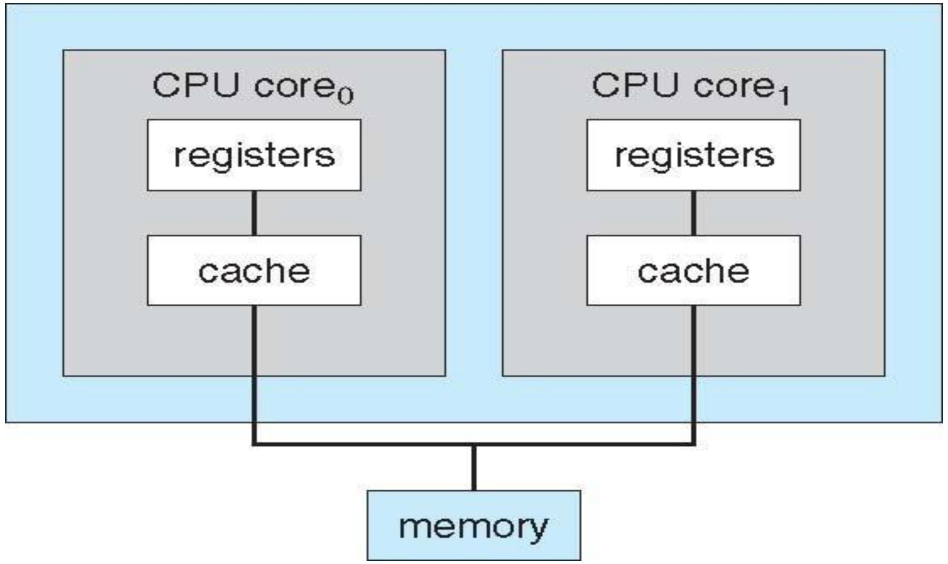


A Dual-Core Design

Multi-chip and **multicore**

Systems containing all chips

- Chassis containing multiple separate systems



Operating System Structure

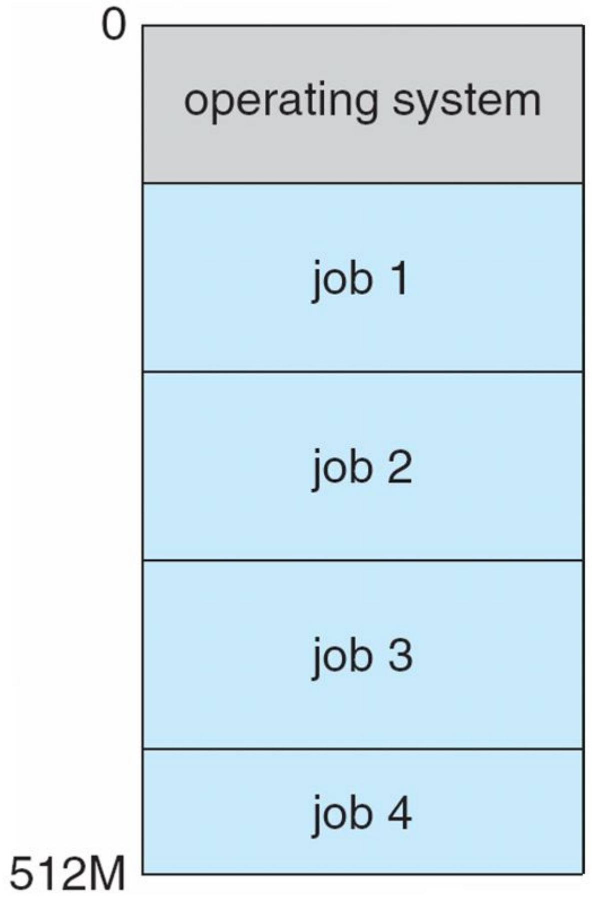
Multiprogramming (**Batch system**) needed for efficiency

- 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
- One job selected and run via **job scheduling**
- When it has to wait (for I/O for example), OS switches to another job

Timesharing (**multitasking**) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing

- **Response time** should be < 1 second
- Each user has at least one program executing in memory □**process**
- If several jobs ready to run at the same time □ **CPU scheduling**
- If processes don’t fit in memory, **swapping** moves them in and out to run
- **Virtual memory** allows execution of processes not completely in memory

Memory Layout for Multiprogrammed System



Operating-System Operations

Interrupt driven (hardware and software)

- Hardware interrupt by one of the devices
- Software interrupt (**exception** or **trap):**
  - Software error (e.g., division by zero)
  - Request for operating system service
  - Other process problems include infinite loop, processes modifying each other or the operating system

Operating-System Operations (cont.)

1.Dual-mode operation allows OS to protect itself and other system components

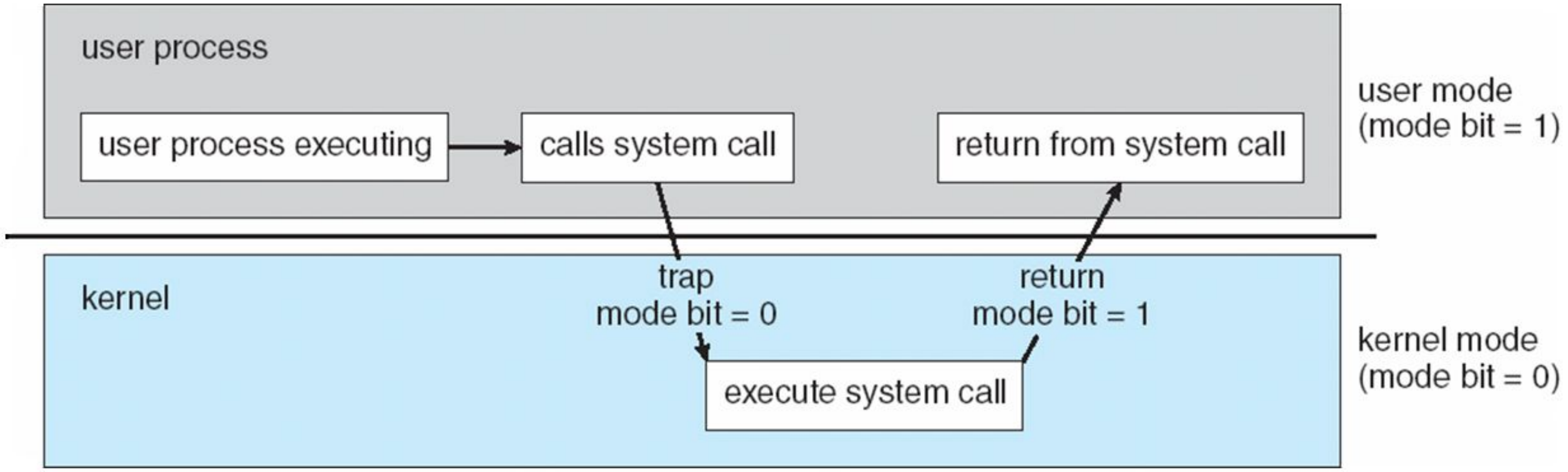
- **User mode** and **kernel mode**
- **Mode bit** provided by hardware
  - Provides ability to distinguish when system is running user code or kernel code
  - Some instructions designated as **privileged**, only executable in kernel mode
  - System call changes mode to kernel, return from call resets it to user

Increasingly CPUs support multi-mode operations

- i.e. **virtual machine manager** (**VMM**) mode for guest **VMs**

Transition from User to Kernel Mode

2. **Timer** to prevent infinite loop / process hogging resources
   - Timer is set to interrupt the computer after some time period
   - Keep a counter that is decremented by the physical clock.
   - Operating system set the counter (privileged instruction)
   - When counter zero generate an interrupt
   - Set up before scheduling process to regain control or terminate program that exceeds allotted time



Process Management Activities

The operating system is responsible for the following activities in connection with process management:

- 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

Memory Management

To execute a program all (or part) of the instructions and data must be in memory

Memory management activities

- Keeping track of which parts of memory are currently being used and by whom
- Deciding which processes (or parts thereof) and data to move into and out of memory
- Allocating and deallocating memory space as needed

Storage Management

OS provides uniform, logical view of information storage

- Abstracts physical properties to logical storage unit - **file**
- Each medium is controlled by device (i.e., disk drive, tape drive)
  - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)

File-System management

- Files usually organized into directories
- Access control on most systems to determine who can access what
- OS activities include
  - 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

Mass-Storage Management

Usually disks used to store data that does not fit in main memory or data that must be kept for a “long” period of time

Proper management is of central importance

Entire speed of computer operation hinges on disk subsystem and its algorithms

OS activities

- Free-space management
- Storage allocation
- Disk scheduling

Some storage need not be fast

- Tertiary storage includes optical storage, magnetic tape
- Still must be managed – by OS or applications
- Varies between WORM (write-once, read-many-times) and RW (read-write)

I/O Subsystem

One purpose of OS is to hide peculiarities of hardware devices from the user

I/O subsystem responsible for

- Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
- General device-driver interface
- Drivers for specific hardware devices

Protection and Security

Protection – any mechanism for controlling access of processes or users to resources defined by the OS

Security – defense of the system against internal and external attacks

- Huge range, including denial-of-service, worms, viruses, identity theft, theft of service

Systems generally first distinguish among users, to determine who can do what

- User identities (**user IDs**, security IDs) include name and associated number, one per user
- User ID then associated with all files, processes of that user to determine access control
- Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
- **Privilege escalation** allows user to change to effective ID with more rights