What is an Operating System?

- Three views of an operating system

  Application View: what services does it provide?

  System View: what problems does it solve?

  Implementation View: how is it built?

An operating system is part cop, part facilitator.

Application View of an Operating System

- The OS provides an execution environment for running programs.
  - The execution environment provides a program with the processor time and memory space that it needs to run.
  - The execution environment provides interfaces through which a program can use networks, storage, I/O devices, and other system hardware components.
    - Interfaces provide a simplified, abstract view of hardware to application programs.
  - The execution environment isolates running programs from one another and prevents undesirable interactions among them.

Other Views of an Operating System

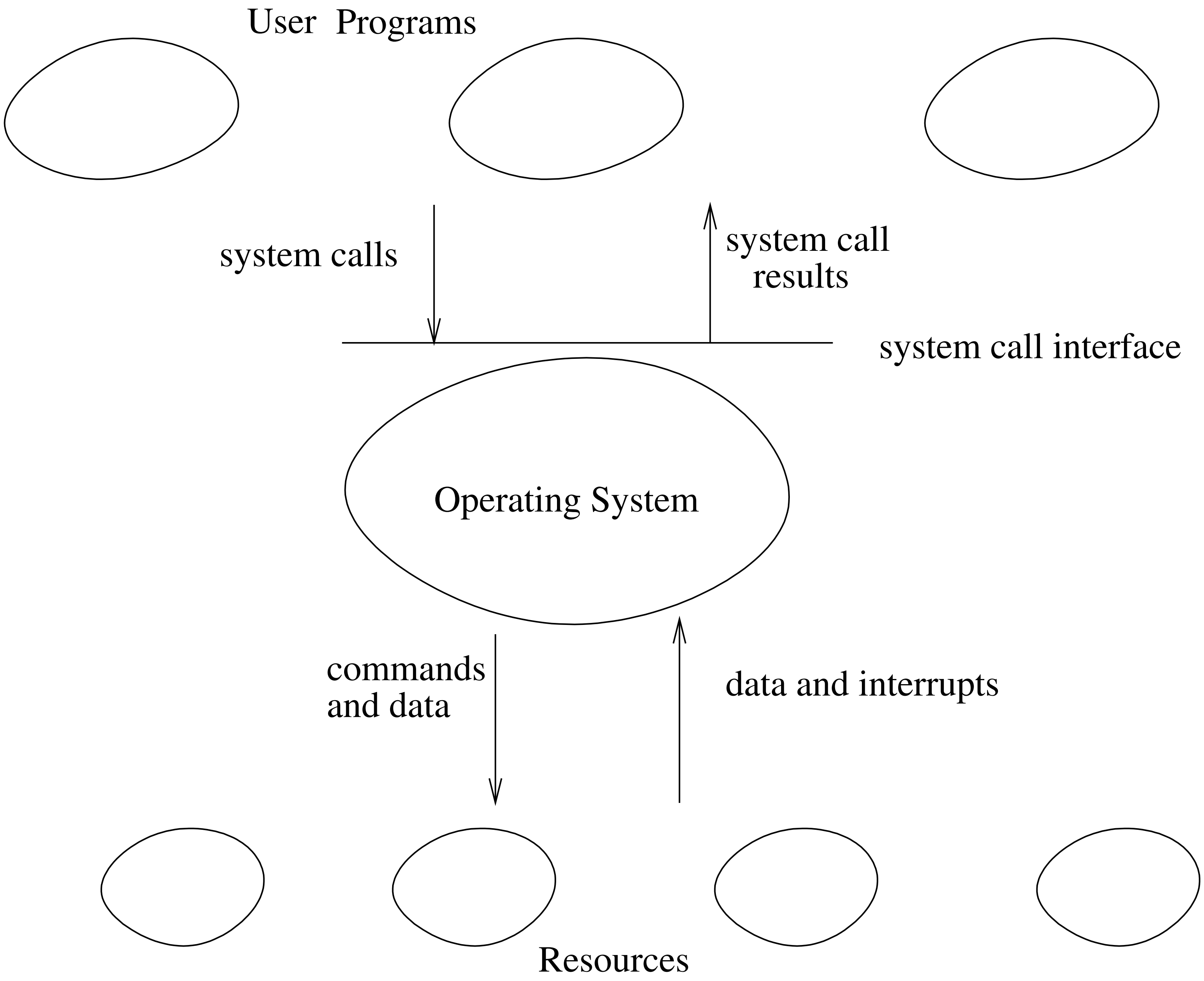
System View: The OS manages the hardware resources of a computer system.

- Resources include processors, memory, disks and other storage devices, network interfaces, I/O devices such as keyboards, mice and monitors, and so on.
- The operating system allocates resources among running programs. It controls the sharing of resources among programs.
- The OS itself also uses resources, which it must share with application programs.

Implementation View: The OS is a concurrent, real-time program.

- Concurrency arises naturally in an OS when it supports concurrent applications, and because it must interact directly with the hardware.
- Hardware interactions also impose timing constraints.

Schematic View of an Operating System



Operating System Abstractions

- The execution environment provided by the OS includes a variety of abstract entities that can be manipulated by a running program. Examples:

  files and file systems: abstract view of secondary storage

  address spaces: abstract view of primary memory

  processes, threads: abstract view of program execution

  sockets, pipes: abstract view of network or other message channels

- This course will cover
  - why these abstractions are designed the way they are
  - how these abstractions are manipulated by application programs
  - how these abstractions are implemented by the OS

Course Outline

- Introduction
- Processes and Threads
- Concurrency and Synchronization
- Address Spaces and Virtual Memory
- Scheduling
- Devices and Device Management
- File Systems
- Interprocess Communication and Networking
- Security