## The Operating System (OS) in Context

- Components of a Computer System

  - *Hardware* – provides basic computing resources (CPU, memory, I/O devices)

  - **Operating system** – controls and coordinates the use of the hardware among the various application programs for the various users

  - *Applications programs* – define the ways in which the system resources are used to solve the computing problems of the users (compilers, databases, video games, business programs)

  - *Users* (people, machines, computers)
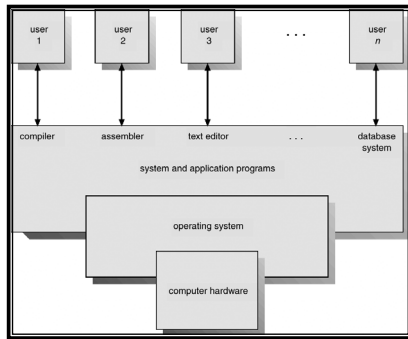
What is an Operating System?

- An *operating system* (OS) is the interface between the user and the hardware

  - It implements a virtual machine that is easier to program than bare hardware

- An OS provides standard **services** (functionality) which are implemented on the hardware, including:

  - Processes, CPU scheduling, memory management, file system, networking

- The OS **coordinates** multiple applications and users (multiple processes) in a fair and efficient manner

- ↪ The goal in OS development is to make the machine both **convenient** to use (a software engineering problem) as well as **efficient** (a system and engineering problem)

Modern OS Functionality

- Textbook talks about OS as a:

  - Control program — manages the execution of user programs, prevents errors and improper use of the computer

  - Resource allocator — CPU time, memory space, file space, I/O devices

- OS must provide:

  - Processes & CPU scheduling
    - Multiple processes active concurrently
    - Processes may need to communicate
    - Processes may require mutually-exclusive access to some resource

  - Memory management — must allocate memory to processes, move processes between disk and memory

  - File system — must allocate space for storage of programs and data on disk
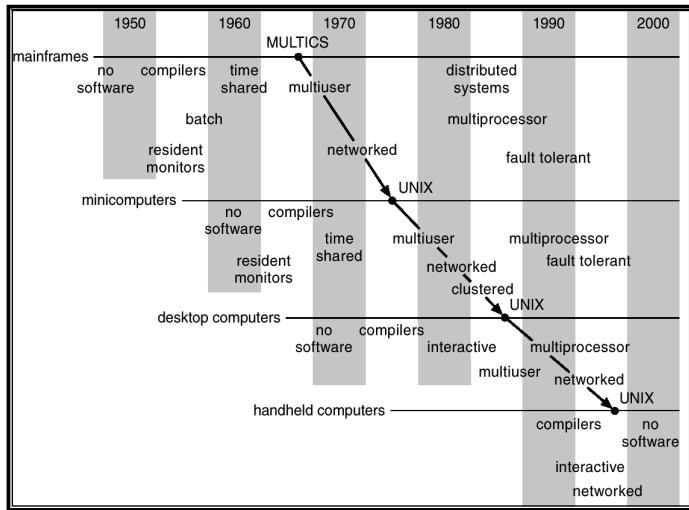
Why Study Operating Systems?

- Abstraction — how do you give the users the illusion of infinite resources (CPU time, memory, file space)?

- System design —tradeoffs between:

  - performance and convenience of these abstractions

  - performance and simplicity of OS

  - functionality in hardware or software

- Primary intersection point — OS is the point where hardware, software, programming languages, data structures, and algorithms all come together

- Curiosity — "look under the hood"

- "Operating systems are among the most complex pieces of software yet developed", William Stallings, 1994

Next Few Lectures

■ L02 (Wednesday) — OS History

	1950	1960	1970	1980	1990	2000

mainframes

MULTICS

no software — compilers — time shared — multiuser

distributed systems

batch

multiprocessor

resident monitors

networked

fault tolerant

UNIX

minicomputers

no software — compilers

time shared — multiuser

multiprocessor

resident monitors

networked

fault tolerant

clustered

UNIX

desktop computers

no software — compilers

interactive

multiprocessor

multiuser

networked

handheld computers

UNIX

compilers — no software

interactive

networked

■ L03 (Friday) — Computer System Structures (review of some important features provided by the architecture)

■ L04 (Wednesday) — OS Structures