Introduction

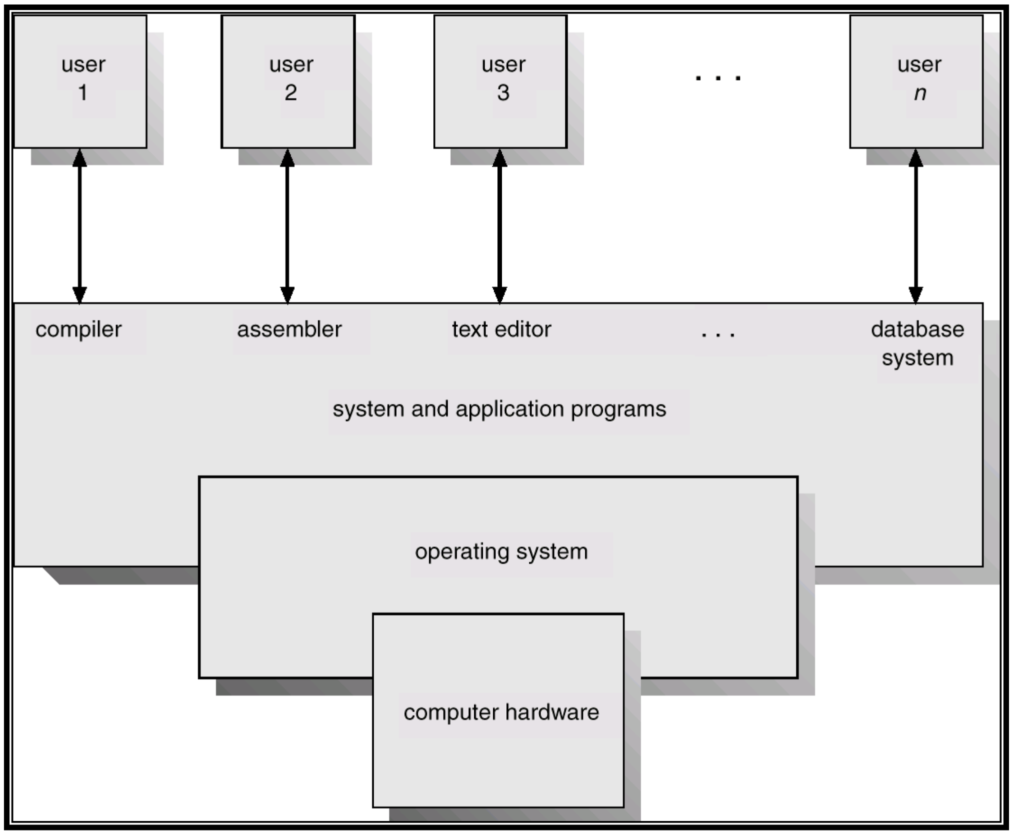
What is an Operating System?

- User-centric definition
  - A program that acts as an intermediary between a user of a computer and the computer hardware
  - Defines an interface for the user to use services provided by the system
  - Provides a "view" of the system to the user
- System-centric definition
  - Resource allocator – manages and allocates resources
  - Control program – controls the execution of user programs and operations of I/O devices

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, databases, games, …).
4. Users (people, machines, other computers).

Abstract View of System Components



Types of Systems

- Batch Systems
  - Multiple jobs, but only one job in memory at one time and executed (till completion) before the next one starts
- Multiprogrammed Batch Systems
  - Multiple jobs in memory, CPU is multiplexed between them
  - CPU-bound vs I/O bound jobs
- Time-sharing Systems
  - Multiple jobs in memory and on disk, CPU is multiplexed among jobs in memory, jobs swapped between disk and memory
  - Allows interaction with users

- Personal Computers
  - Dedicated to a single user at one time
- Multiprocessing Systems
  - More than one CPU in a single machine to allocate jobs to
  - Symmetric Multiprocessing, NUMA machines …
  - Multicore
- Other Parallel Systems, Distributed Systems, Clusters…
  - Different types of systems with multiple CPUs/Machines
- Real Time Systems
  - Systems to run jobs with time guarantees
- Other types possible depending on resources in the machine, types of jobs to be run…

- OS design depends on the type of system it is designed for
- Our primary focus in this course:
  - Uniprocessor, time-sharing systems running general purpose jobs from users
  - Effect of multicore/multiprocessors
- Will discuss some other topics at end

Resources Managed by OS

- Physical
  - CPU, Memory, Disk, I/O Devices like keyboard, monitor, printer
- Logical
  - Process, File, …

Main Components of an OS

- Resource-Centric View
  - Process Management
  - Main Memory Management
  - File Management
  - I/O System Management
  - Secondary Storage Management
  - Security and Protection System
  - Networking (this is now integrated with most OS, but will be covered in the Networks course)
- User-centric view
  - System Calls
  - Command Interpreter (not strictly a part of an OS)

Process Management

- A *process* is a program in execution.
- Needs certain resources to accomplish its task
  - CPU time, memory, files, I/O devices…
- OS responsibilities
  - Process creation and deletion.
  - Process suspension and resumption.
  - Provide mechanisms for:
    - process synchronization
    - interprocess communication

Main-Memory Management

- OS responsibilities
  - Keep track of which parts of memory are currently being used and by whom
  - Decide which processes to load when memory space becomes available
  - Allocate and deallocate memory space as needed

File Management

- OS responsibilities
  - File creation, deletion, modification
  - Directory creation, deletion, modification
  - Support of primitives for manipulating files and directories
  - Mapping files onto secondary storage.
  - File backup on stable (nonvolatile) storage media

I/O System Management

- The I/O system consists of:
  - A buffer-caching system
  - Device driver interface
  - Drivers for specific hardware devices

Secondary-Storage Management

- Most modern computer systems use disks as the principle on-line storage medium, for both programs and data.
- OS responsibilities
  - Free space management
  - Storage allocation
  - Disk scheduling

Security and Protection System

- *Protection* refers to a mechanism for controlling access by programs, processes, or users to both system and user resources.
- The protection mechanism must:
  - distinguish between authorized and unauthorized usage
  - specify the controls to be imposed
  - provide a means of enforcement

System Calls

- System calls provide the interface between a running program and the OS
  - Think of it as a set of functions available to the program to call (but somewhat different from normal functions, we will see why)
  - Generally available as assembly-language instructions.
  - Most common languages (e.g., C, C++) have APIs that call system calls underneath
- Passing parameters to system calls
  - Pass parameters in *registers*
  - Store the parameters in a table in memory, and the table address is passed as a parameter in a register
  - *Push* (store) the parameters onto the *stack* by the program, and *pop* off the stack by operating system

Command-Interpreter System

- Strictly not a part of OS, but always there
  - the *shell*
- Allows user to give commands to OS, interpretes the commands and executes them
  - Calls appropriate functions/system calls
  - You will write one in your lab