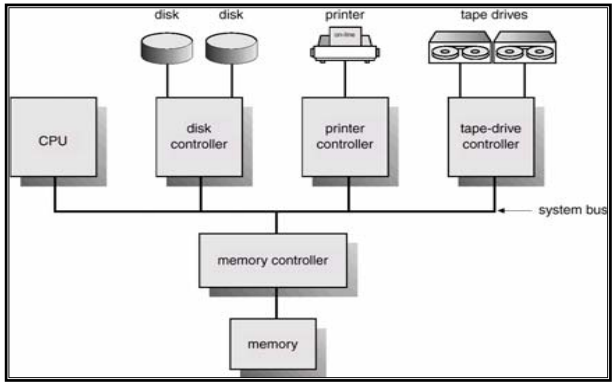
Computer System Organization

CS 256/456
Dept. of Computer Science, University of Rochester

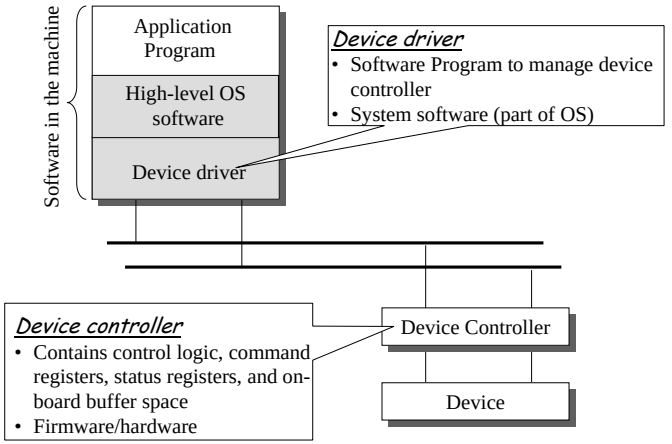
Overview

- Recap of last class
  - What is operating system?
  - Roles of operating system?
  - Types of operating systems.
- Computer hardware issues
  - Hardware interface (I/O interface) to OS
  - Hardware support for OS mechanisms
- Operating system organization
  - OS components
  - OS architectures

Computer-System Architecture



The Device-Controller-Software Relationship



I/O Operations

- How is I/O done?
  - I/O devices are much slower than CPU
- Synchronous (polling)
  - After I/O starts, busy polling the device status register until it shows the operation completes.
- Asynchronous (interrupt-driven)
  - After I/O starts, control returns to user program without waiting for I/O completion.
  - Device controller later informs CPU that it has finished its operation by causing an *interrupt*.
  - When an interrupt occur, current execution is put on hold; the CPU jumps to a service routine called "interrupt handler".

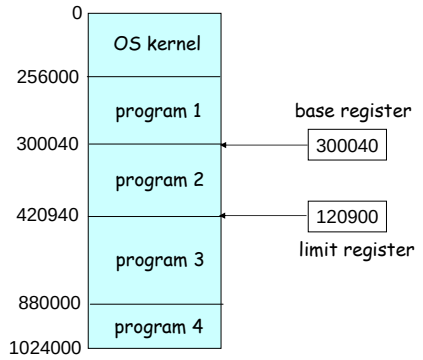
Hardware Support for OS Protection

- User programs (programs not belonging to the OS) are generally not trusted
  - A user program may use unfair amount of resource
  - A user program may maliciously cause other programs or the OS to fail
- Need protection against untrusted user programs
- Provide hardware support to differentiate between at least two modes of operations
  1. *User mode* – execution of user programs
     - untrusted
     - not allowed to have complete/direct access to hardware resources
  2. *Kernel mode* (also *system mode* or *monitor mode*) – execution of the operating system
     - trusted
     - allowed to have complete/direct access to hardware resources

Memory Protection

- Goal of memory protection?
  - A user program can't use arbitrary amount of memory
  - A user program can't access data belonging to the operating system or other user programs.
- How to achieve memory protection?
  - Indirect memory access
    - Memory access with a virtual address which needs to be translated into physical address
  - Add two registers that determine the range of legal addresses a program may access:
    - Base register – holds the smallest legal physical memory address
    - Limit register – contains the size of the range
    - Memory outside the defined range is protected.

Hardware Address Protection



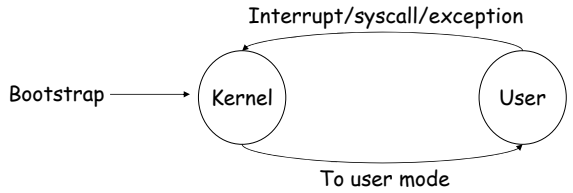
- Address of each memory address is checked against "base" and "base+limit"
- Trap to the OS kernel if it falls outside of the range

Protection of I/O Devices

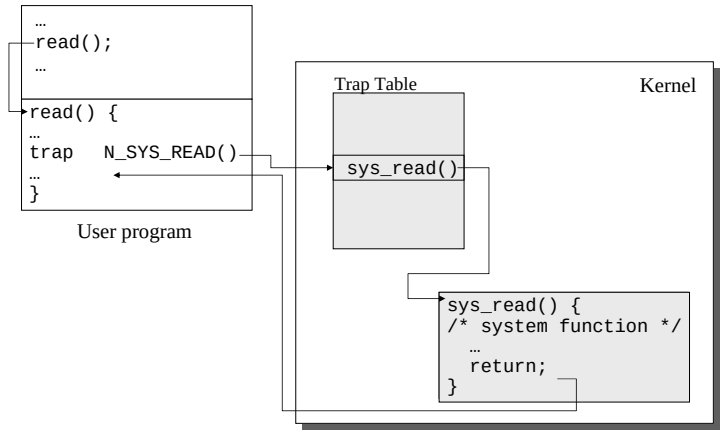
- User programs are not allowed to directly access I/O devices
  - Special I/O instructions can only be used in kernel mode
  - Controller registers can only be accessed in kernel mode
- So I/O interrupt handlers must run in kernel mode.
- How does a user program perform I/O?

Transition between User/Kernel Mode

- When does the machine run in kernel mode?
  - after machine boot
  - interrupt handler
  - system call
  - exception



System Call Using the **Trap** Instruction



CPU Protection

- Goal of CPU protection
  - A user program can't hold the CPU for ever
- *Timer* – interrupts computer after specified period to ensure the OS kernel maintains control.
  - Timer is decremented every clock tick.
  - When timer reaches the value 0, an interrupt occurs.
  - CPU time sharing is implemented in the timer interrupt.

Operation System Organization

- System Components
  - process management
  - memory management
  - I/O system
  - file and storage
  - networking, …
- Operating System Architectures
  - monolithic architecture
  - microkernel architecture
  - layered architecture
  - virtual machines

Process Management

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

Memory Management

- Memory
  - A large array of addressable words or bytes.
  - A data repository shared by the CPU and I/O devices.
- OS responsibility for memory management:
  - Allocate and deallocate memory space as requested.
  - Keep track of which parts of memory are currently being used and by whom.
  - Efficient utilization when the memory resource is heavily competed.

I/O System Management

- A computer needs I/O to interact with outside world:
  - Console/terminal
  - Non-volatile secondary storage – disks
  - Networking
- The I/O system consists of:
  - A buffer-caching system
  - A general device-driver interface
  - Drivers for specific hardware devices

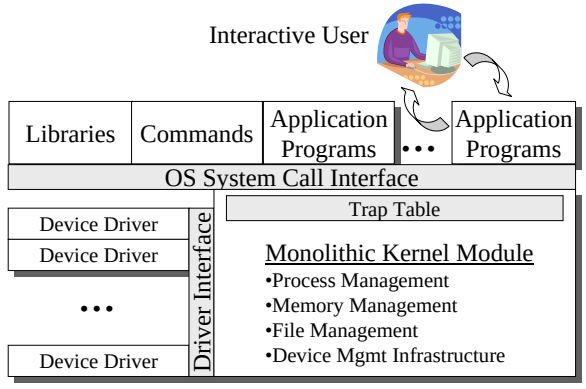
File and Secondary Storage Management

- A file is a collection of information defined by its creator. Commonly, both programs and data are stored as files.
- OS responsibility for file management:
  - Manipulation of files and directories.
  - Map files onto (nonvolatile) secondary storage - disks.
- OS responsibility for disk management:
  - Free space management and storage allocation.
  - Disk scheduling.
- They are not all always together
  - Not all files are mapped to secondary storage!
  - Not all disk space are used for the file system!

Networking (Distributed Systems)

- A *distributed* system
  - A collection of processors that do not share memory.
  - Processors are connected through a communication network.
  - Communication takes place using a *protocol*.
- A distributed system provides user access to various shared system resources, which allows:
  - Computation speed-up
  - Enhanced reliability
- How much of it is in OS?

OS Architecture: Monolithic Structure



Most modern OSes fall into this category!

Microkernel System Architecture

- Microkernel architecture:
  - Moves as much from the kernel into "*user*" space.
  - Communication takes place between user modules using message passing.
- What must be in the kernel and what can be in user space?
  - Mechanisms determine how to do something.
  - Policies decide what will be done.
- Benefits:
  - More reliable (less code is running in kernel mode)
  - More secure (less code is running in kernel mode)
- Disadvantage?

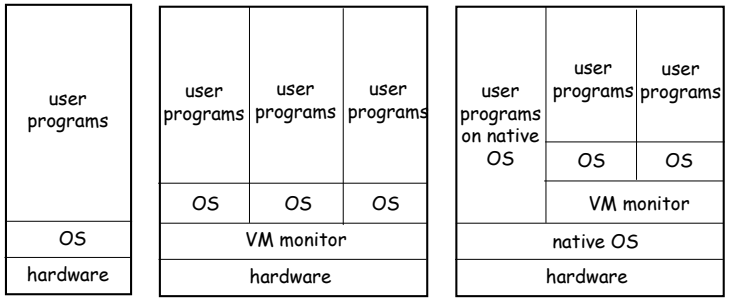
Layered Structure

- Layered structure
  - The operating system is divided into a number of layers (levels), each built on top of lower layers.
  - The bottom layer (layer 0), is the hardware.
  - The highest (layer N) is the user interface.
  - Decreased privileges for higher layers.
- Benefits:
  - more reliable
  - more secure
  - more flexibility, easier to extend
- Disadvantage?
  - Weak integration results in performance penalty (similar to the microkernel structure).

Virtual Machines

- A virtual machine
  - Virtualization: It is a piece of software that provides an interface *identical* to the underlying bare hardware.
  - Multiplexing: It provides several copies of this interface on top of a single piece of hardware.
  - Resource management: The resources of the physical computer are shared to create the virtual machines.
    - e.g., CPU scheduling can create the appearance that each piece of upper-layer software has its own processor.
- It is not an operating system in the strict sense
  - It is a resource manager, but not an extended machine

VM Models



Usage of Virtual Machines

- Running several different OSes on a single machine
- Education
- Research and development of operating systems and computer architecture
- Enhanced reliability and security

Disclaimer

- Parts of the lecture slides contain original work of Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Andrew S. Tanenbaum, and Gary Nutt. The slides are intended for the sole purpose of instruction of operating systems at the University of Rochester. All copyrighted materials belong to their original owner(s).