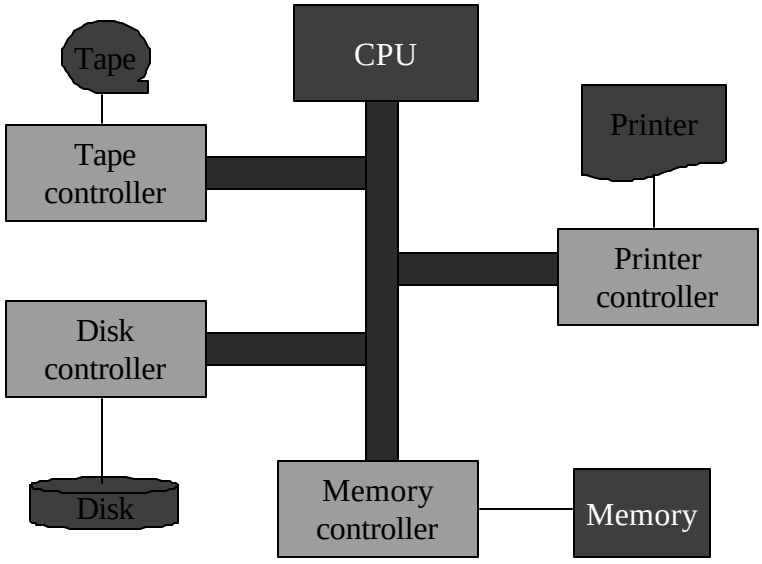
CSC 453 Operating Systems

Lecture 2: Computer-System Structures

A Modern Computer System



“Booting” The Computer

- Before the computer can start running application programs, it needs to load the operating system. This process is called “***booting***” the computer. The program that boots the computer is called a ***bootstrap*** program.

The Bootstrap Process

- Bootstrap process includes:
  - Initializing registers, device controller, memory
  - Loading the operating system kernel into memory
- The operating system must start the execution of the first process and wait for an event to occur.

An Operating System Is "Event-Driven"

- Events include:
  - The completion of a disk operation
  - A tick of the system clock
  - Request of an application program for services
  - Division by zero
  - A memory violation
- Each of these events requires a response from the operating system.
- Each of these events generates an interrupt.

Interrupts and System Calls

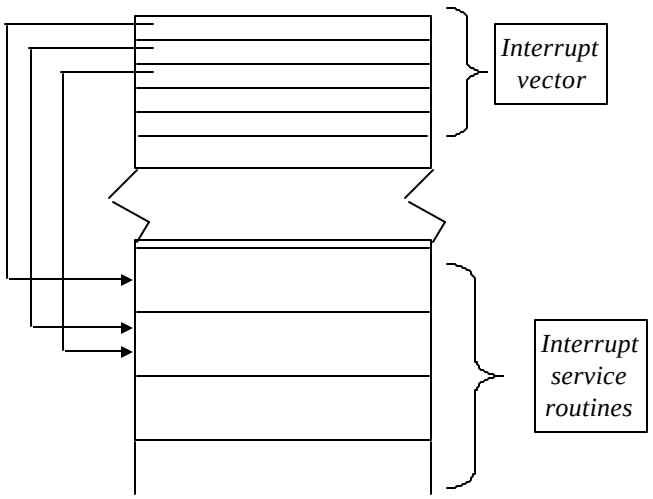
- The computer's hardware will trigger interrupts at any time when they may need service by the operating system.
- Application programs will trigger interrupts by using a hardware instruction called a system call, eg., **`int`** in Intel assembly language.

Processing Interrupts

Processing an interrupt requires transferring control to the appropriate service routine. This transfer is usually performed in several steps:

1. The CPU stops what its doing.
2. It transfers control to the appropriate service routine.
3. The service routine completes its work and the application program regains control of the CPU.

The Interrupt Vector



The Details Of Handling Interrupts

- The interrupt handler must save the address of the interrupted instruction.
  - After the interrupt is processed, that job will continue where it is interrupted.
- The interrupt handler must ***disabled*** other interrupts while the interrupt is being processed.
  - This avoids ***lost interrupts.***
  - This delays processing of other interrupts.

Disk Controllers

- Each device controller controls a particular *type* of device.
- Each controller may have more than one attached device.
  - SCSI controllers can have up to 7 devices.
- A device controller maintains a memory buffer and a set of special purpose registers.
  - Buffer size depends on the device
  - Register set varies from one controller type to another.

Starting An I/O Operation

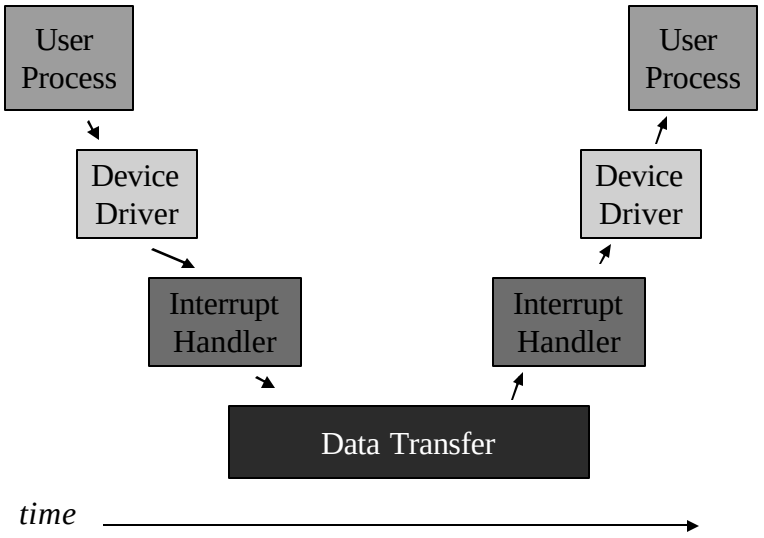
- The CPU starts an I/O operation by loading the device controller's registers with the appropriate values.
- The contents of these registers determines the action that will be taken by the controller.
- If it indicates a read operation, the controller will transfer data from the disk into its local buffer and signal the CPU when it has finished.

Synchronous vs. Asynchronous I/O

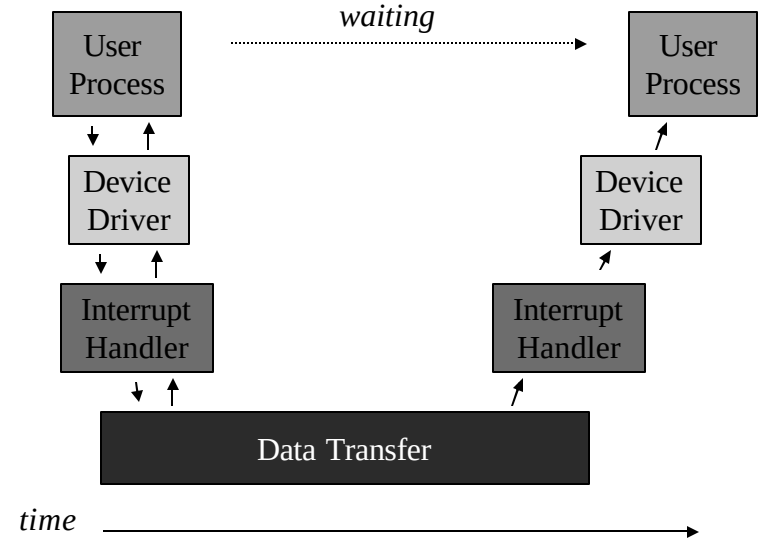
Once the I/O request is scheduled, there are two things the operating system can do:

- Wait until the operation is over – then return the CPU to the job that requested the operation. This is ***synchronous I/O***.
- Put that job on hold and work on something else. This is ***asynchronous I/O***.

Synchronous I/O



Asynchronous I/O



Direct Memory Access

- If the operating system had to be involved in reading every byte of input, there would be no time for running programs.
- To avoid delays, high-speed devices (such as disk drives) can transfer an entire block of data into memory buffers without the CPU's intervention. This is called ***direct memory access***.

Why External Storage?

- Memory is ***much*** more expensive.
- Computers never have enough memory to store all its programs and data permanently.
- Memory is volatile.
- Disk and tape storage are much less expensive and convenient for data and programs that are not in constant use.

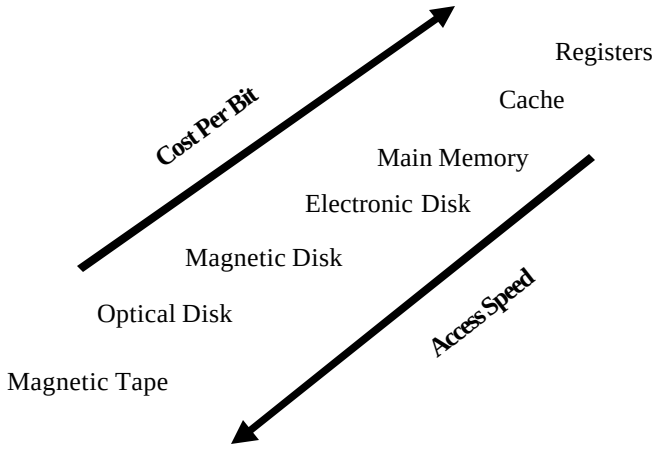
Memory-Mapped I/O

- To allow quick and more convenient access to I/O devices with fast response time, some computers set aside ranges of memory addresses for that are mapped to the device registers. These device registers can be accessed in the same manner as regular memory.
- This is called ***memory-mapped I/O***.
- The IBM PC uses this for the video display and for serial and parallel ports.

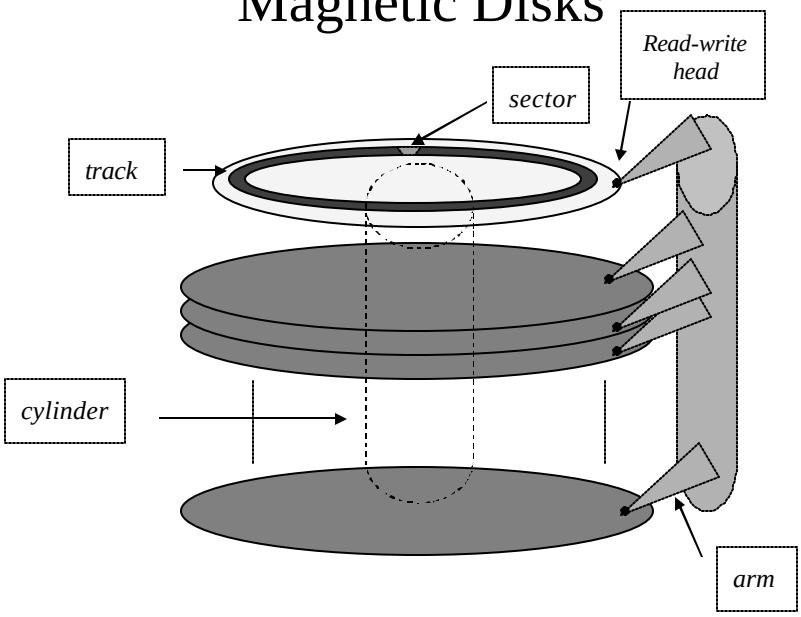
Programmed I/O vs. Interrupt-Driven I/O

- If the computer is transmitting data through a port one byte at a time, it must wait for the status bit in the devices control register to change.
- If the CPU constantly checks the register waiting for it to change, this is called ***polling*** and the CPU is using ***programmed I/O***.
- If the CPU works on other tasks while awaiting an interrupt from the device's control register, this is called ***interrupt-driven I/O***.

Storage Hierarchy



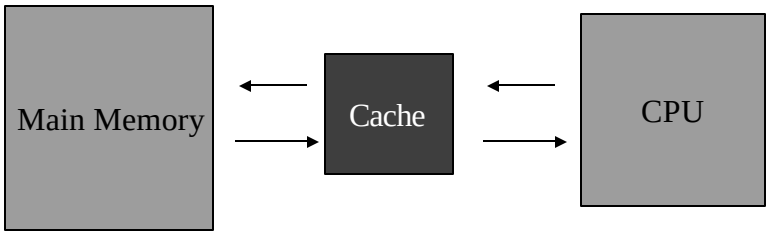
Magnetic Disks



Magnetic Tapes

- Magnetic tape was an early form of secondary storage.
- Accessing data stored on tape is much slower than accessing data stored on disk.
- Tape drives are sequential access devices while disk drives are direct-access devices.
- Tape is mainly used for archiving and retrieving rarely-used data.

Caching



Coherency and Consistency

- Imagine reading a file so that you could update one record. You have to:
  1. read the record
  2. change the value in memory
  3. rewrite it on disk.
- Between steps 2 and 3, the value is different in memory and on disk. We need to save this new value of the record to ensure ***coherency***..

The Need For Protection

- Early computers were used by one users at a time.
- As operating systems developed, certain functions were assigned to the operating system.
- As operating systems started to use multiprograming to run multiple jobs concurrently, the operating systems needed to protect user programs from potential errors in other programs.

Dual-Mode Operation

- Computers need to differentiate between user operations and systems operations.
- The mode bit indicates if the computer is running in *user mode* or *supervisor mode*.
- The computer is booted in supervisor mode and switches to user mode as it loads the first user job.

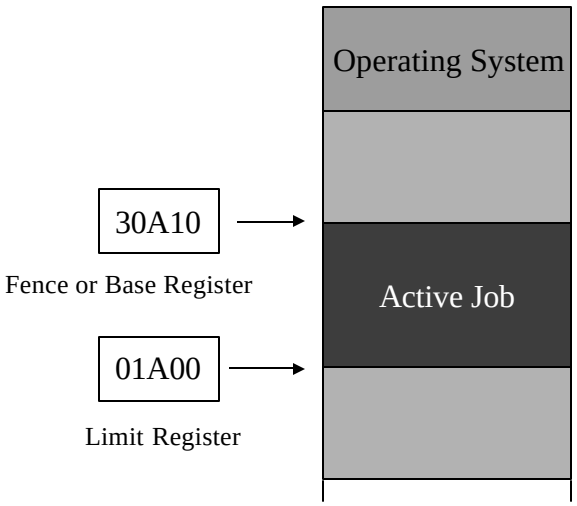
Privileged Instructions

- Certain ***privileged instructions*** can be user only in supervisor mode.
- These instructions restrict access to most hardware devices to supervisor mode.
- This requires operating system to perform input/output and other tasks.

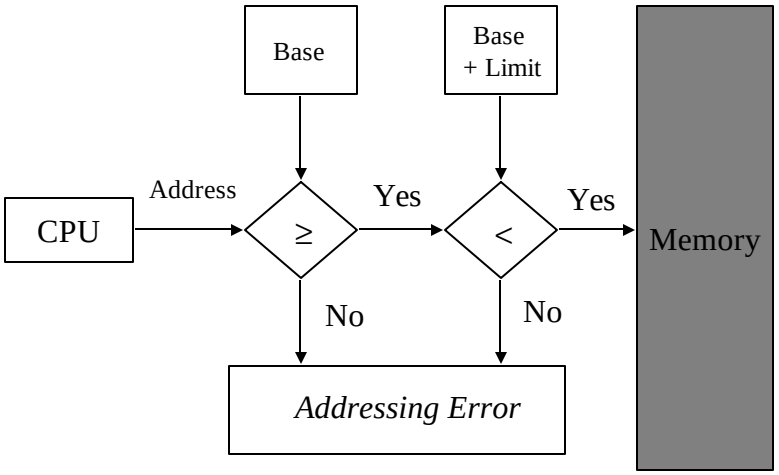
What if There is No Dual Mode?

- The Intel 8088 processor was designed without a mode bit.
  - User programs can overwrite the operating system.
  - Multiple programs can write to the same device at the same time.
- This makes it virtually impossible to implement multitasking operating systems and it allows the creation of computer viruses.
- Intel corrected this shortcoming by providing dual mode operation in all its processors since the 286.

Memory Protection



Hardware Address Protection



CPU Protection

- What do we do if a program falls into an infinite loop? ***A timer!!***
- A fixed-rate clock ticks a predetermined number of times per second. The operating systems sets a counter to track clock ticks.
- A program is allowed to run for a fixed time period before it must surrender the CPU.

Time Slices

- In interactive operating systems, each terminal is allowed only a fixed amount of CPU time before the CPU is given to another terminal. The fixed time period is called a ***time slice***.
- The operating system takes over the CPU at the end of every time slice and performs housekeeping chores before giving the CPU to the next job.

I/O and System Calls

```
mov   ah, 9          ; set the high end of ax to 9
mov   dx, offset message
                     ; move message's offset to dx
int   21h            ; call interrupt 21h
```