Input / Output

CSE 410, Spring 2004

Computer Systems

http://www.cs.washington.edu/education/courses/410/04sp/

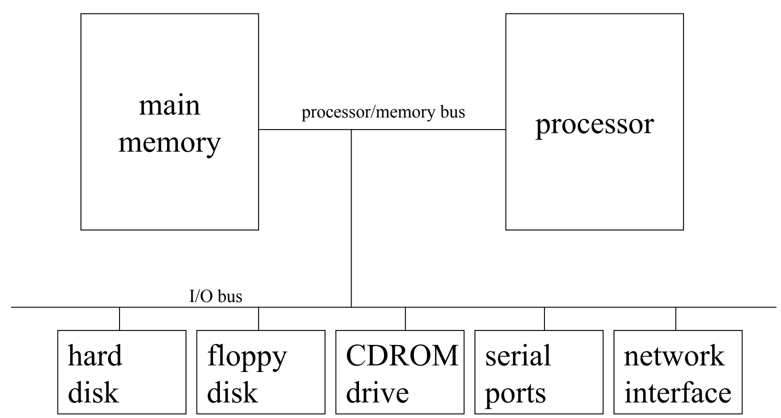
Reading and References

- Reading
  - » Section 8.1-8.5, *Computer Organization and Design, Patterson and Hennessy*

Why Input and Output?

- Everything we have done so far is based on moving data / instructions between main memory and the CPU
- How does the information get into main memory and out to the user?
  - » during manufacture: burn it in
  - » during operation: input / output

A typical organization



Types of I/O devices

- Behavior
  - » input only (keyboard, mouse, sensor)
  - » output only (monitor, LED display, actuator)
  - » input and output (network, disk, tape, CD-RW)
- Partner
  - » human or machine
- Data rate
  - » negligible to KiloBytes/Second to MegaBytes/S

Three Characteristic Devices

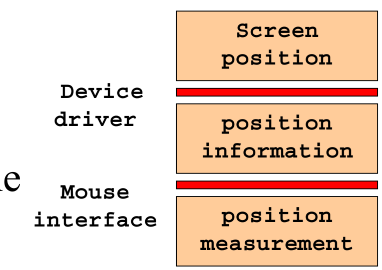
- Mouse
  - » input only; human; .01-.02 KB/s
- Magnetic disks
  - » input and output; machine; 100-10,000 KB/s
- Networks
  - » input and output; machine; 500-6000 KB/s

Detecting Mouse Motion

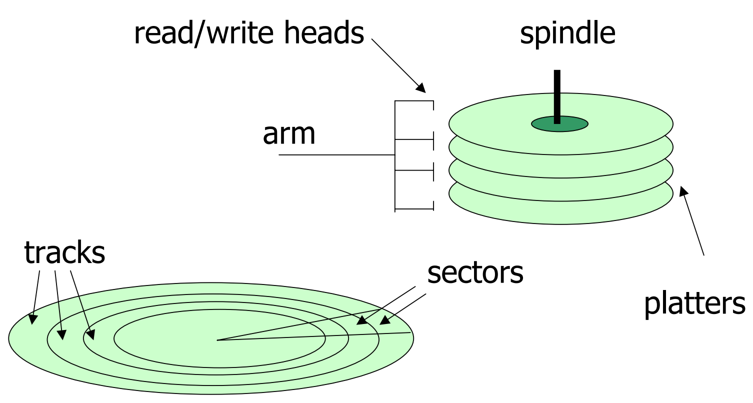
- Mechanical
  - » Two perpendicular wheels on the inside connected to potentiometers
- Optical
  - » LED and a photodetector on the bottom
- Optomechanical
  - » The wheels have slots; an LED shines through them
- Lots of others
  - » embedded microprocessor in the mouse

Mouse Interfaces

- Pointing device must provide
  - » Status of each button
  - » Position in X and Y
- Software must interpret the input
  - » Double clicks
  - » Limits to motion, speed



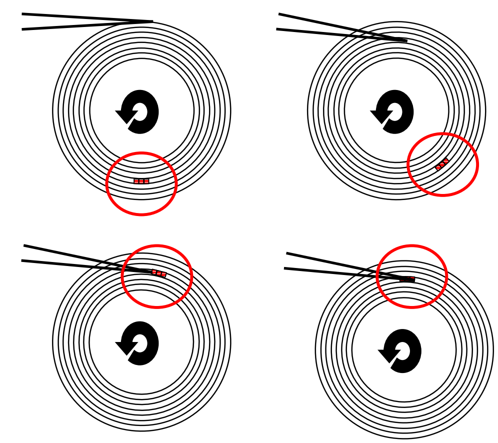
Anatomy of a Magnetic Disk



cylinder: all tracks at the same radius, one/two per platter

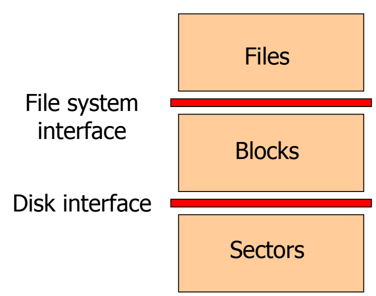
Accessing Data from a Disk

- First, the disk arm moves over the right cylinder: *seek time*
- Next, wait for the data to rotate under the disk arm: *rotational delay,* (on average half the time for one revolution)
- Finally, read the data from the disk: *transfer time*
- At this point, the data is in the *disk controller* and can be transferred to memory



Disk Interface

- Only two data transfer operations on a disk:
  - » read block, write block
- Hidden behind the interface:
  - » block ↔ sector mappings
- Unknown to the disk:
  - » contents of the blocks/sectors



Disk Specifications

- Consider the disk in my (old) laptop
  - » 5GB IBM DJSA-205 (from device manager)
- Specifications (from IBM data sheet)
  - » rotation speed 4200 RPM
    - rotational delay = 60 * 1/4200 * 0.5 = 7.1 ms
  - » seek times
    - avg: 12.0 ms, 1 track: 2.5 ms, full stroke: 23.0 ms
  - » layout
    - heads: 1, cylinders: 22784 (user), 10336 (actual)
    - sectors per track: 293-560

Data Transfer Specifications

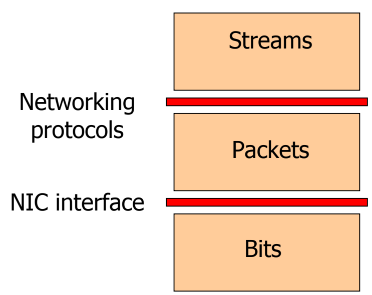
- Data buffer
  - » 512 Kbytes on board the disk
- Media Transfer rate
  - » 108.8 to 202.9 Mbits / sec
- Interface Transfer rate
  - » Ultra-DMA mode 4: 66.6 Mbytes/sec
  - » PIO mode 4: 16.6 Mbytes/sec

Networks

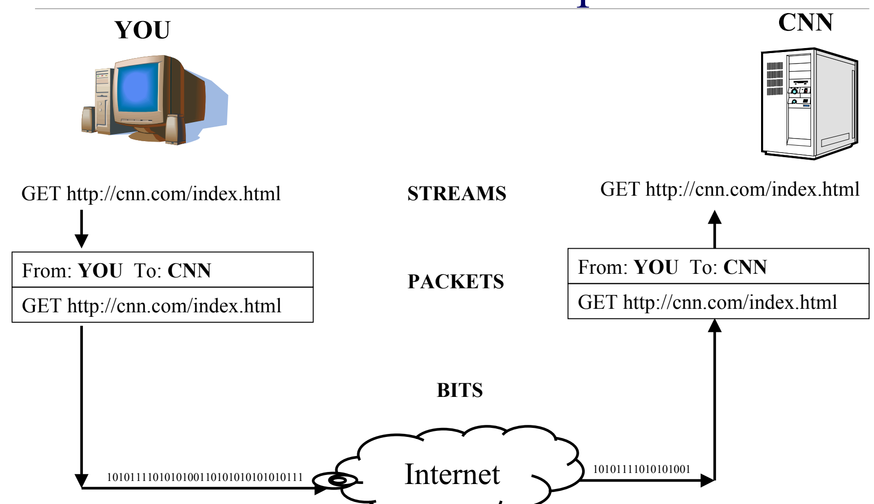
- The device that lets a system connect to a network: *network interface card*
- Listens for data on the network important to this system
- Bundles the bits into packets and signals the OS when a packet is complete
- Also takes packets from OS and sends them as bits on the wire

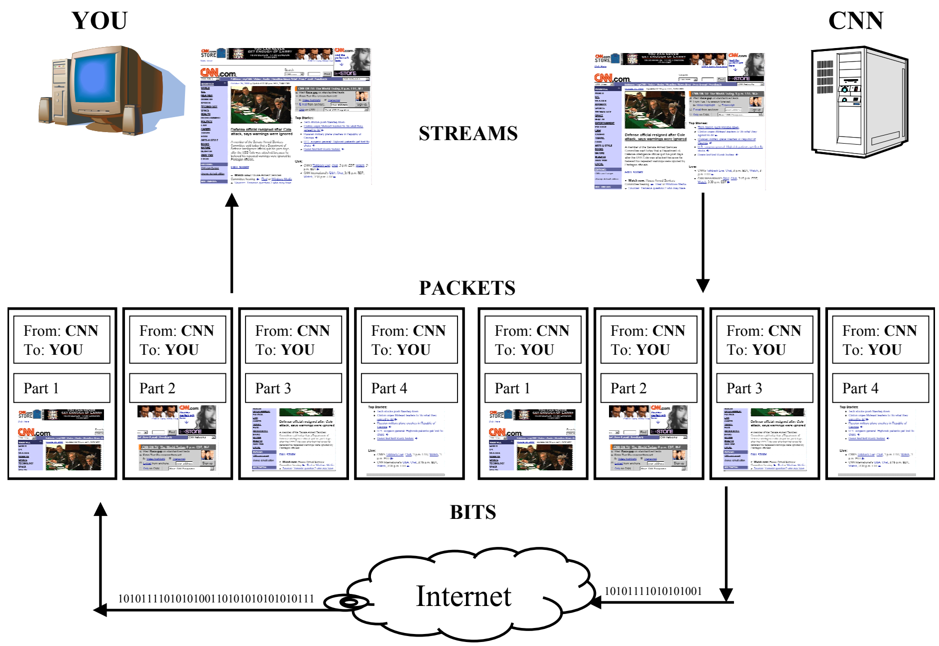
Networking Interfaces

- OS puts extra packets in to define where stream begins and ends
- NIC puts extra bits in to define where packets begin and end

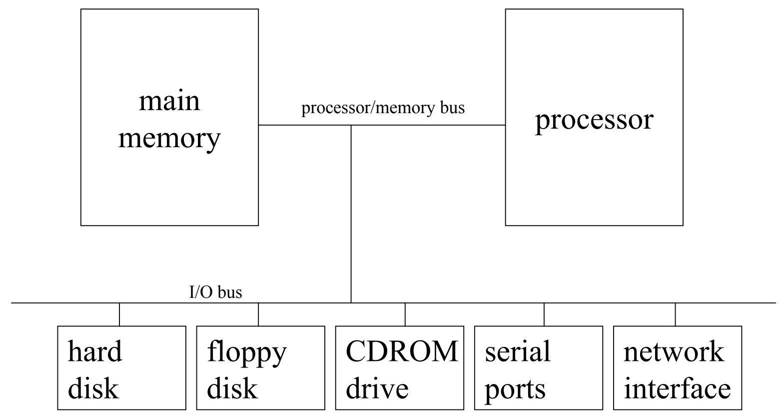


Network Example





A typical organization



Commands to I/O Devices

- Memory-mapped I/O
  - » A special region of memory is set aside for a device
  - » Loads and stores to addresses in this region are interpreted as commands to the device
  - » Provides easy access control via the memory system
- Special I/O instructions

Device to Processor

- The device has some information for the processor. Two ways to convey it:
  - » Issue an interrupt
  - » Wait for the processor to ask for it - *polling*
- Which is better, interrupt-driven I/O or polling? Depends on:
  - » time sensitivity of data
  - » whether data is expected

Device to Memory

- *Direct Memory Access (DMA)* allows devices and memory to communicate without involvement of processor
- Processor sets up the transaction
- Device and memory transfer the data
- Device interrupts processor to signal completion
- The processor gets a lot of other work done while transfer is happening

Performance Issues in I/O

- Processors double in speed every 18 months
- Networks double in speed more slowly, perhaps every 3 years
- Disks improve more slowly, because they are limited by mechanical factors
  - » however, bit density has gone up rapidly

The I/O Bottleneck

- System A
  - » processor speed = 100 MHz (clock cycle 10 ns)
  - » disk transfer takes 10 ms
  - » How many clock cycles elapse while disk transfer takes place?
- System B
  - » processor speed = 1 GHz (clock cycle 1 ns)
  - » disk transfer *still* takes 10 ms
  - » How many clock cycles now?

I/O Bus Constraints

- Two primary design points that must be met
- High speed
  - » processor / memory, bulk devices / memory
- Flexibility
  - » many types of I/O devices with widely varying characteristics
  - » characteristics of future devices are unknown at design time

Designs

- The speed and flexibility constraints lead to designs which are
  - » designed for speed
    - processor-memory bus
  - » designed for flexibility
    - I/O bus
  - » designed for both
    - backplane bus

Speed ? Synchronous Bus

- For highest speed, all devices are designed to work together at the same high rate
- Synchronous buses have a clock signal that all devices on the bus are aware of
- Protocol for accessing the bus is relatively simple
  - » control signals at specified clock cycles
  - » data at specified clock cycles

Synchronous Issues

- Runs fast
- but
  - » all attached devices must be designed for this particular (probably proprietary) bus
  - » must be short so that signals can propagate across the whole bus
  - » fast today is slow tomorrow

Flexibility ? Asynchronous Bus

- Devices access the bus by handshaking to determine who can go next
- No single clock
  - » transactions are defined by control signal transitions
- Can accommodate a wide variety of device speeds and device types

Asynchronous Issues

- Flexible
- but
  - » the handshake adds overhead to each transfer
  - » special cases pollute the protocol as it is extended to provide higher speed capabilities
  - » extreme network effect: once a bus is popular, it lives long past its expected lifetime because there are so many devices that use it

Bus Bandwidth

- Width of the bus
  - » number of data lines can be increased to transfer more bits of data in parallel
- Multiplexing
  - » data signals and control signals can be put on the same lines at different times to save hardware
  - » or separated to overlap handshake and data transfer
- Multi-word transfers
  - » block transfers move more data per handshake

Controlling bus access

- With multiple devices on the bus, something must control access
- Bus Master
  - » device that is allowed to initiate transfers
- Single bus master
  - » simple, because no contention
  - » potential bottleneck, because one device is busy for every single transfer on the bus

Multiple Masters and Arbitration

- Let several devices act as bus masters
- Must decide who is in control for any particular transaction
- Arbitration
  - » daisy chain - serial decision
  - » centralized parallel - one decider
  - » distributed parallel - many deciders
  - » distributed with collision detection