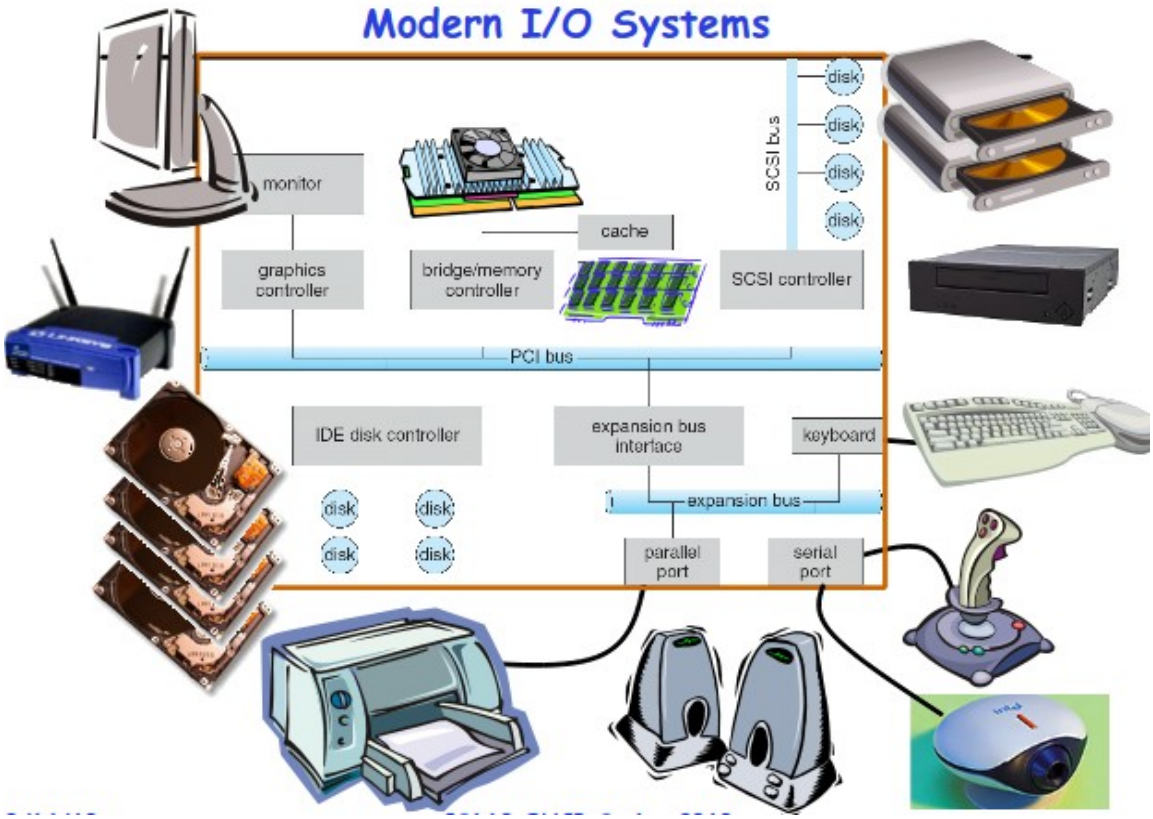
INPUT AND OUTPUT SYSTEMS



What about I/O?

- Without I/O, computers are useless (disembodied brains?)
- But… thousands of devices, each slightly different
- How can we standardize the interfaces to these devices?
- Devices unreliable: media failures and transmission errors
- How can we make them reliable???
- Devices unpredictable and/or slow
- How can we manage them if we don't know what they will do or how they will perform?

Some operational parameters:

Byte/Block
Some devices provide single byte at a time (*e.g.* keyboard)
Others provide whole blocks (*e.g.* disks, networks, etc)
Sequential/Random
Some devices must be accessed sequentially (*e.g.* tape)
Others can be accessed randomly (*e.g.* disk, cd, etc.)

Polling/Interrupts
Some devices require continual monitoring
Others generate interrupts when they need service

I/O devices can be roughly divided into two categories: **block devices and character devices**. A block device is one that stores information in fixed-size blocks, each one with its own address. Common block sizes range from 512 bytes to 32,768 bytes. The essential property of a block device is that it is possible to read or write each block independently of all the other ones. Disks are the most common block devices.

The other type of I/O device is the **character device.** A character device delivers or accepts a stream of characters, without regard to any block structure. It is not addressable and does not have any seek operation. Printers, network interfaces, mice (for pointing), rats (for psychology lab experiments), and most other devices that are not disk-like can be seen as character devices.

Block Devices: e.g. disk drives, tape drives, DVD-ROM
Access blocks of data
Commands include open() , read() , write() , seek()
Raw I/O or file-system access
Memory-mapped file access possible

Character Devices: e.g. keyboards, mice, serial ports, some USB devices
Single characters at a time
Commands include get() , put()
Libraries layered on top allow line editing

Network Devices: e.g. Ethernet, Wireless, Bluetooth
Different enough from block/character to have own interface
Unix and Windows include socket interface
Separates network protocol from network operation
Includes select() functionality
Usage: pipes, FIFOs, streams, queues, mailboxes