OPERATING SYSTEMS

I/O SYSTEM

Categories of I/O Devices

- External devices that engage in I/O with computer systems can be grouped into three categories

Human readable

- Suitable for communicating with the computer user
- Printers, terminals, video display, keyboard, mouse

Machine readable

- Suitable for communicating with electronic equipment
- Disk drives, USB keys, sensors, controllers

Communication

- Suitable for communicating with remote devices
- Modems, digital line drivers

Differences in I/O Devices

Data Rate

There may be **differences of magnitude** between the data transfer rates

Application

The use to which a device is put has an **influence on the software**

Complexity of Control

The effect on the operating system is filtered by the **complexity of the I/O module** that controls the device

Unit of Transfer

Data may be transferred as a **stream** of bytes or **characters** or in larger **blocks**

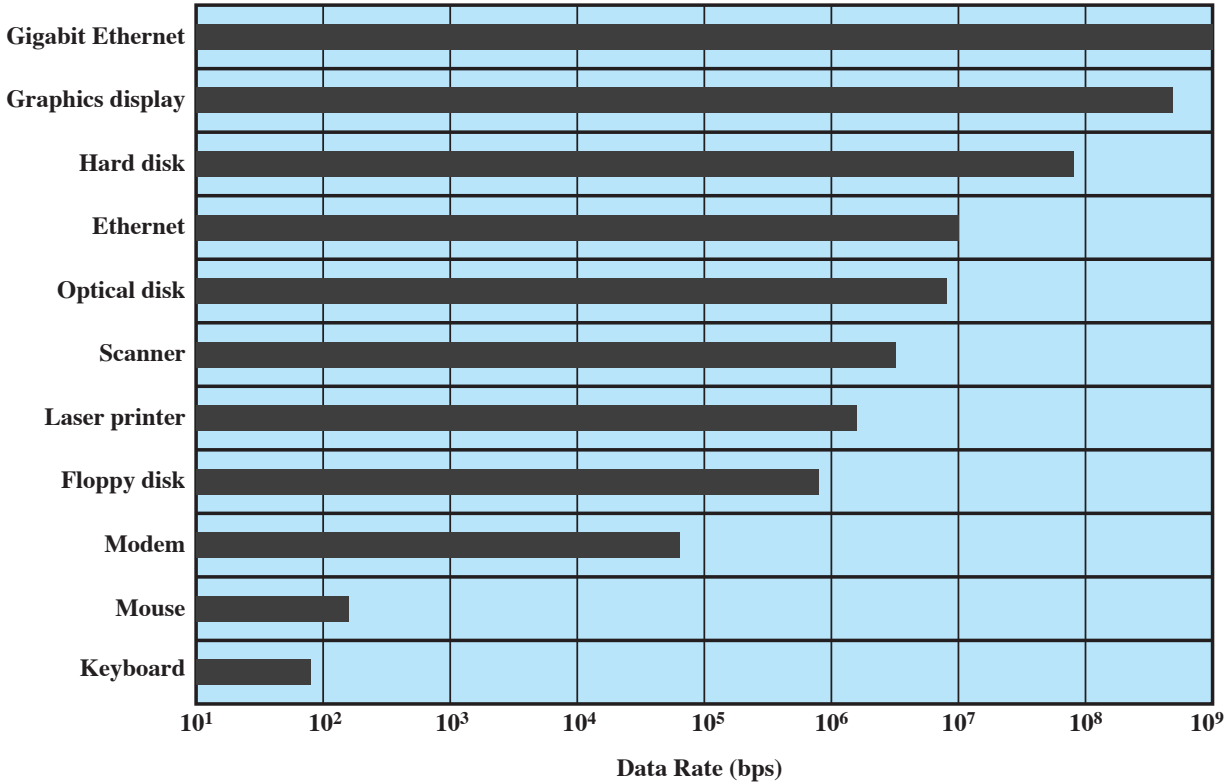
Data Representation

Different **data encoding** schemes are used by different devices

Error Conditions

The **nature** of errors, the way in which they are **reported**, their **consequences**, and the available range of **responses** differs from one device to another

Typical I/O Device Data Rates



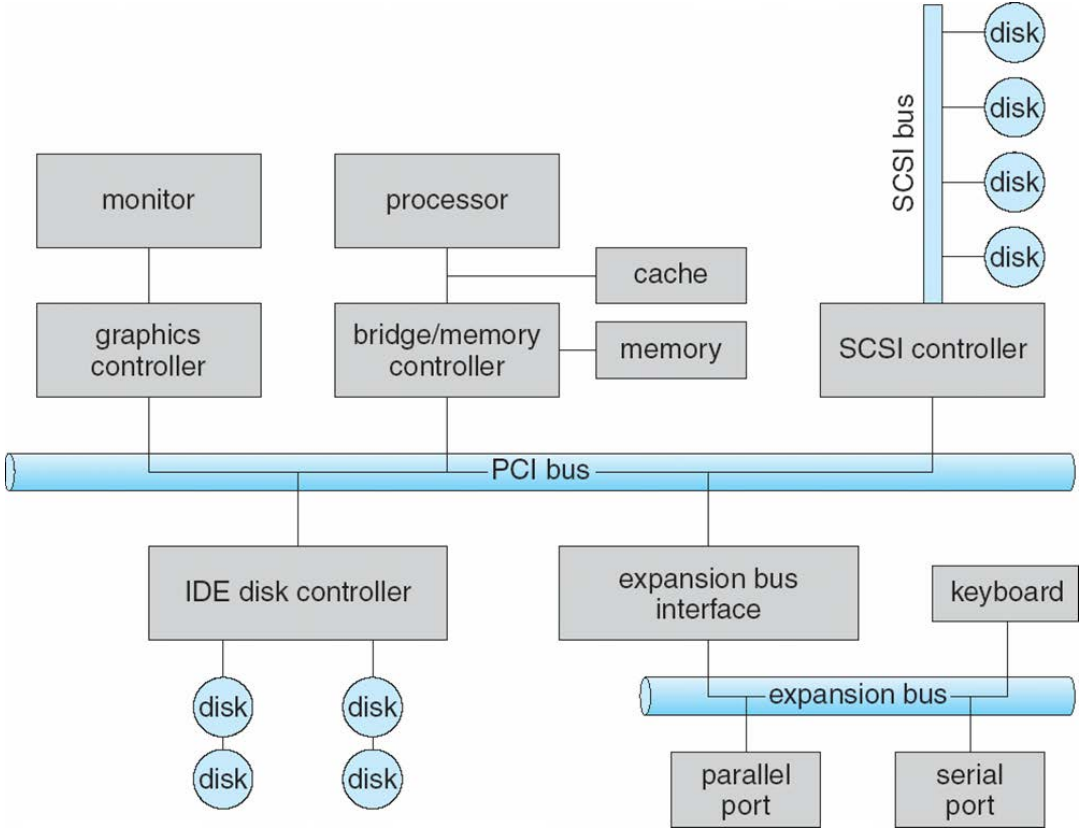
Nonblocking and Asynchronous I/O

- **Blocking**
  process suspended until I/O completed
  - Easy to use and understand
  - Insufficient for some needs
- **Nonblocking**
  I/O call returns as much as available
  - User interface, data copy
  - Implemented via multi-threading
  - Returns quickly with count of bytes read or written
- **Asynchronous**
  process runs while I/O executes
  - Difficult to use
  - I/O subsystem signals process when I/O completed

Characteristics of I/O Devices

aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read–write	CD-ROM graphics controller disk

A Typical PC Bus Structure



Organization of the I/O Function

- Three techniques for performing I/O
- **Programmed I/O**
  - The processor issues an I/O command on behalf of a process to an I/O module
  - busy waiting
- **Interrupt-driven I/O**
  - The processor issues an I/O command on behalf of a process
    - If non-blocking – processor continues to execute instructions
    - If blocking – the next instruction the processor executes is from the OS, which will put the current process in a blocked state and schedule another process
- **Direct Memory Access (DMA)**
  - After the interrupt, a DMA module controls the exchange of data between main memory and an I/O module

Evolution of the I/O Function

1. Processor directly controls a peripheral device
2. A controller or I/O module is added
3. Same configuration as step 2, but now interrupts are employed
4. The I/O module is given direct control of memory via DMA
5. The I/O module is enhanced to become a separate processor, with a specialized instruction set tailored for I/O
6. The I/O module has a local memory of its own and is a computer in its own right

I/O Hardware

- Devices usually have registers where device driver places commands, addresses, and data to write, or read data from registers after command execution
  - Data-in register, data-out register, status register, control register
  - Typically 1-4 bytes, or FIFO buffer
- Devices have addresses, used by
  - Direct I/O instructions
  - Memory-mapped I/O
    - Device data and command registers mapped to processor address space
    - Especially for large address spaces (graphics)

Design Objectives

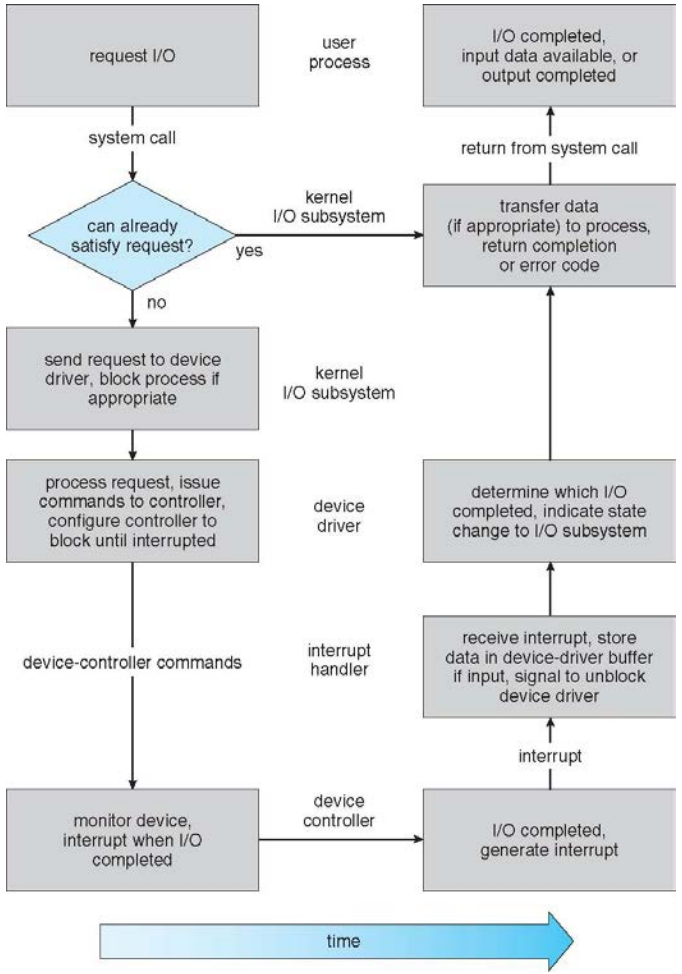
Efficiency

- Major effort in I/O design
- Important because I/O operations often form a bottleneck
- Most I/O devices are extremely slow compared with main memory and the processor
- The area that has received the most attention is disk I/O

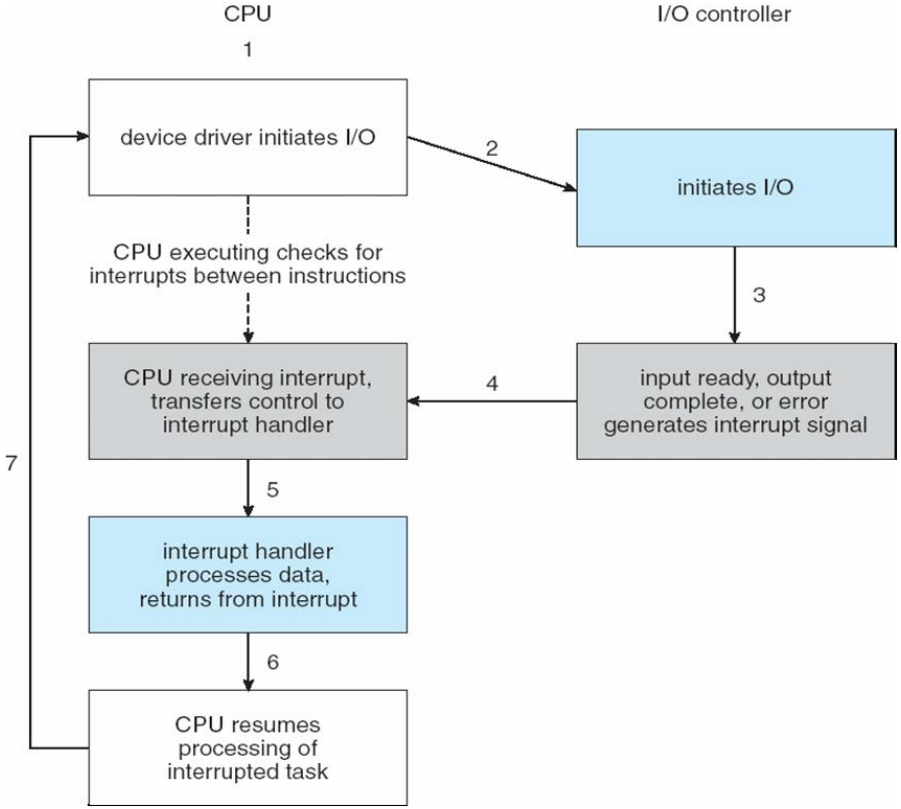
Generality

- Desirable to handle all devices in a uniform manner
- Applies to the way processes view I/O devices and the way the operating system manages I/O devices and operations
- Diversity of devices makes it difficult to achieve true generality
- Use a hierarchical, modular approach to the design of the I/O function

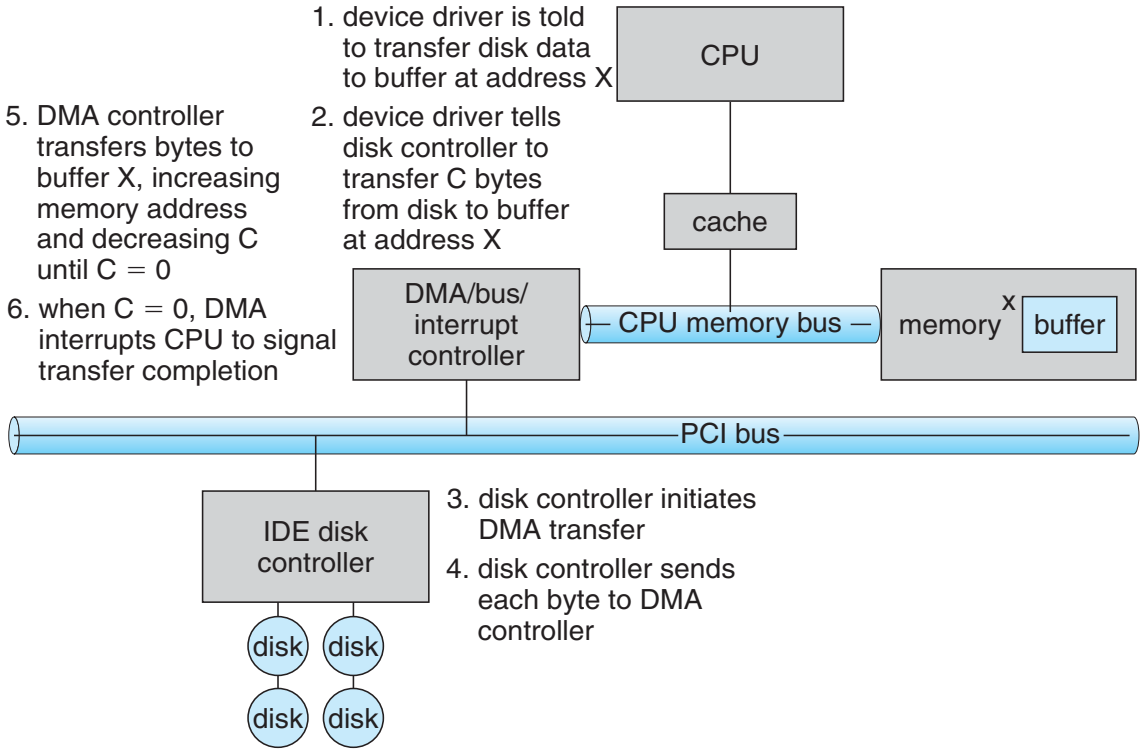
Life Cycle of an I/O Request



Interrupt-Driven I/O Cycle



DMA Transfers



Kernel I/O Subsystem

- **Scheduling**
  - Some I/O request ordering via per-device queue
  - Some OSs try fairness
  - Some implement Quality Of Service
- **Buffering**
  store data in memory while transferring between devices
  - To cope with device speed mismatch
  - To cope with device transfer size mismatch
  - To maintain *copy semantics*

Kernel I/O Subsystem

- **Caching**
  faster device holding copy of data
  - Always just a copy
  - Key to performance
  - Sometimes combined with buffering
- **Spooling**
  hold output for a device
  - If device can serve only one request at a time (i.e., printing)
- **Device reservation**
  provides exclusive access to a device
  - System calls for allocation and de-allocation
  - Watch out for deadlock

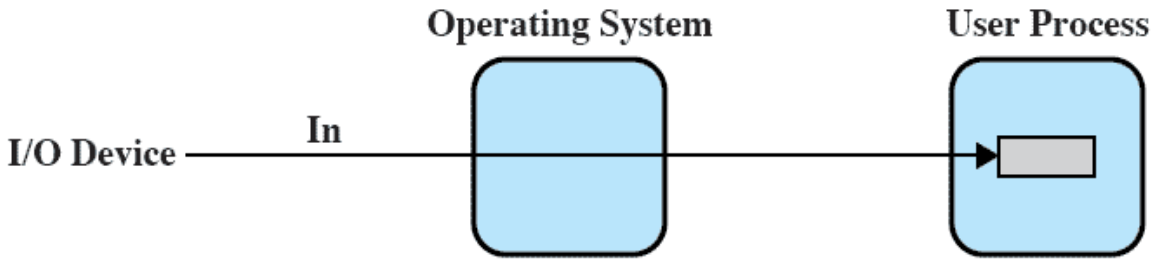
Buffering

- To avoid overheads and inefficiencies, it is sometimes convenient to perform input transfers in advance of requests being made, and to perform output transfers some time after the request is made

Block-oriented device	Stream-oriented device
• Stores information in blocks that are usually of fixed size • Transfers are made one block at a time • Possible to reference data by its block number • Disks and USB keys are examples	• Transfers data in and out as a stream of bytes • No block structure • Terminals, printers, communications ports, mouse and other pointing devices, and most other devices that are not secondary storage are examples

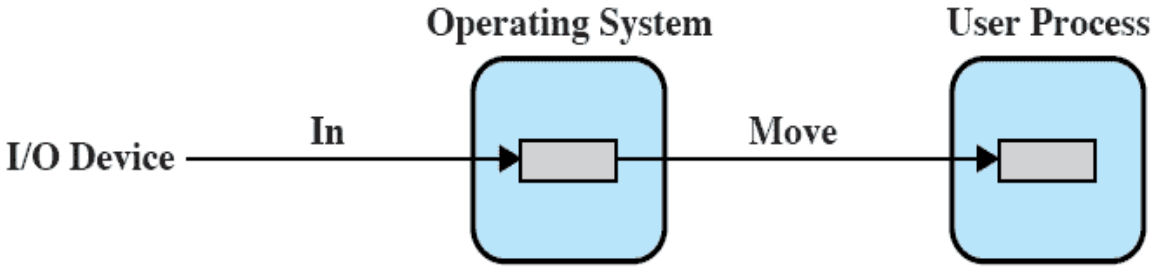
No Buffer

- Without a buffer, the OS directly accesses the device when it needs



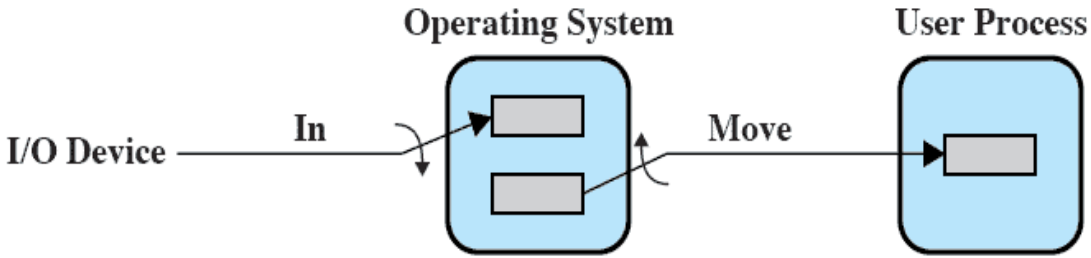
Single Buffer

- The simplest type of support that the operating system can provide
- When a user process issues an I/O request, the OS assigns a buffer in the system portion of main memory to the operation



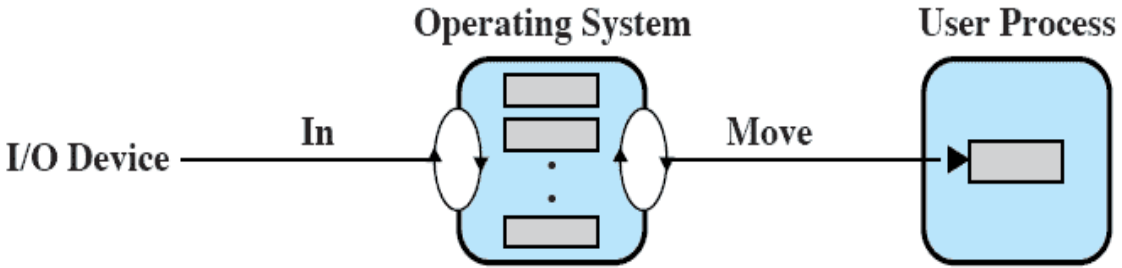
Double Buffer

- Assigning two system buffers to the operation
- A process now transfers data to or from one buffer while the operating system empties or fills the other buffer
- Also known as buffer swapping



Circular Buffer

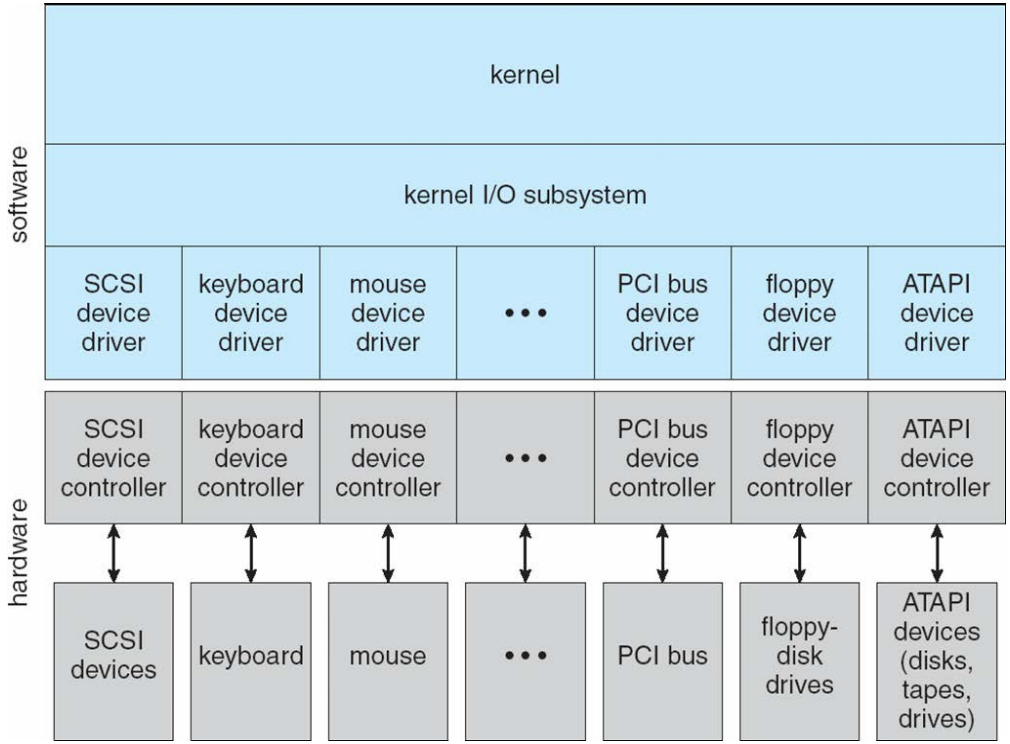
- When more than two buffers are used, the collection of buffers is itself referred to as a circular buffer
- Each individual buffer is one unit in the circular buffer



Application I/O Interface

- **I/O system calls** encapsulate device behaviors in generic classes
- **Device-driver** layer hides differences among I/O controllers from kernel
- New devices talking already-implemented protocols need no extra work
- Each OS has its own I/O subsystem structures and device driver frameworks

A Kernel I/O Structure



Characteristics of I/O Devices (Cont.)

- Subtleties of devices handled by device drivers
- Broadly I/O devices can be grouped by the OS into
  - Block I/O
  - Character I/O (Stream)
  - Memory-mapped file access
  - Network sockets
- For direct manipulation of I/O device specific characteristics, usually an escape / back door
  - Unix ioctl() call to send arbitrary bits to a device control register and data to device data register

Device-Functionality Progression