I/O Management and Disk Scheduling

Categories of I/O Devices

- Difficult area of OS design
  - Difficult to develop a consistent solution due to a wide variety of devices and applications
- Three Categories:
  - Human readable
  - Machine readable
  - Communications

Human readable

- Devices used to communicate with the user
- Printers and terminals
  - Video display
  - Keyboard
  - Mouse etc

Machine readable

- Used to communicate with electronic equipment
  - Disk drives
  - USB keys
  - Sensors
  - Controllers
  - Actuators

Communication

- Used to communicate with remote devices
  - Digital line drivers
  - Modems

Differences in I/O Devices

- Devices differ in a number of areas
  - Data Rate
  - Application
  - Complexity of Control
  - Unit of Transfer
  - Data Representation
  - Error Conditions

Data Rate

- May be massive difference between the data transfer rates of devices

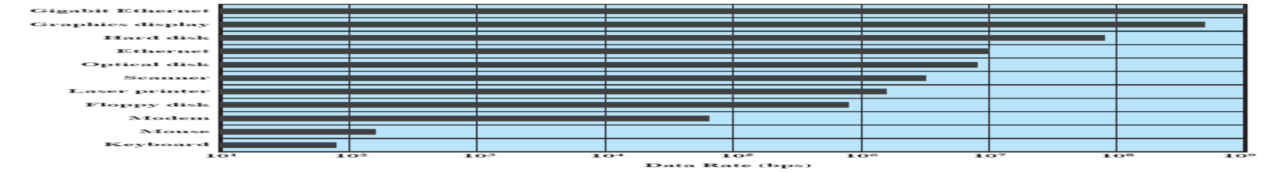

Figure 11.1 Typical I/O Device Data Rates

Application

- The use to which a device is put has an influence on the software and policies in the O.S. and supporting utilities.
  - Disk used to store files requires file management software
  - Disk used to store virtual memory pages needs special hardware and software to support it
  - Terminal used by system administrator may have a higher priority

Complexity of control

- A printer requires a relatively simple control interface.
- A disk is much more complex.
- This complexity is filtered to some extent 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 (e.g., terminal I/O)
  - or in larger blocks (e.g., disk I/O).

Data representation

- Different data encoding schemes are used by different devices,
  - including differences in character code and parity conventions.

Error Conditions

- The nature of errors differ widely from one device to another.
- Aspects include:
  - the way in which they are reported,
  - their consequences,
  - the available range of responses

Organization of the I/O Function

Techniques for performing I/O

- Programmed I/O
- Interrupt-driven I/O
- Direct memory access (DMA)

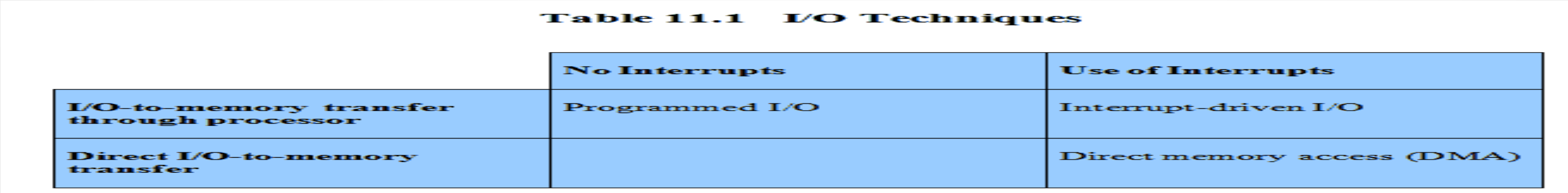

Table 11.1 I/O Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

Programmed I/O

- The processor issues an I/O command on behalf of a process to an I/O module;
- that process then busy-waits for the operation to be complete before proceeding.

Interrupt-driven I/O

- The processor issues an I/O command on behalf of a process.
  - if non-blocking – processor continues to execute instructions from the process that issued the I/O command
  - 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)

- 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. Controller or I/O module is added
   - Processor uses programmed I/O without interrupts
   - Processor does not need to handle details of external devices
3. Controller or I/O module with interrupts
   - Efficiency improves as processor does not spendtime waiting for an I/O operation to be performed

Evolution of the I/O Function

- 4. Direct Memory Access
  - Blocks of data are moved into memory without involving the processor
  - Processor involved at beginning and end only

5. I/O module is enhanced as a separate processor

CPU directs the I/O processor to execute an I/O program in main memory.

6. I/O processor

– I/O module has its own local memory

– Commonly used to control communications with interactive terminals