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

- Devices differ in a number of areas:
  - 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

Figure 11.1 Typical I/O Device Data Rates
Three techniques for performing I/O are:

1. **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 completed before proceeding.

2. **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.

3. **Direct Memory Access (DMA)**
   - A DMA module controls the exchange of data between main memory and an I/O module.

---

**Organization of the I/O Function**

**Evolution of the I/O Function**

---

**Figure 11.2** Typical DMA Block Diagram

**Figure 11.3** Alternative DMA Configurations
**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

**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

**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

![Diagram of Single Buffering](image)

**Disadvantages:**
- Complicates the logic in the operating system
- Swapping logic is also affected

**Single Buffering for Stream-Oriented I/O**

- Can be used in a line-at-a-time fashion or a byte-at-a-time fashion
  - Line-at-a-time operation is appropriate for scroll-mode terminals (dumb terminals)
  - With this form of terminal, user input is one line at a time, with a carriage return signaling the end of a line
  - Output to the terminal is similarly one line at a time

- Byte-at-a-time operation is used on forms-mode terminals,
  - When each keystroke is significant and for many other peripherals, such as sensors and controllers

**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

![Diagram of Double Buffering](image)
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.

![Circular Buffer Diagram](image)

Disk Performance Parameters

- The actual details of disk I/O operation depend on the:
  - Computer system
  - Operating system
  - Nature of the I/O channel and disk controller hardware

<table>
<thead>
<tr>
<th>Wait for Device</th>
<th>Wait for Channel</th>
<th>Seek</th>
<th>Rotational Delay</th>
<th>Data Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 11.6 Timing of a Disk I/O Transfer**

Disk Performance Parameters

- When the disk drive is operating, the disk is rotating at constant speed.
- To read or write, the head must be positioned at the desired track and at the beginning of the desired sector on that track.
- Track selection involves moving the head in a movable-head system or electronically selecting one head on a fixed-head system.
- On a movable-head system, the time it takes to position the head at the track is known as seek time.
- The time it takes for the beginning of the sector to reach the head is known as rotational delay.
- The sum of the seek time and the rotational delay equals the access time.
Seek Time

- The time required to move the disk arm to the required track
- Consists of two key components:
  - The initial startup time
  - The time taken to traverse the tracks that have to be crossed once the access arm is up to speed
- Settling time
  - Time after positioning the head over the target track until track identification is confirmed
- Much improvement comes from smaller and lighter disk components
  - A typical average seek time on contemporary hard disks is under 10ms

Disk Performance

- Rotational delay
  - The time required for the addressed area of the disk to rotate into a position where it is accessible by the read/write head
  - Disks rotate at speeds ranging from 3,600 rpm (for handheld devices such as digital cameras) up to 15,000 rpm

First-In, First-Out (FIFO)

- Processes in sequential order
- Fair to all processes
- Approximates random scheduling in performance if there are many processes competing for the disk

Priority (PRI)

- Control of the scheduling is outside the control of disk management software
- Goal is not to optimize disk utilization but to meet other objectives
- Short batch jobs and interactive jobs are given higher priority
- Provides good interactive response time
- Longer jobs may have to wait an excessively long time
- A poor policy for database systems
Select the disk I/O request that requires the least movement of the disk arm from its current position
Always choose the minimum seek time

Also known as the elevator algorithm
Arm moves in one direction only
Satisfies all outstanding requests until it reaches the last track in that direction then the direction is reversed
Favors jobs whose requests are for tracks nearest to both innermost and outermost tracks and favors the latest-arriving jobs

Restricts scanning to one direction only
When the last track has been visited in one direction, the arm is returned to the opposite end of the disk and the scan begins again

Segments the disk request queue into subqueues of length N
Subqueues are processed one at a time, using SCAN
While a queue is being processed new requests must be added to some other queue
If fewer than N requests are available at the end of a scan, all of them are processed with the next scan
FSCAN

- Uses two subqueues
- When a scan begins, all of the requests are in one of the queues, with the other empty
- During scan, all new requests are put into the other queue
- Service of new requests is deferred until all of the old requests have been processed

© 2017 Pearson Education, Inc., Hoboken, N.J. All rights reserved.