Disk Drives

Data is stored on a disk as **magnetic flux transitions in a thin film of magnetic material on the surface of the disk**. The disk itself is a circle of some material (**thin plastic in the case of floppy disks, metal in the case of hard disks**), which is spun on its central axis by a motor. A small coil in the head assembly senses the flux transitions as the disk surface passes under it. This signal is then amplified and processed, and a stream of bits results.

The disk drive head assembly contains a **separate head for each disk surface** that can contain data. **Floppy disks were originally single sided**, which means that data was contained on only one side of the disk material. **Current floppy disk drives are all double sided**, meaning that there is a separate head for each side of the disk, and data is stored on both sides of the disk. **Hard drives typically contain several platters** (the name for an individual disk within the stack), and **have a separate head for each side of each platter**.

The head assembly can be moved in or out across the disk surface in **discrete steps called “seeks”**. The circle on the disk passing under a head at a particular position is called a “**track**”. All of the tracks across all disk surfaces for a given head position is called a “**cylinder**”. Around a track, the data is broken up into fixed sized blocks called a “**sector**”.

**Sectors are generally 512 bytes**, although different disk formats have had sector sizes from **as small as 128 bytes to as large as 2k bytes**. Whenever data is read from or written to the disk, it is **always in units of sectors**. The **location of a sector on the disk is specified by cylinder number, head number, and then sector number**. **Cylinder and head numbers start with 0. Sector numbers start with 1**.
On floppy disks, there is a small hole, called the index hole, in the disk material near the center. As the disk rotates, an optical sensor detects when this hole passes by. This index mark indicates the beginning of the sectors on the disk.

**Sector 1 starts at the index mark**, with the other sectors on the track following in succession. The data on the disk surface is made up of a preamble, a synch pattern, a sector header, the sector data, and then, the preamble for the next sector.

- The **preamble** is a standard pattern of bits that marks empty space between sectors.
- The **synch field** is a special bit pattern that allows the read electronics to get synchronized with the bit stream to follow.
- The **sector header** is also called the *address mark*, and contains the track number and sector number of the sector.

The data transfer rate onto or off from the disk is determined by the density of the bits and the linear speed at which the disk surface passes under the head. Magnetic disks generally always rotate at a constant velocity. This causes the linear speed of the disk surface to be higher near the outer edge than near the center.

If the **bit rate is held constant**, the bit density will be lower on the outer tracks than on the inner tracks, and each track will contain the same number of bits. **There is some maximum bit density that the magnetic material of the disk and the technology of the read/write heads can support**, and this determines the maximum number of bits that can be contained on the innermost track. Outer tracks contain the same number of bits, but they are not stored at the maximum density that the disk material could support.

**This is the way all floppy disks and most hard disks work. Some hard disks can vary the bit rate, and thus store bits at a higher rate on the outer tracks, achieving a higher bit density on the outer tracks than could be achieved if a constant bit rate was used, and thus increasing the disk capacity.**
The amount of time that it takes to access a particular sector on the disk is called the latency. The total latency is made up of several components.

- First, unless the head is already at the correct position, it will take time for the head to move (“seek”) from its current position to the correct cylinder and stabilize.
- Next, it takes the read electronics a small amount of time to synchronize to the data stream coming off of the disk, and at least one sector may pass under the head before it is synchronized and reliably reading the data.
- It is then necessary to wait for the requested sector to pass under the head before it can be read. This is called rotational latency, and averages one half of the disk rotation time for random accesses to the disk.

Accessing the Disk.

On IBM PC type machines, the disk can be accessed directly by using the BIOS disk services. The BIOS disk services are accessed via INT 13h.

The original IBM PC BIOS defined 6 disk services. Later versions added additional services, and there are now about 22 disk access functions defined by the BIOS. Some of these functions work only on floppy disk, some only on hard disks, and some are usable for either.

When dealing with the BIOS disk services, disk addresses are specified using cylinder, head, and sector numbers. It is necessary to know the logical format of the disk (i.e. how many head, how many cylinders, and how many sectors per track).
DOS provides two software interrupts for low level disk access. These are INT 25h for disk reads and INT 26h for disk writes.

**INT 25h (37) Absolute Disk Read**

Reads one or more sectors on a specified logical disk.

**On entry:**
- AL: Drive number (0=A, 1=B)
- CX: Number of sectors to read
- DX: Starting sector number
- DS:DX: Buffer to store sector read

**Returns:**
- AX: Error code (if CF is set; see below)
- Flags: DOS leaves the flags on the stack

**INT 26h (38) Absolute Disk Write**

Writes one or more sectors on a specified logical disk.

**On entry:**
- AL: Drive number (0=A, 1=B)
- CX: Number of sectors to write
- DX: Starting sector number
- DS:DX: Address of sectors to write

**Returns:**
- AX: Error code (if CF is set; see below)
- Flags: DOS leaves the flags on the stack

Error information is reported in AX as follows:

**Error code in AH:**
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01h</td>
<td>Invalid command</td>
</tr>
<tr>
<td>02h</td>
<td>Address mark not found</td>
</tr>
<tr>
<td>03h</td>
<td>Attempt to write on write-protected diskette</td>
</tr>
<tr>
<td>04h</td>
<td>Sector not found</td>
</tr>
<tr>
<td>05h</td>
<td>Reset failure</td>
</tr>
<tr>
<td>07h</td>
<td>Drive parameter activity failure</td>
</tr>
<tr>
<td>08h</td>
<td>DMA overrun</td>
</tr>
<tr>
<td>09h</td>
<td>DMA boundary error</td>
</tr>
<tr>
<td>10h</td>
<td>CRC or ECC data error</td>
</tr>
<tr>
<td>11h</td>
<td>Possible error corrected by ECC</td>
</tr>
<tr>
<td>20h</td>
<td>Controller failure</td>
</tr>
<tr>
<td>40h</td>
<td>Bad seek</td>
</tr>
<tr>
<td>80h</td>
<td>Drive timeout</td>
</tr>
<tr>
<td>BBh</td>
<td>Undefined error</td>
</tr>
<tr>
<td>FFh</td>
<td>Sense operation failure</td>
</tr>
</tbody>
</table>

*(AL contains burst length)*

**Error code in AL:**

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>Write-protect error</td>
</tr>
<tr>
<td>01h</td>
<td>Invalid drive number</td>
</tr>
<tr>
<td>02h</td>
<td>Drive not ready</td>
</tr>
<tr>
<td>03h</td>
<td>Invalid command</td>
</tr>
<tr>
<td>04h</td>
<td>CRC error</td>
</tr>
<tr>
<td>05h</td>
<td>Bad request structure length</td>
</tr>
<tr>
<td>06h</td>
<td>Seek error</td>
</tr>
<tr>
<td>07h</td>
<td>Unknown medium; disk format not recognized</td>
</tr>
<tr>
<td>08h</td>
<td>Sector not found</td>
</tr>
<tr>
<td>09h</td>
<td>Printer out of paper</td>
</tr>
<tr>
<td>0Ah</td>
<td>Write error</td>
</tr>
<tr>
<td>0Bh</td>
<td>Read error</td>
</tr>
<tr>
<td>0Ch</td>
<td>General, nonspecific error</td>
</tr>
</tbody>
</table>

Note that the AH error information is the same as that returned in AH by Interrupt 13h, Service 2, and the AL error information is the same returned in DL in response to a critical error (Interrupt 24h).

*(The above was obtained from www.clipx.net.)*
These DOS services are comparable to service provided by the ROM-BIOS in Interrupt 13h, except for these two differences:

1. **DOS numbers disk sectors sequentially**, starting at cylinder 0, head 0, sector 1. The BIOS service identifies sectors by three separate coordinates—cylinder, head, and sector. The following formula converts BIOS-numbered sectors to the DOS format:

   \[
   \text{DOS.Sector.Number} = (\text{BIOS.Sector} - 1) + \\
   \text{BIOS.Head} \times \text{Sectors.per.Head} + \\
   \text{BIOS.Cylinder} \times \text{Sectors.per.Head} \times \\
   \text{Heads.per.Disk}
   \]

2. **DOS works with logical drives**, while the **BIOS works with physical drives only**. That means that the DOS interrupt can be used to read a phantom drive B, a RAM drive, or a logical drive that has been mapped to a nondefault physical drive via the ASSIGN command.

For these functions, a disk address is specified as a logical sector number. All of the sectors on the disk are given a sequential number starting at 0. DOS knows the logical format of the disk, and translates logical sector number into the appropriate cylinder, head, and sector numbers and then calls on the BIOS services to perform the read or write.

Generally, the preferred way to access the disk is through the file system maintained by the operating system. DOS provides a number of functions for creating, deleting, opening, closing, reading, and writing files.
Organization of Data on the Disk

The logical organization of the FAT file system on a disk is made up of the following elements.

- BOOT Sector
- Root Directory Structure
- File Allocation Tables (FAT)
- Data Clusters

The **BOOT sector** is described in Section 9.2 of the text. It contains a data structure that describes the format of the disk, followed by the **boot loader**. The boot loader is a short program that loads the operating system from the disk into memory and starts it executing.

In the case of DOS, the boot loader actually loads the operating system. In the case of many other operating systems, such as Windows or Windows NT, the **boot loader in the boot sector only loads a more sophisticated loader**, which then loads the actual operating system.

The **Root directory structure contains the top level directory structure of the hierarchical file system**. The root directory is special, in that the information in it is stored in a hard coded location on the disk, and a fixed amount of space is allocated for it when the disk format is defined. If a disk has hard sector errors in the region where the root directory is stored, the disk is unusable. Sub-directories of the root directory are stored in special files.

A directory is a table of fixed sized directory entries.

Each directory entry is 32 bytes long and contains the **file name**, **creation and access dates and times for the file**, **file attributes**, and all other information that DOS knows about the file.

For the root directory, this table is stored at a fixed location on the disk. Sub-directories are special files that contain directory entries.
Windows NT and versions of Windows beginning with Windows '95, long file names are supported. Files with a long file name (or a name containing embedded spaces) will have multiple directory entries.

- There will be a directory entry containing a short file name (confirming to the 8.3 character file naming convention of DOS).
- There will also be multiple additional directory entries containing the complete file name. The number of directory entries will vary depending on the length of the file name.

These directory entries immediately precede the short file name entry in the directory, and contain reserved fields with special values that cause earlier versions of DOS to ignore them and to indicate that they are all bound together to make up a complete file name.

One of the key pieces of information that is stored in the root directory is the head of the FAT chain. This is the information that allows the DOS file manager to find the data associated with the directory entry (i.e. the actual file data for the file).

Space on a disk is allocated in units called a cluster. A cluster is some number of sectors that are treated by DOS as an atomic unit. When space is allocated for a file, it is always allocated a cluster at a time.

- On small disks, such as floppy disks, a cluster is 1 sector.
- On large hard disks, a cluster may be as large as 16 or 32 sectors.

Each cluster on a disk is assigned an index number, starting at 0. The file allocation table (FAT) is an array of entries indexed by the cluster number.

If the cluster is free, the FAT entry for cluster will have a special value in it indicating that it is free.

If the cluster contains a bad sector, there will be another special value that marks the cluster as bad, and not to be used.
Clusters that are part of a file, will be linked together into a linked list of all of the clusters that make up the file. So, for clusters that are part of a file, the FAT entry for a cluster will contain the index of the next cluster in the file. The last cluster assigned to a file will have a special value indicating that it is the end of the chain for that file. The directory entry for a file has a field that indicates the cluster number of the first cluster assigned to the file (this is the head of the linked list).