Input/Output
Disks

- Plates of magnetic material, organized in cylinders, divided in tracks, divided in sectors. Sectors go up to several hundreds. Heads vary from 1 to 16.
- Some have little electronics, IDE drives have a microcontroller in the disk itself.
- Controller: some can do overlapped seeks that is, while waiting for one seek to complete on one disk it can start another one on another disk.
### Example of the numbers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>IBM 360Kb floppy</th>
<th>WD 18300 HD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of cylinders</td>
<td>40</td>
<td>10601</td>
</tr>
<tr>
<td>Tracks per cylinder</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Sectors per track</td>
<td>9</td>
<td>281 (average)</td>
</tr>
<tr>
<td>Sectors per disk</td>
<td>720</td>
<td>35742000</td>
</tr>
<tr>
<td>Bytes per sector</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>Disk capacity</td>
<td>360K</td>
<td>18.3G</td>
</tr>
<tr>
<td>Seek time</td>
<td>6ms</td>
<td>0.8ms</td>
</tr>
<tr>
<td>Seek time (average)</td>
<td>77ms</td>
<td>6.9ms</td>
</tr>
<tr>
<td>Rotation time</td>
<td>200ms</td>
<td>8.33ms</td>
</tr>
<tr>
<td>Motor start/stop</td>
<td>250ms</td>
<td>20s</td>
</tr>
<tr>
<td>Time to transfer 1 sector</td>
<td>22ms</td>
<td>17μs</td>
</tr>
</tbody>
</table>
Organization of the disk

- To simplify searching for sectors on the disk most disk presents:
  - A virtual organization in (c, t, s) that are mapped to the physical one
  - The number of sector per track changes as we move along the disk (minimum size of the magnetic site)
- On modern disks there’s something called LBA (logical block addressing) where sectors are numbered sequentially without considering cylinder, track, or sector
Disk formatting

- **Cylinder skew**
  - Start cylinders at different points to give the head time to jump from one to the next

- **Interleaving**
  - To give the controller time to transfer to main memory

- **Scheduling of head movements**
  - Controllers can schedule the movements of the head

512 bytes

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Data</th>
<th>ECC</th>
</tr>
</thead>
</table>

16 bytes

Disk sector

OS 2009-10
The controller can transparently take care of replacing a *bad* sector with a *spare* sector.

Bad sectors are due to wear and tear of the magnetic medium or construction defects (technology is always pushed to the extreme).
Stable storing

- Two identical disks
  - Probability of same block spontaneously go bad on two disk is negligible
- Stable write:
  - Write block on disk 1, read it back (re-read until it works up to N times). This would eventually work. Write on disk 2. Same procedure. In absence of CPU crashes the block is written twice
- Stable read:
  - Read from block 1. Read up to N times. If disk 1 went bad for any reason, read from disk 2. Since a stable write always succeeds also the read succeeds
- Crash recovery:
  - See next slide
The recovery program scans both disks. If a pair is good nothing is done, if one has an error (ECC) copy the good one into the bad one. If a pair has both block good but different, write 1 onto 2.
A “strip” is K sectors

RAID 0
Strip 0
Strip 1
Strip 2
Strip 3
Strip 4
Strip 5
Strip 6
Strip 7
Strip 8
Strip 9
Strip 10
Strip 11

RAID 1
Strip 0
Strip 1
Strip 2
Strip 3
Strip 4
Strip 5
Strip 6
Strip 7
Strip 8
Strip 9
Strip 10
Strip 11

RAID 2
Bit 0
Bit 1
Bit 2
Bit 3
Bit 4
Bit 5
Bit 6

RAID 3
Bit 0
Bit 1
Bit 2
Bit 3
Bit 4

Parity

RAID 4
Strip 0
Strip 1
Strip 2
Strip 3
Strip 4
Strip 5
Strip 6
Strip 7
Strip 8
Strip 9
Strip 10
Strip 11
Parity 0-3
Parity 4-7
Parity 8-11

RAID 5
Strip 0
Strip 1
Strip 2
Parity 0-3
Strip 4
Strip 5
Parity 4-7
Strip 8
Parity 8-11
Strip 9
Strip 10
Parity 12-15
Strip 3
Strip 7
Strip 11

OS 2009-10
CD technology

- **Pits and lands:**
  - Pit: depression in the plastic
  - Land: unburned area
- **Different reflectance of the pits**
  - It can be identified as a 1 or 0
- **Recording follows a spiral**
- **Various materials for CD-R and CD-RW**
- **Improvement for DVD (also changed the laser wavelength)**
Clocks

- Preventing one process to monopolize the CPU
- Maintaining the time of the day
- Accounting for CPU usage
- Handling ALARM signals
  - A device driver considers all the events (e.g. alarms) within the system (not a timer for every process/thread)
- Providing watchdog timers for part of the OS
  - e.g. timeouts – stop rotation of disk if not used
- Doing profiling, monitoring, statistics
Soft timers

- Handling an interrupt at each clock time would be too much
  - Interrupts involve delays
  - Can only be serviced with certain rates
- Idea! Why not calling the timer handling whenever in kernel mode
  - Occurrence of system calls, TLB misses, page faults, I/O interrupts, CPU idle, etc.
  - Experimentally
    - Going in kernel mode on average between 2 and 18\(\mu\)s
- Combine soft timer with a lower frequency one
• **Terminals**
  - Character oriented
  - Over a serial line

• **GUIs**
  - MS Windows (GUI into the kernel)
  - X-Windows (GUI as a user process, native networking)
    - X-server: does the display
    - X-client: an application

• In practice the controller is seen by the OS as a buffer (the video buffer) mapped in memory
Last but not least

- Since batteries are big, expensive, and not particularly efficient
- Need to save power
- Power management (on laptops)...
- ...but also on desktop computers
  - For fun. Each desktop has 200W power supply, 85% efficient. 100,000 PCs consume 20MW equivalent to 20 average-size nuclear power plants