Devices

- Specialized units that implements functionality. E.g. disks, NIC card, camera, graphics card etc.

- Structure
  - **Hardware Interface**: Similar to issuing instructions on CPU to perform operations (like addition or copy etc.), instructions need to be sent to the device to achieve the desired operation to make the device perform some operation (like read or write from disk). A typical device usually exposes a set of registers (status, data and command register) for the CPU to communicate with them.
  - **Internals**: Hardware interface is the abstraction exposed by the device to the outside world to communicate with the device. In order to actually perform the operations like read/write or send/receive packet) it guarantees, it contains more hardware logic. These could vary for different devices - e.g. Though keyboard primarily sends the characters typed by the user to OS, different models could support different hotkeys.
  - **Software Interface**: Typically this is referred as drivers or modules and are needed since OS does not know about the hardware interface exposed by the device. The hardware vendor usually provide these software modules. E.g. You need the driver provided by the keyboard vendor to get the special hotkeys to work.

Reduce overhead

- Once a task has been assigned to the device, the only way to identify if the task has been completed by the device is by checking the status register. This results in wasting CPU resource. **Interrupts** are used to make the device notify the CPU once it is done and thus enabling asynchronous activities.
- Certain task requires data to be sent from memory to the device (e.g. sending a packet via network). Again, cpu is wasted by copying every byte of data to the device. Instead **DMA** mechanism allows offloading the copy job to a separate logic thus freeing cpu to do useful work.

Communication with device

- It is mentioned before that drivers are responsible for communicating with devices. Following are the two ways by which drivers could communicate with devices.
  - Programmed IO
  - Memory mapped IO

Hard Disk Drives

- Primary storage device in today's system (Though SSDs are available, hard disk still provides higher capacity)
- Data is stored in a magnetic medium called as a **platter** and a single disk could contain multiple platters that are rotates around the a central axis which is called as a **spindle**.
- Magnetic medium could be visualized as a concentric circles where every circle is referred as **tracks** and data is stored along tracks.
- Tracks are divided into **sectors** (512 bytes) that is the atomic unit guaranteed by the disk.
Accessing data

- Data is laid out along the tracks and there could be millions of tracks.
- **Head unit** is responsible for reading/writing data from the magnetic medium and a head unit exists for every platter.
- The disk always spins and the head unit could be moved around to the desired sector to read/write.
- **Latency**
  - Rotational delay: If data is located in the same track but not in the successive sector, then head unit has to wait till
  - Seek delay: If data is located in a different track then head is moved to the desired track

Other operations

- Track skew: To avoid rotational delay when successive data is stored in the next track.
- **Zones**: Storage area is divided into zones where tracks in different zones have different number of sectors
- **Caches**: The hardware internal could include a cache that could result in improved performance

Disk scheduling

- **SSTF (Shortest seek time first)**
- **Elevator**
  - F-SCAN
  - C-SCAN
- **SPTF (Shortest positioning first)**

Questions

- Does the seek delay and rotational delay apply to any flash drives? Why?
- What are the problems caused by using a generic driver for devices?
- If accessing disk is costly, how can the delays be avoided to improve performance?