Overview

We've explored how the what an I/O is and how an OS interacts with it. Now let's look at the hard disk drive and how to store and access data on one.

This section should help us the following questions:

- **How do modern hard-disk drives store data?**
- **How is the information organized and accessed?**
- **How does disk scheduling improve performance?**

Introduction

Hard disk drives are persistent storage devices for computers. It's important to understand the inner workings of a disk before creating the file system software that controls it.

Interface

A hard drive should have the ability to both **read** and **write**.

Internals

Internally, a hard drive has:
* A **controller**
* This exports the **interface**
* It also controls the operation of each request given to the device

- **Mechanics**
  - Disk platters
  - Arm
  - Head, etc.

▼ **Persistent**

Persistent storage means that when we turn the power to the device off and back on, the data will still be there.

The Interface

The hard disk **interface** connects the hard disk to the host computer and transfers data between the hard disk cache and the host memory. The quality of the hard disk interface has a direct impact on the program's speed and system performance.

A hard disk drive's interface includes several **sectors** (512-byte blocks) making up the drive. Each one can be read or written to. It's like an array of n sectors, with an **address space** ranging from 0 to $n - 1$.

There are five different types of hard disk interfaces:
- **IDE**
- Hard drives with an IDE interface are mostly used in consumer electronics and, to a lesser extent, in servers.
- **SATA**
- SATA is mostly used in the home market, with SATA, SATA II, and SATA III being the most common.
- **SCSI**
- Servers are the primary users of hard drives using SCSI interfaces.
- **SAS**
- **Fiber Channel**
- Fiber Channel is only used in sophisticated servers thanks to its high cost.

A variety of specialized interface types exist under the broad category of **IDE** and **SCSI**. They have varied transmission rates and technical requirements.
#
Checkpoint
Fill in the blanks to complete the statement below.

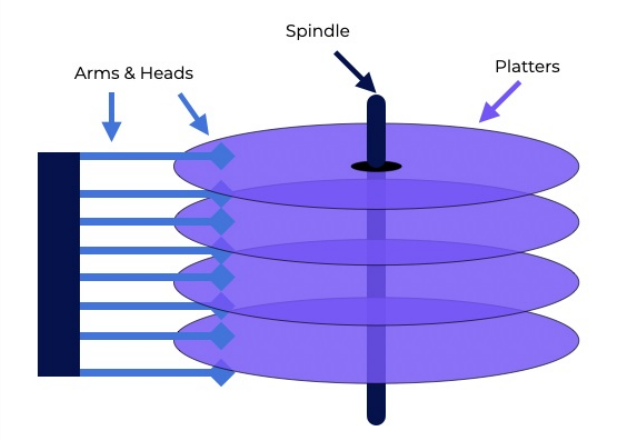
Basic Geometry

Let's look at what makes up a modern disk drive.

We have a **platter**, a hard circular surface where data is permanently stored by causing magnetic variations. A disk can have one or more platters, each with two sides, called **surfaces**.

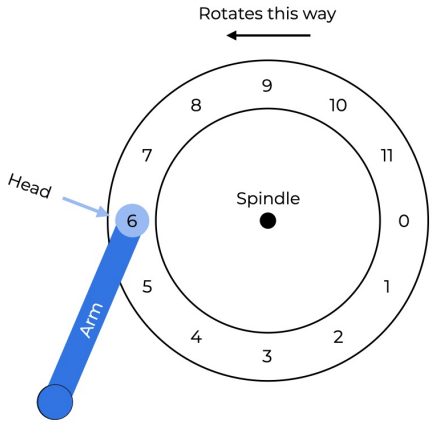
These platters are made of a hard material (like aluminum) and covered with a thin magnetic coating that lets the drive store bits even when it is turned off.

The platters are connected around a **spindle** coupled to a motor that spins them at a constant speed measured in **rotations per minute (RPM)**. Normal current values range from $7,200$ to $15,000$ RPM. Many times, we'll want to know the time it takes for it to complete a single spin.



Each surface has data encoded in nested circles called **tracks**. Thousands of tracks are packed together on one surface. **Clusters** are subdivided portions of these tracks. Two or more **sectors** make up a cluster. A cylinder is a vertical collection of one set of tracks stacked on top of each other.

The **disk head**, one per surface of the drive, does the reading and writing by sensing (i.e., read) or creating a change in (i.e., write) the magnetic patterns on the disk. One disk **arm** is attached to the disk head that moves across the surface to put it over the track we want.

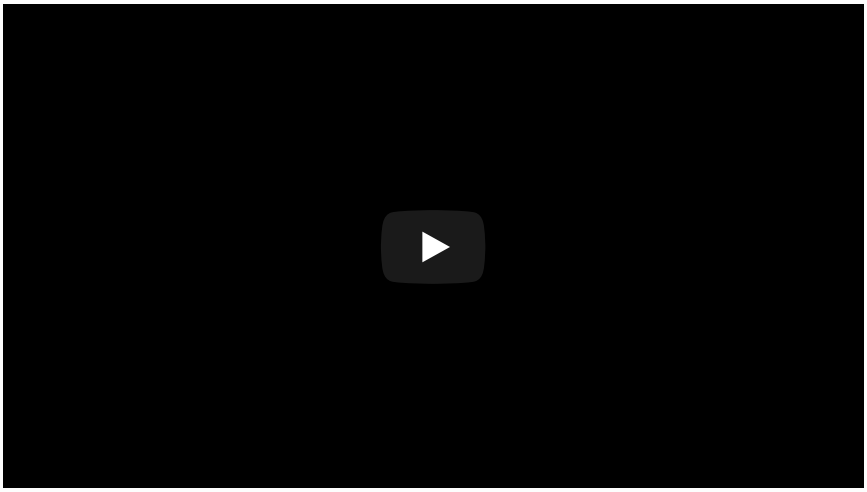


Read/Write operations generally have 3 parts:

- **Seek** - positioning the read/write head over desired track
- **Rotation (Rotational Delay)** - Waiting for the target sector to rotate under the head
- **Transfer** - performing the read/write

A well designed drive will spend most of its time transferring.

Check out this video of a real hard disk drive in slow motion! Can you identify all of the basic parts?



warning

⚠ Warning

Don’t pour water on your hard drive! (It probably won’t end well)

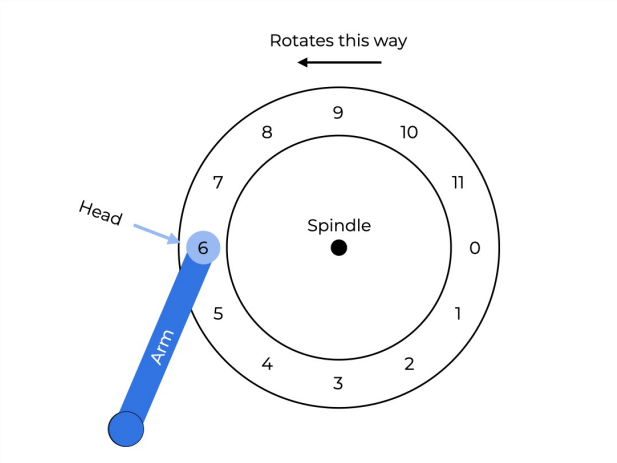
Checkpoint

Fill in the blank to match the term with its description. Every definition won't be used.

A Simple Disk Drive

Say we have a single-track disk. This track has 12 sectors, each is 512 bytes in size and is addressed by numbers 0 through 11. The single platter here spins around a spindle driven by a motor.

The disk head is above sector 6 in the graphic, and the surface is spinning counter-clockwise.



Single-track Latency: The Rotational Delay

Imagine we get a request to read block 0 on our one-track disk.

Our disk has to wait for this sector to rotate under the disk head. This wait is called **rotational delay**. This partially depends on how fast the disk platters are spinning.

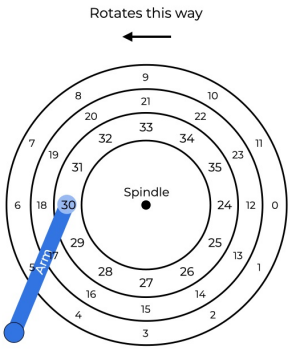
If the rotational delay is R to spin around the entire disk once, the disk has to wait about $\frac{R}{2}$ for 0 to come under the read/write head (if we start at 6). So, a worst-case request on this single track would be to sector 5, creating a near-full rotational delay.

Various Tracks: Seek Time

Our disk only has one track, which is unrealistic. Modern disks have millions. Let's look at a more realistic disk surface with three tracks.

- The head is currently over the innermost track (sectors 24–35).
- The next track over has sectors (12–23), and

- The outermost track contains the initial sectors (0 through 11).



To accesses a particular sector, like sector 11, the drive has to first **seek** the disk arm to the right track. Along with **rotations**, **seeks** are the most expensive disk operations.

The disk arm positioned the head over the correct track after the seek. The arm was moved to the desired track, and the platter was rotated by 3 sectors. Sector 11 is transferred under the disk head with only a little **rotational delay**.

Data is read from or written to the surface when sector 11 passes under the disk head.

Checkpoint

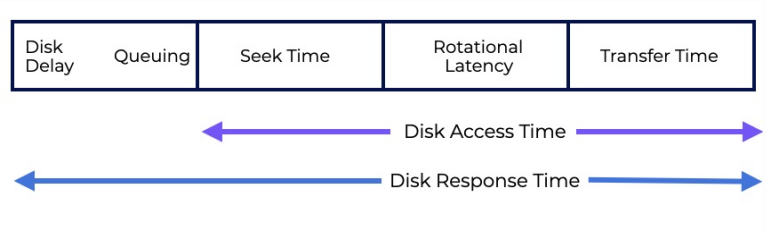
Fill in the blanks to describe the parts of a read/write operation.

Time for I/O: Doing the Math

Now that we have a model of the disk, we can analyze its performance. We can represent **Disk Access Time** as:

$$T_{I/O} = T_{seek} + T_{rotation} + T_{transfer}$$

Disk Response Time is the average time a request spends waiting for an I/O operation. The average response time is the average response time across all requests.



.guides/img/diskResponse

The **rate of I/O ($R_{I/O}$)**, which is often used to compare drives, is computed from the formula below:

$$R_{I/O} = \frac{Size_{Transfer}}{T_{I/O}}$$

Two markets are important when designing disk drives.

1. A **high performance** drive is designed to spin quickly, have low seek times, and transfer data quickly.
2. The **capacity** market is where the cost per byte is important. The drives are slower but pack as many bits as possible into the available space.

Let's compare the performance of 2 hard drives. Their specifications are listed below.

Specs.	Drive A	Drive B
Capacity	1 T	36.7 GB
RPM	5400 RPM	7200 RPM
Average Seek	13 ms	6.3 ms
Max Transfer	140 MB/s	160 MB/s
Platters	1	4
Cache	128 MB	4 MB
Connects via	SATA	SCSI

There are two different types of workloads to consider.
* **Random workloads** read small ($4KB$) blocks of data from the disk at **random**. Database management systems, in particular, use random workloads.
* **Sequential workloads** read several sectors from the disk in a linear fashion. Sequential access patterns are pretty common.

Using these values, we can estimate how well the drives will perform under our two workloads.

Random Workload

Say we have a $4KB$ read request to a random disk location, if we calculate how long each read would take on each drive, we'd get:

Measurement	Drive A	Drive B
T_{seek}	$13ms$	$6.3ms$
$T_{rotation}$	$5.5ms$	$4.15ms$
$T_{transfer}$	$28.5microsecs$	$25microsecs$

So we can calculate $T_{I/O}$ for each drive:

$$TA_{I/O} = 13ms + 5.5ms + 28.5microsecs \approx 18.5ms$$

$$TB_{I/O} = 6.3ms + 4.15ms + 25microsecs \approx 10.45ms$$

Here, microseconds barely make a difference in our time because they're so tiny.
Considering our, $4KB$ request, calculating $R_{I/O}$ for each drive would go as follows:

$$\begin{aligned} RA_{I/O} &= \frac{4KB}{18.5ms} = \frac{0.004MB}{18.5ms} = \frac{4MB}{18.5sec} \approx 0.22MB/sec \\ RB_{I/O} &= \frac{4KB}{10.45ms} = \frac{0.004MB}{10.45ms} = \frac{4MB}{10.45sec} \approx 0.38MB/sec \end{aligned}$$

Sequential Workload

In this example, we can count on there being one seek and rotation before a long transfer. Let's say our transfer size for this is $500MB$

$T_{I/O}$ for each drive would be:

$$\begin{aligned} TA_{I/O} &= 13ms + 5.5ms + 3125ms \approx 3143.5ms \\ TB_{I/O} &= 6.3ms + 4.15ms + 3570ms \approx 3580.45ms \end{aligned}$$

And $R_{I/O}$:

$$\begin{aligned} RA_{I/O} &= \frac{500MB}{3143.5ms} = \frac{500MB}{3.1435sec} \approx 159.05MB/sec \\ RB_{I/O} &= \frac{500MB}{3580.45ms} = \frac{500MB}{3.58045sec} \approx 139.65MB/sec \end{aligned}$$

So our two drives perform as summarized below:

Workload	Drive A	Drive B
$R_{I/O}$ Random	$0.22MB/sec$	$0.38MB/sec$
$R_{I/O}$ Sequential	$159.05MB/sec$	$139.65MB/sec$

We can see a big difference in performance between the two workloads for both drives. Customers are often willing to spend on a high-performance drive while trying to get as much capacity as possible for as little money as possible.

Checkpoint

Given the following measurements, calculate the $R_{I/O}$ in MB/sec for a $10KB$ request.

Note: 1 MB = 1000 KB; 1 sec = 1000 ms

Measurement	Drive
T_{seek}	$8ms$
$T_{rotation}$	$3ms$
$T_{transfer}$	$50microsecs$

Summary

We've covered the basics of how disks operate.

- We've covered the basic geometry and anatomy of a hard disk drive.
- We also discussed key measurements in determining hard disk drive performance.

Using the proper I/O scheduling paradigm is important for maximizing efficiency during various types of I/O requests. We will talk about these next.