Hard Drives

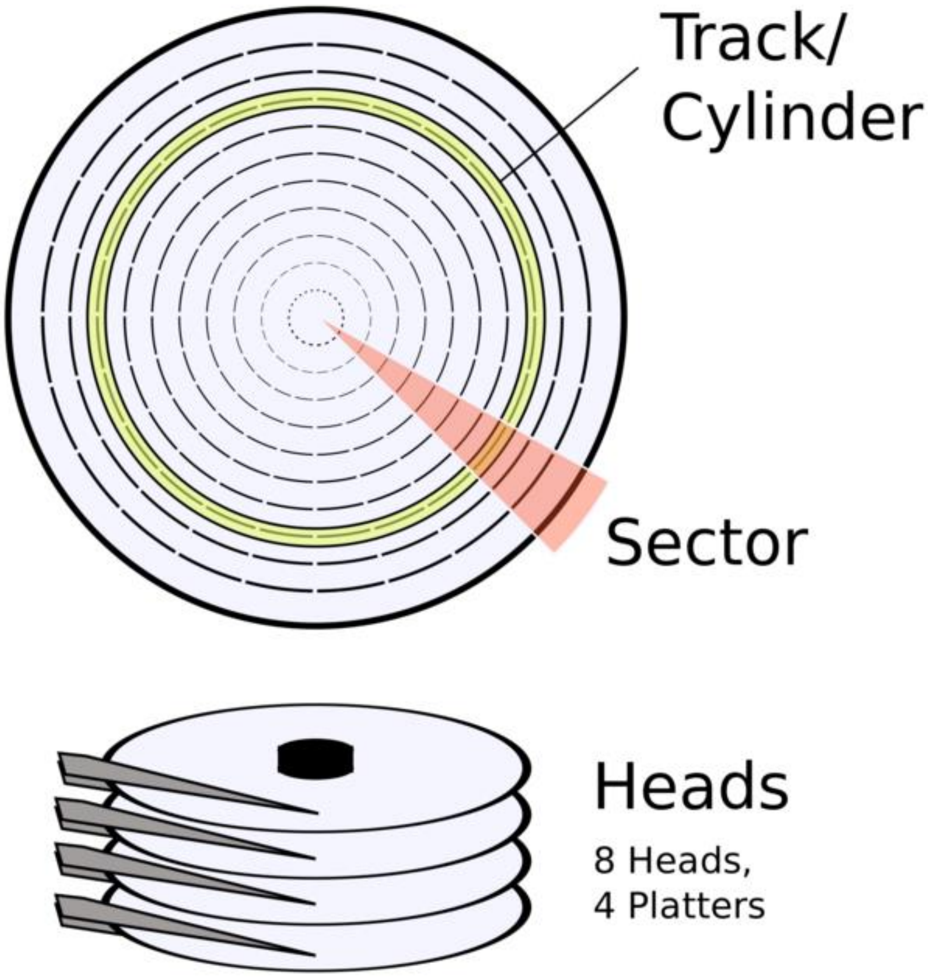
Glenn Bruns
CSUMB

Lecture Objectives

After this lecture, you should be able to:

- explain performance characteristics of hard drives
- compute hard drive transfer rates for different workloads
- describe the operation of some I/O schedulers

Hard drives

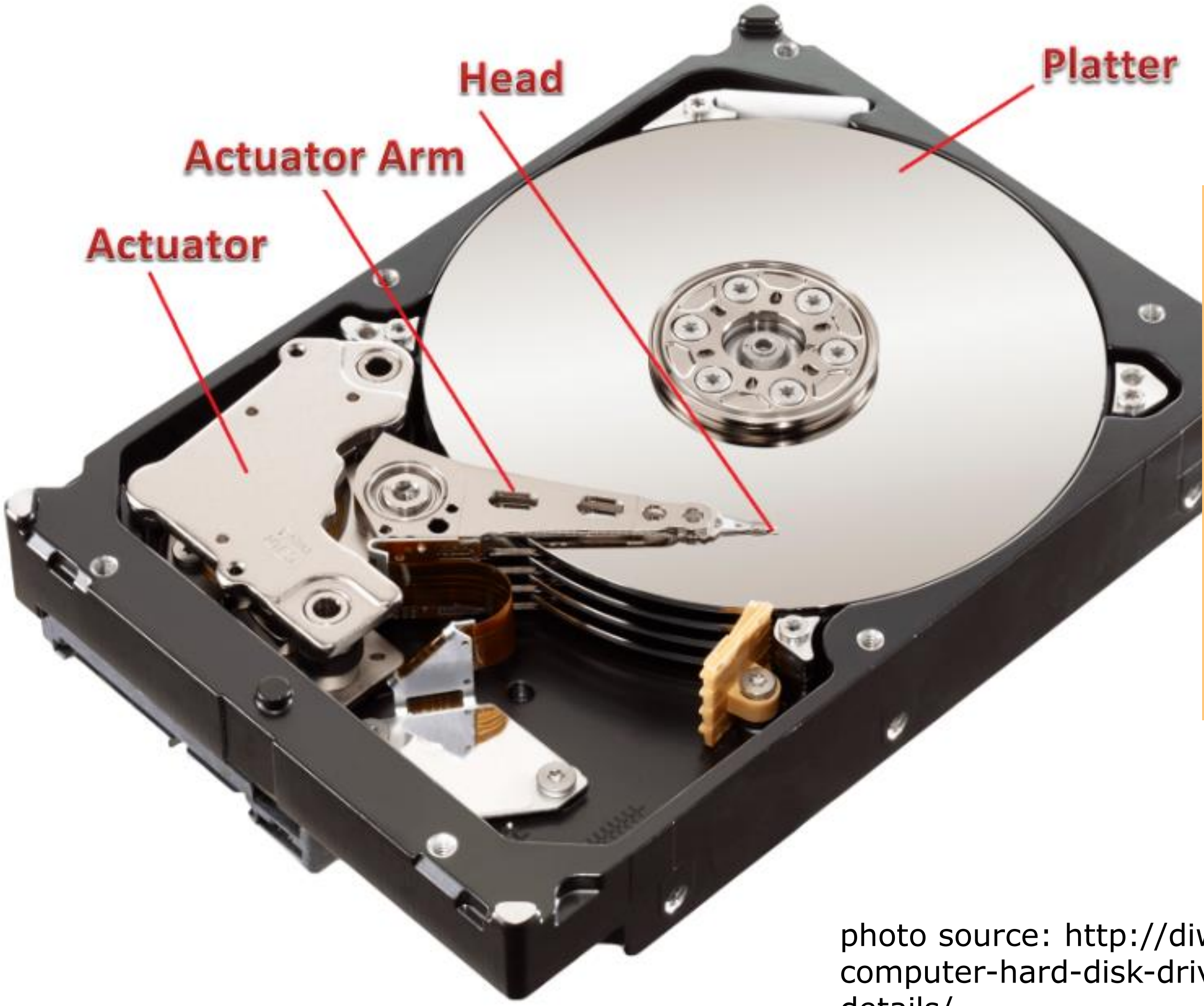


Usually one head for each surface of a platter.

Sectors toward the outside of a platter are bigger.

Physical unit of storage = a block

Photo detail



The heads are tiny.

The space between a head and the disk surface is about 5000x thinner than a human hair.

photo source: http://diwakarpro.com/types-computer-hard-disk-drives-hdd-complete-details/

Seagate video

Seagate video   https://www.youtube.com/watch?v=NtPc0jI21i0

- About 300,000 tracks per inch (along radius) !
- There is actually aerodynamics involved in the design of the head so it will float over the disk surface

Head stack assembly, from Seagate video

A drive in operation

https://youtu.be/p-JJp-oLx58?t=406 (Van Svenson)

A hard drive is a block device

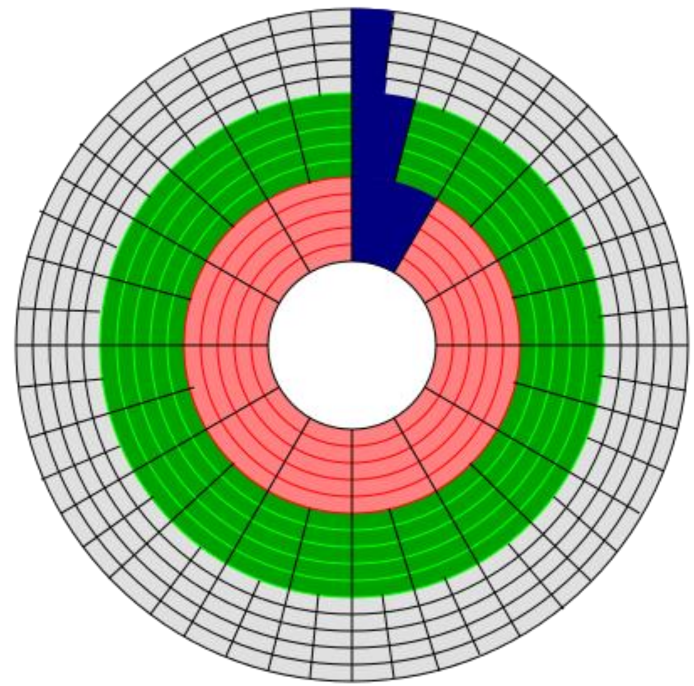
In early disk interfaces, "CHS addressing" was used:

- **C**ylinder
- **H**ead
- **S**ector

Now "logical block addressing" (LBA) is used. A "linear addressing scheme"

Analogy:

- CHS is like street number/street/city
- LBA is like giving every home a single number

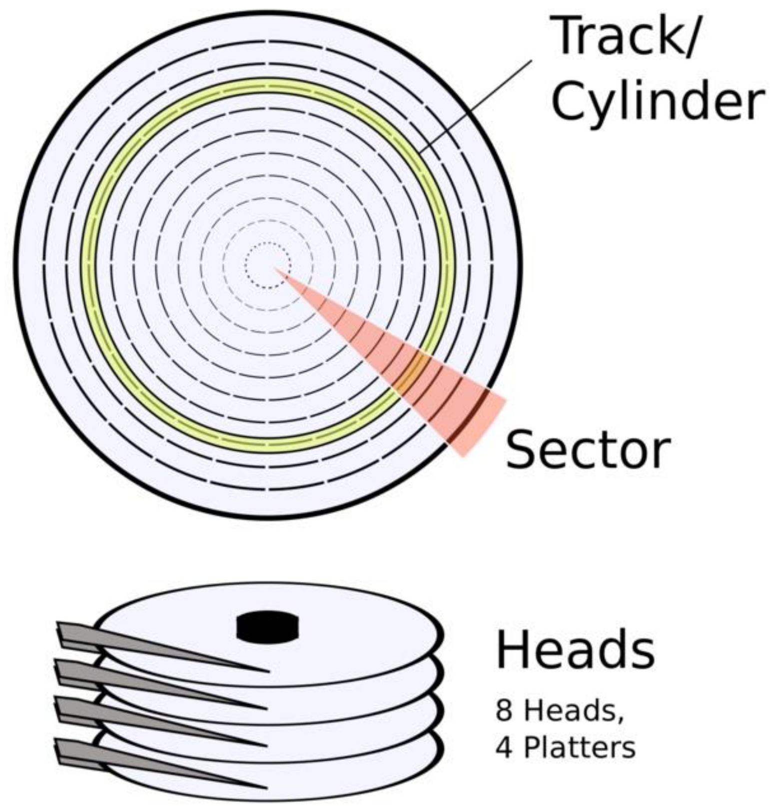


multi-zoned drive

Drive performance

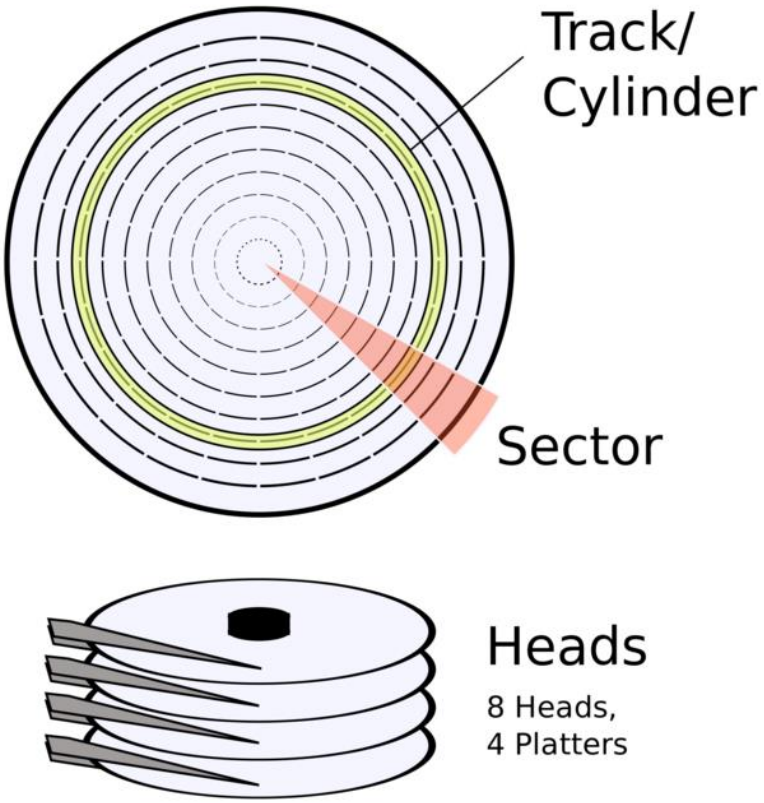
Reading/Writing of a drive involves 3 steps:

1. wait for sector to rotate underneath head (rotational delay)
2. move the head to the right track (seek)
3. actually transfer the data (transfer)



How long to transfer data?

- if a drive spins at 7200 RPM, rotational delay is about 4 ms
- typical seek time is about 5 ms
- typical max transfer rate is about 100MB/sec



"access time" = rotational delay + seek time

Calculating time for a read or write

1. wait for sector to rotate underneath head (rotational delay)
2. move the head to the right track (seek)
3. actually transfer the data (transfer)

assume:

rotational delay = 4 ms

seek time = 5 ms

transfer rate = 100 MB/s

amount to be read is 2 MB

How long will this take?

Access time = $4\,ms + 5\,ms = 9\,ms$

Transfer time = $\frac{1\,s}{100\,MB} * 2\,MB = 0.02\,s * \frac{1000\,ms}{s} = 20\,ms$

Total = 9 ms + 20 ms = 29 ms

Seagate Cheetah 600 MB specs

Performance	
Spindle Speed (RPM)	15K
Average Latency (ms)	2.0
Seek Time	
Average Read/Write (ms)	3.4/3.9
Track-to-Track Read/Write (ms)	0.2/0.4
Internal Transfer Rate, Outer to Inner Diameter(MB/s)	1450 to 2370
Sustained Transfer Rate, Outer to Inner Diameter(MB/s)	122 to 204
Cache, Multisegmented (MB)	16

(from Amazon)

Calculating average rotational delay

If a disk spins at 7200 RPM, what is the average rotational delay?

$$7200\ RPM = \frac{7200\ rev}{min}$$

$$\frac{7200\ rev}{min} * \frac{min}{60\ sec} = \frac{120\ rev}{sec}$$

$$\frac{1\ sec}{120\ rev} * \frac{1000\ ms}{sec} = 8.3\frac{ms}{rev}$$

so about 4.2 ms per ½ revolution

in one go:

$$\frac{min}{7200\ rev} * \frac{60\ sec}{min} * \frac{1000\ ms}{sec} = \frac{600\ ms}{72} = 8.3\frac{ms}{rev}$$

Workload analysis

Drive performance will depend a lot on how drive is used.

Consider two cases:

Random workload:

- read 32 KB from each of 100 random locations

Sequential workload:

- read 3.2 MB sequentially

Assume access time is 10 ms, transfer rate is 100 MB/sec

Exercise

Given: access time is 10 ms, transfer rate is 100 MB/s

We need to read 3.2 MB.

1. How long to do a sequential read of 3.2 MB?

2. How long to do 100 random reads of 32 KB each?

Random vs. Sequential workload

Sequential workload:

- time: access time + transfer time ~ 42 ms
- overall rate: 3.2 MB/0.042 s ~ 76 MB/sec

Random workload:

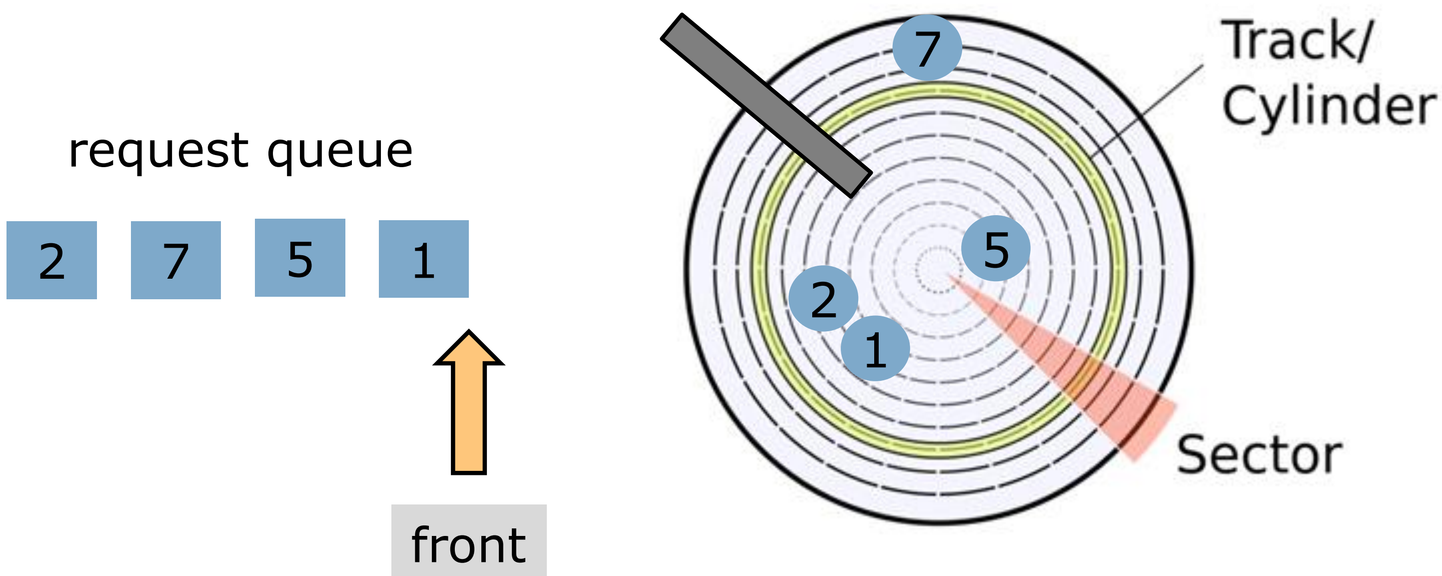
- time: 100 * (access time + transfer time) ~ 1030 ms
- overall rate: 3.2 MB/1.03 s ~ 3.2 MB/sec

Overall rate is about **24x** better for sequential workload

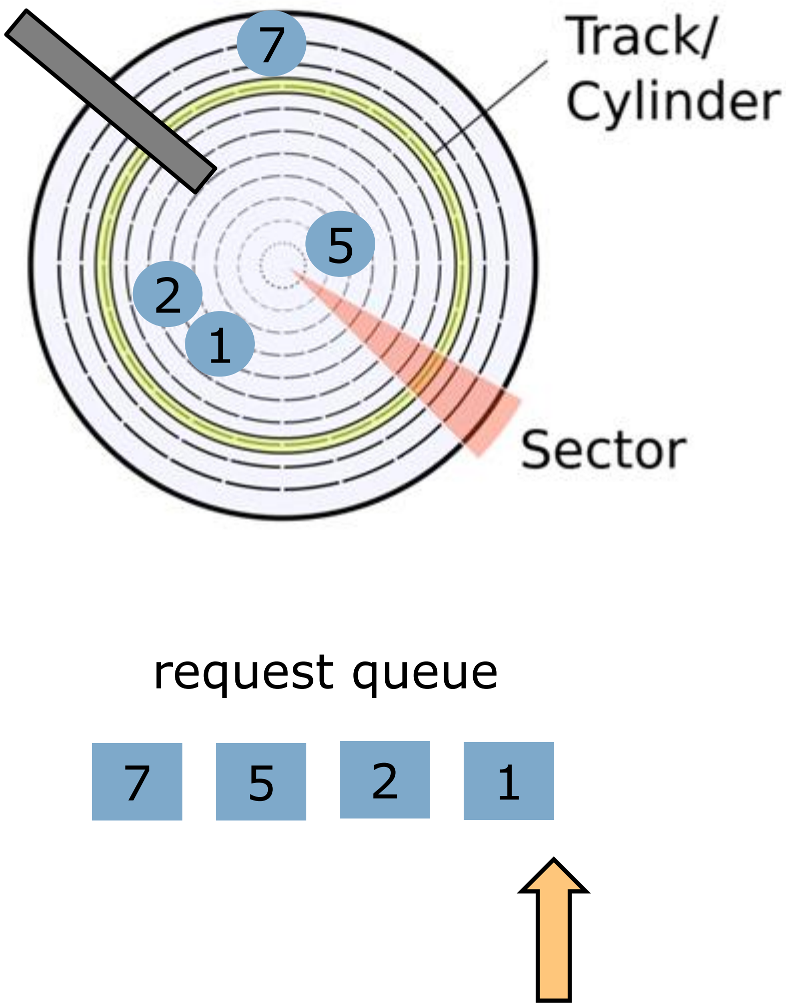
Disk scheduling (aka "I/O scheduling")

Sending requests to a disk in the order the OS receives them → terrible performance

The disk scheduler decides the order in which disk requests should be processed.



Shortest Seek Time First (SSTF)



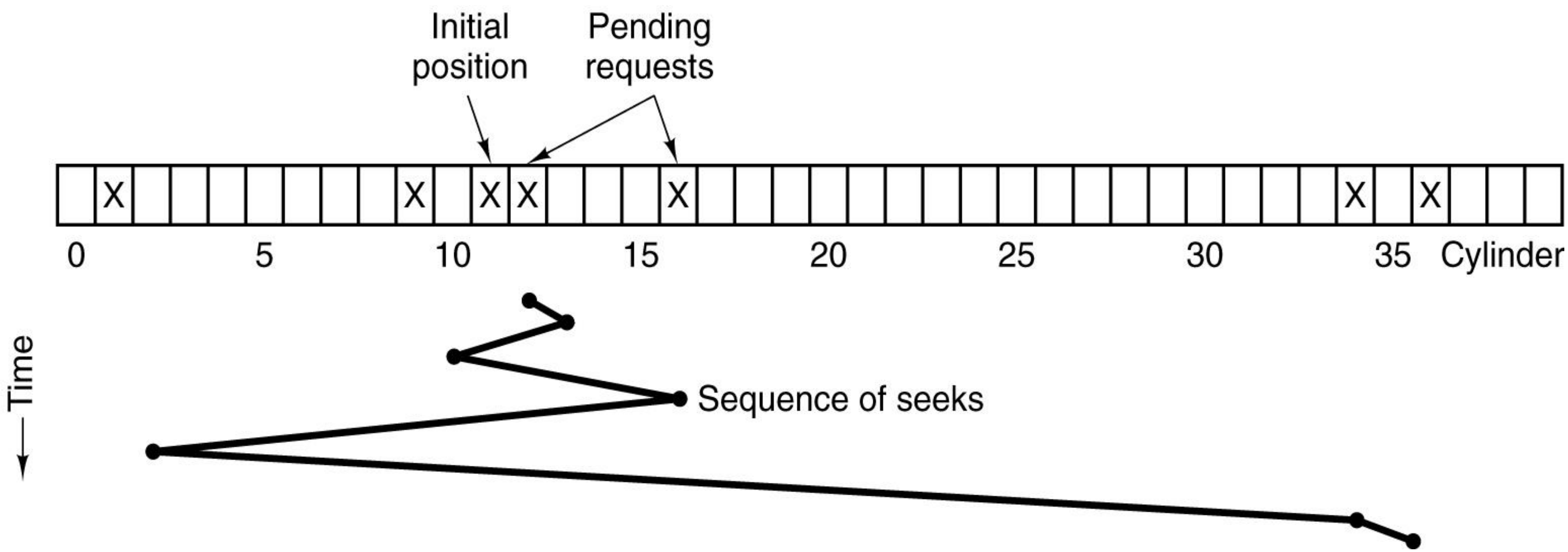
One way to do I/O scheduling: put requests closest to current track at front of queue.

Problem 1: the block address doesn't give track number.

Problem 2: …?

Visualization of SSTF

A request arrives to read a block on cylinder 11. While the seek is in progress, new requests come in for cylinders 1, 36, 16, 34, 9, and 12, in that order.



Text and figure from Tananbaum, Modern Operating Systems

Elevator scheduling

I/O scheduling is like elevator scheduling

What would happen if an elevator always picked up people from the closest floor?

starvation

Linux I/O schedulers

- Linus Elevator
  - performs merging and sorting (replaced in 2.6)
- Deadline I/O Scheduler
  - gives up elevator approach if old requests exist
- Anticipatory I/O Scheduler
  - waits a few ms after a seek for more read requests (Linux 2.6 default)
- Completely Fair Queuing I/O Scheduler
  - one queue for each process
  - designed for multimedia workloads
- Noop I/O Scheduler
  - maintains request queue in FIFO order

Summary

- disk drive hardware
- disk drive performance specs
- workload analysis
- I/O scheduling