ILLINOIS TECH

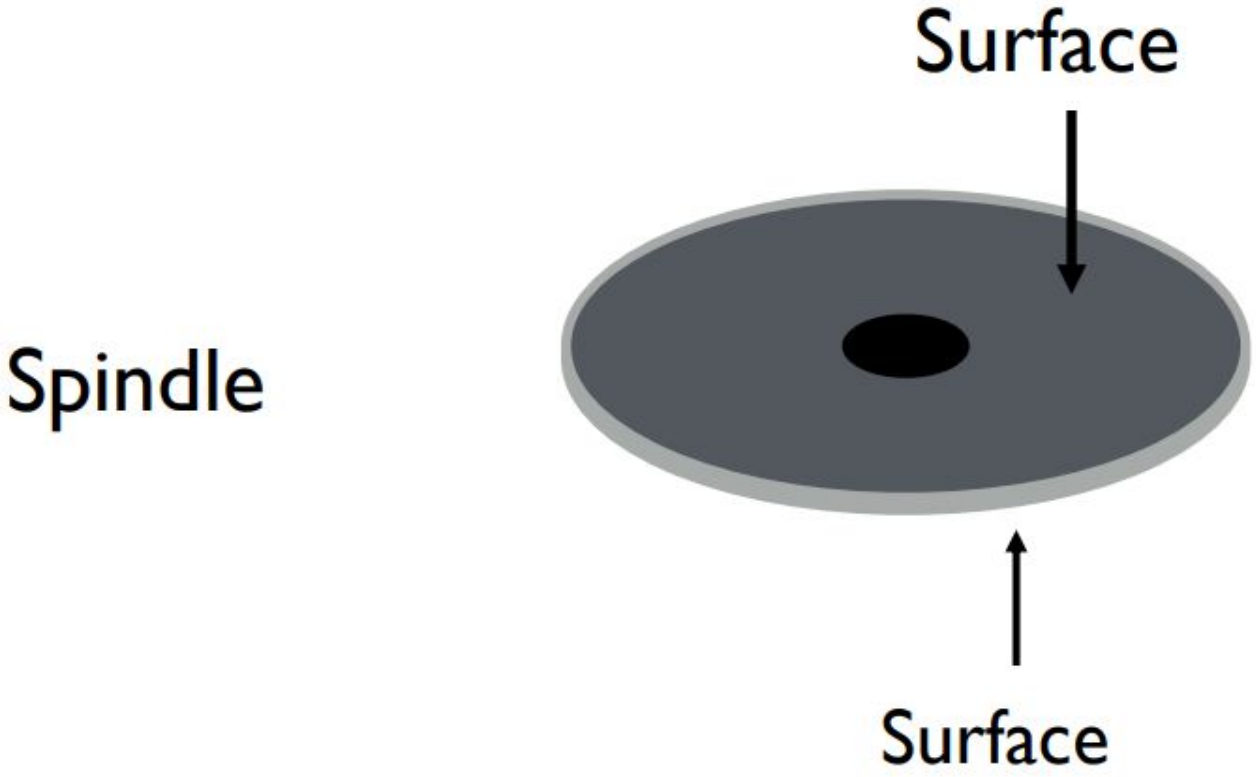
College of Computing

CS 450 Operating Systems

Hard Disks

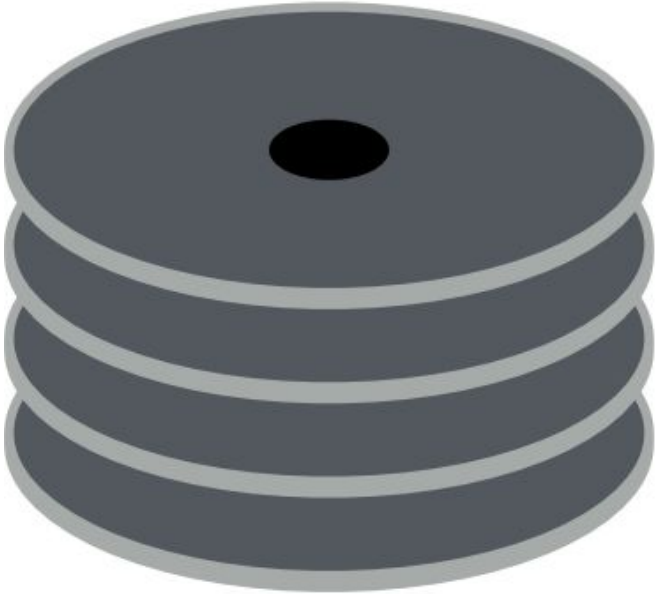
Yue Duan

Recap: Hard Disks



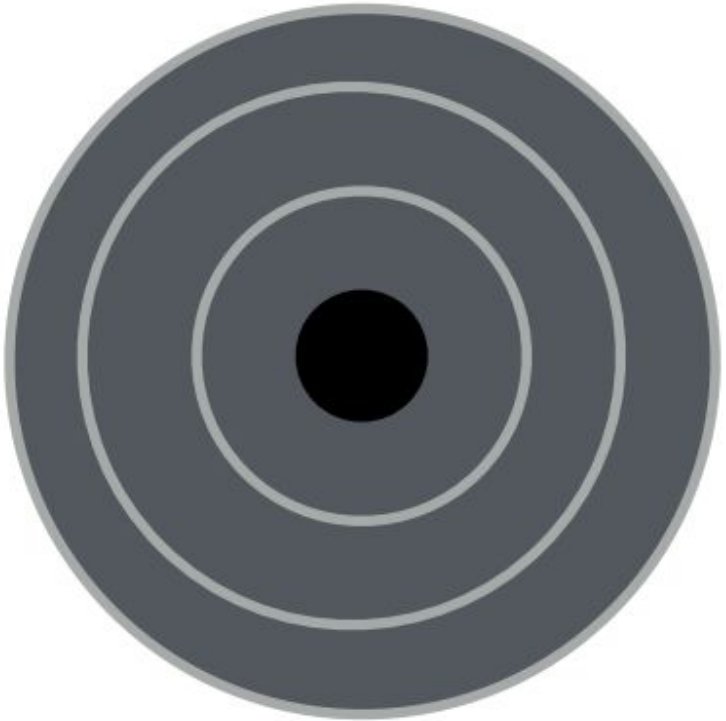
Recap: Hard Disks

- Motor connected to spindle spins platters
- Rate of rotation
  - RPM (rotation per minute)
- 10000 RPM ===>
  - 1 min == 60 sec == 60,000 ms
  - single rotation is 6 ms



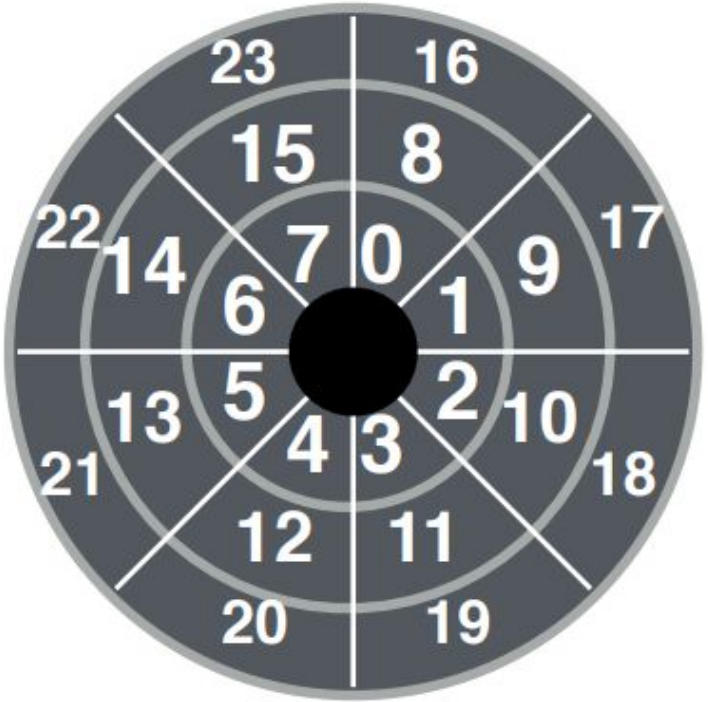
Recap: Hard Disks

- Surface is divided into rings: **tracks**
- Stack of tracks(across platters): **cylinder**



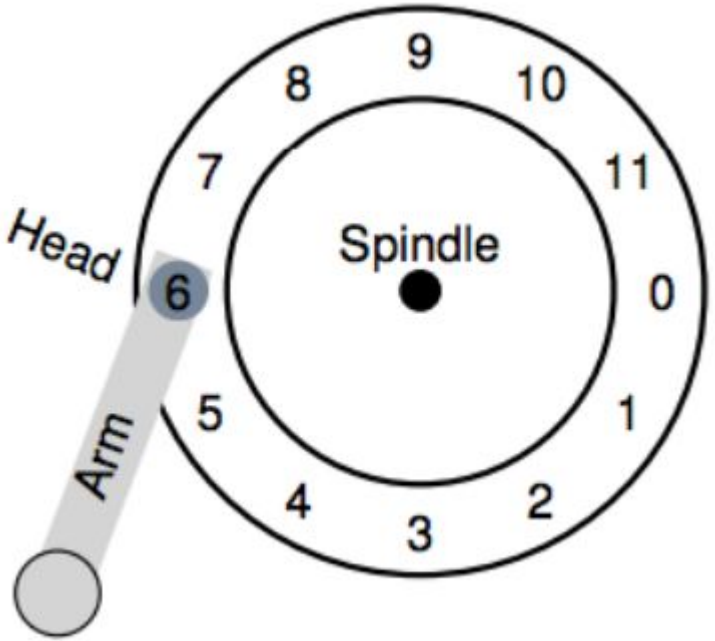
Recap: Hard Disks

- **Tracks** are further divided into numbered **sectors**



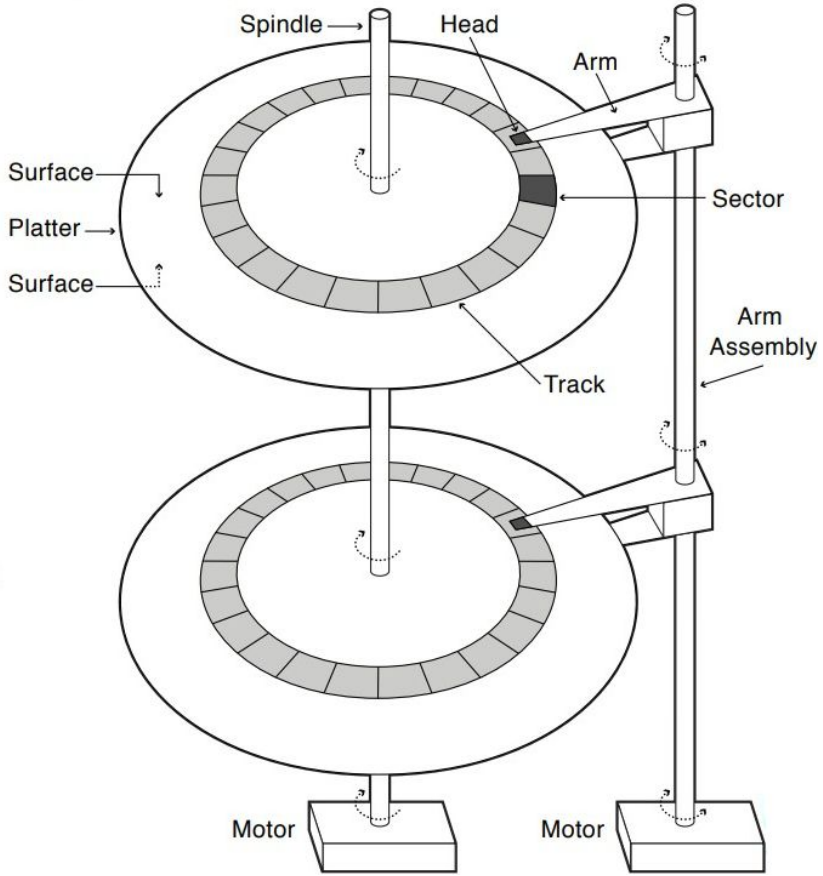
Recap: Hard Disks

- **Heads** on a moving **arm** can read from each surface.



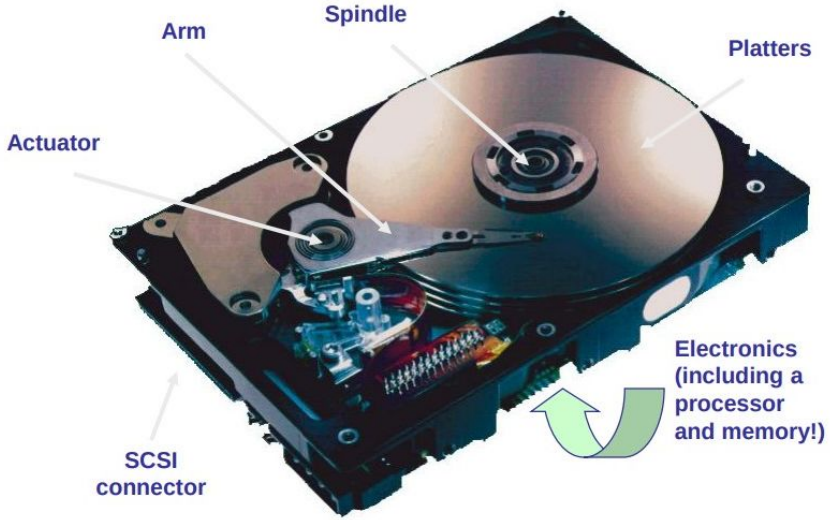
Recap: Hard Disks

- Operations:
  - seek
  - read
  - write
- Must specify
  - cylinder # (distance from spindle)
  - head #
  - sector #
  - transfer size
  - memory address



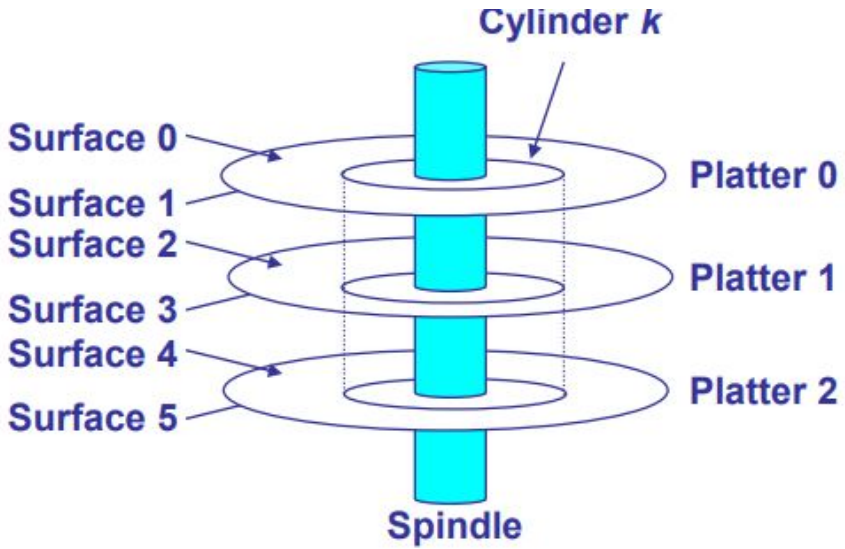
Physical Disk Structure

- Disk components
  - platters
  - surfaces
  - tracks
  - sectors
  - cylinders
  - arm
  - heads

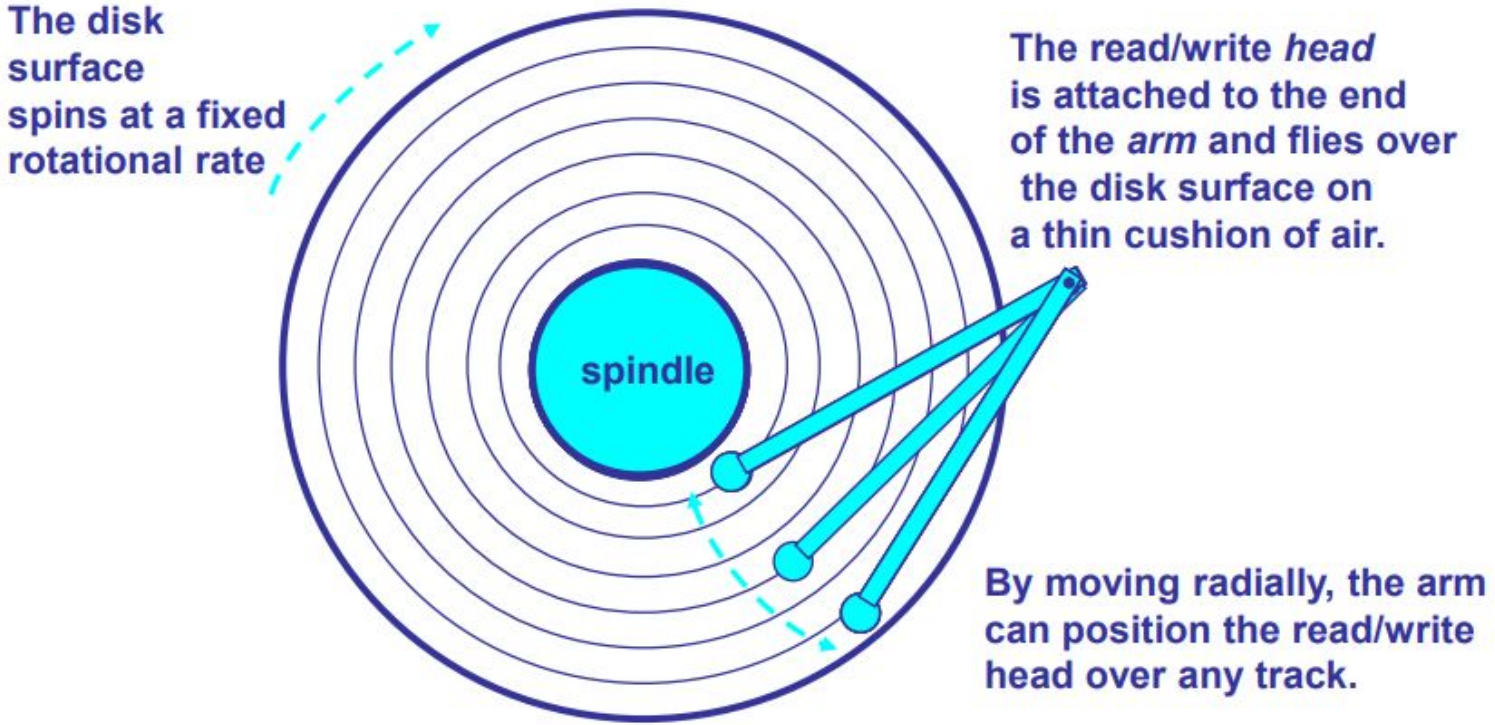


Disk Geometry: Multiple-Platter View

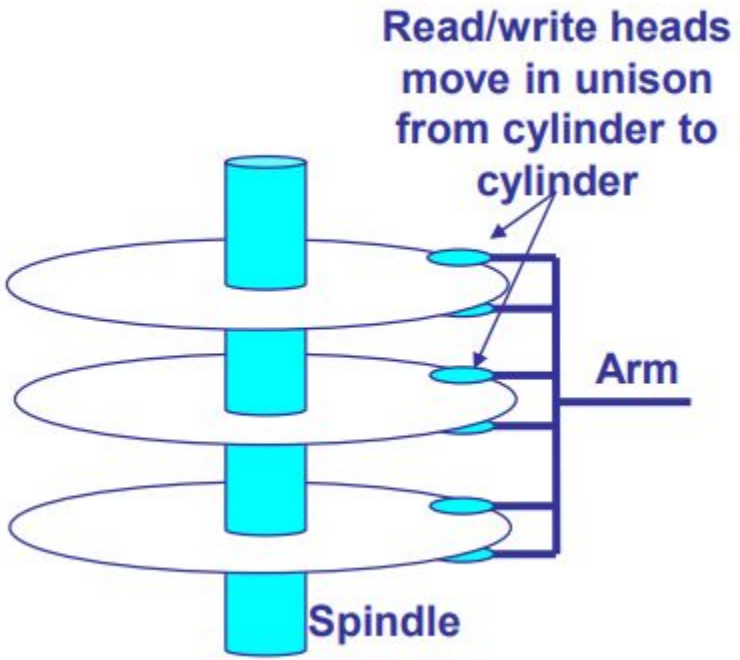
- Aligned tracks form a cylinder.



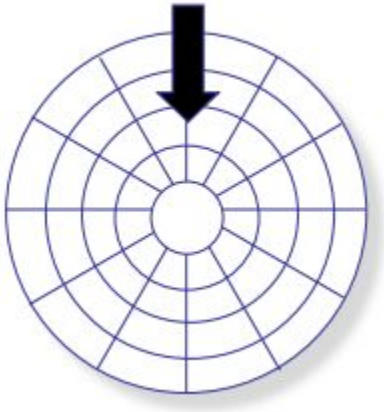
Disk Operation: Single-Platter View



Disk Operation: Multi-Platter View

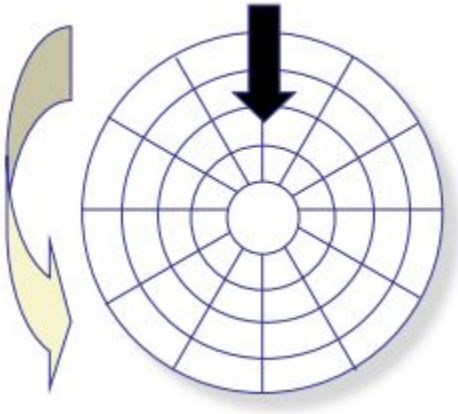


Disk Access



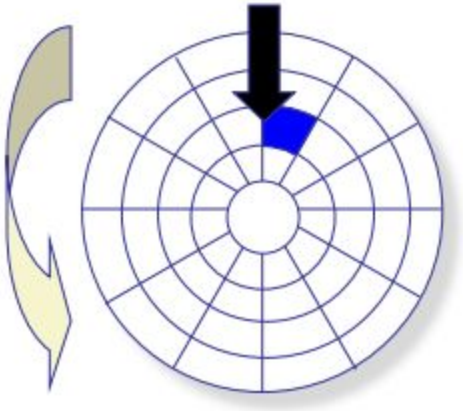
- Head in position above a track

Disk Access



- Rotation is counter-clockwise

Disk Access - Read

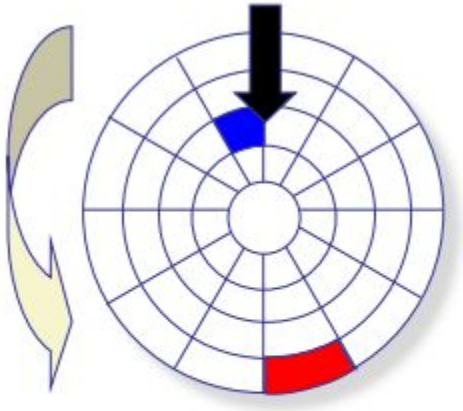


- About to read blue sector

Disk Access - Read

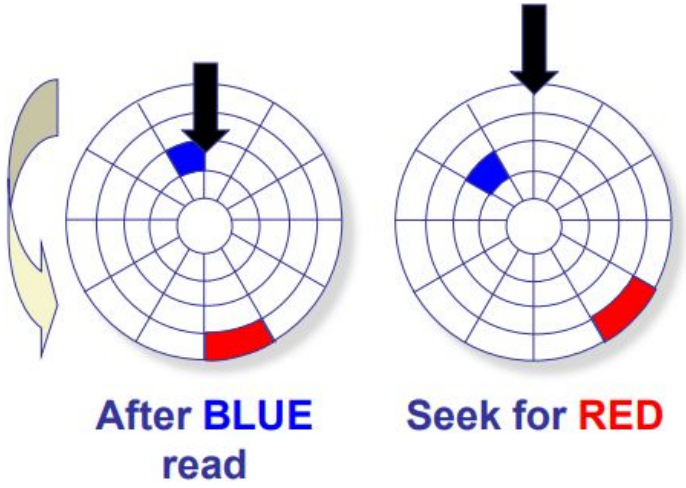
- After reading blue sector

Disk Access - Read



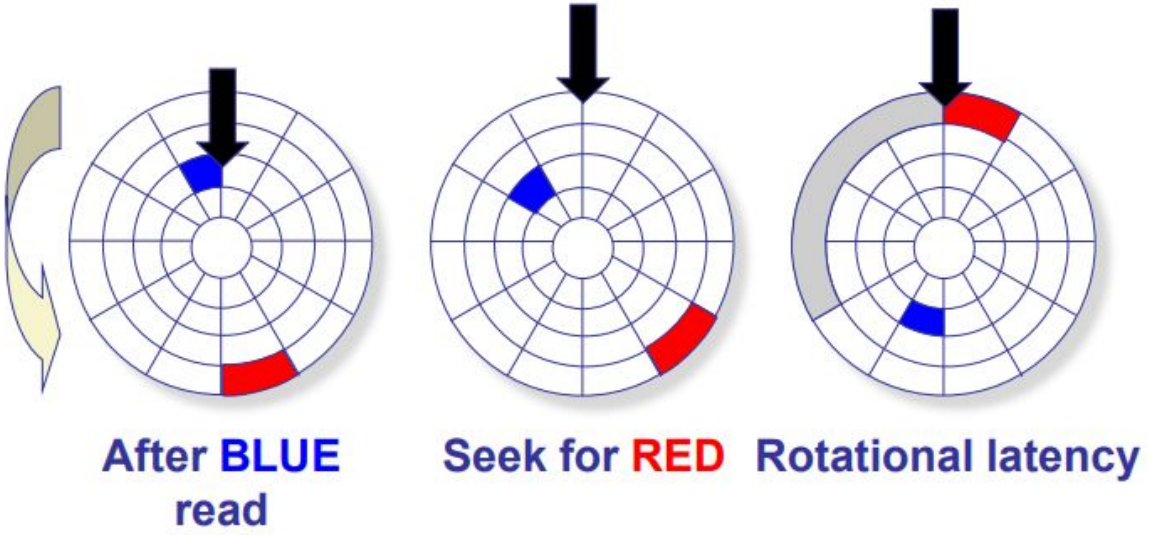
- Red request scheduled next

Disk Access - Seek



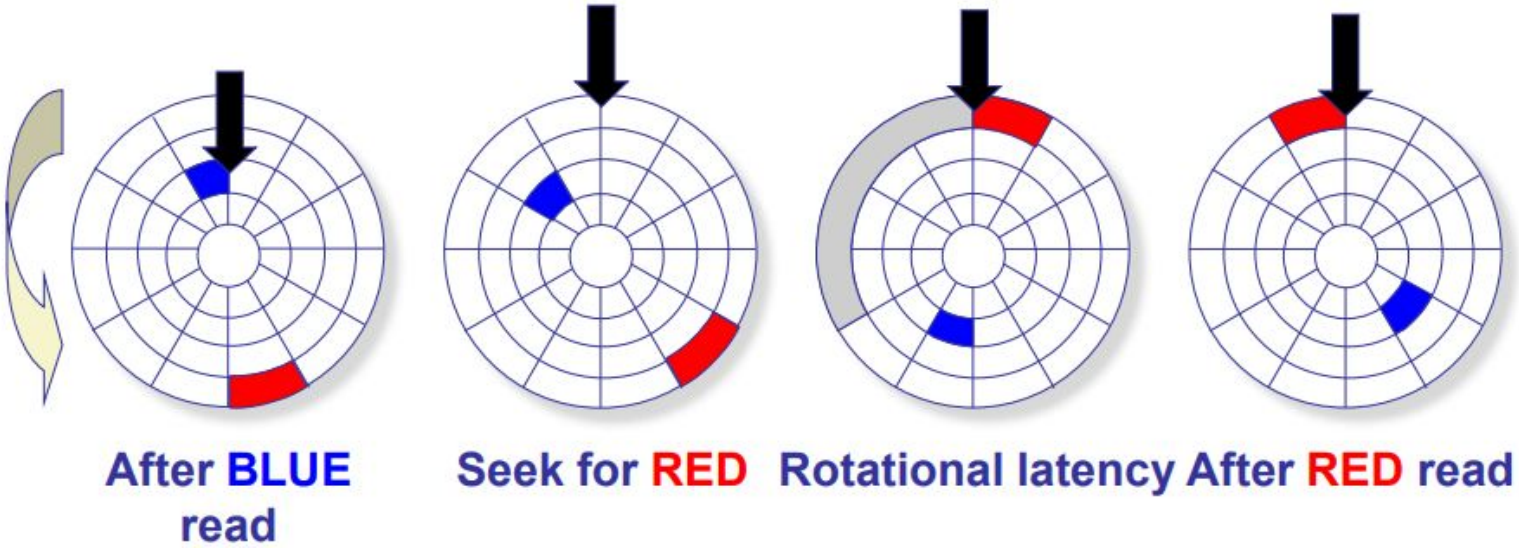
- Seek to red’s track

Disk Access - Rotational Latency



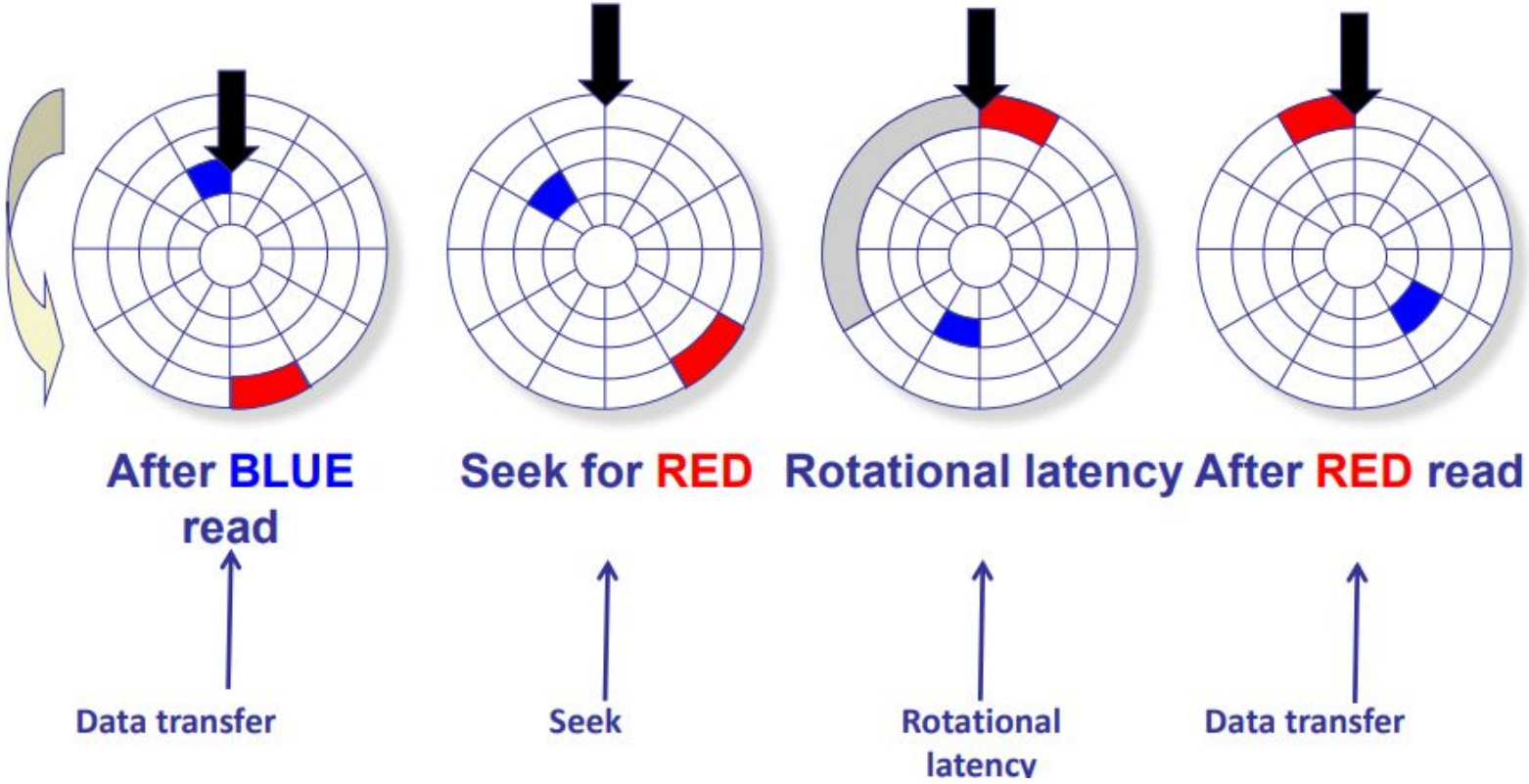
- Wait for red sector to rotate around

Disk Access - Read



- Complete read of red

Disk Access - Service Time Components



Disk Access Time

- Average time to access some target sector approximated by :
  - Taccess = Tavg seek + Tavg rotation + Tavg transfer
- **Seek time** (Tavg seek)
  - Time to position heads over cylinder containing target sector
  - Typical Tavg seek is 3—9 ms
- **Rotational latency** (Tavg rotation)
  - Time waiting for first bit of target sector to pass under r/w head
  - Tavg rotation = 1/2 x 1/RPMs x 60 sec/1 min
- **Transfer time** (Tavg transfer)
  - Time to read the bits in the target sector
  - Tavg transfer = 1/RPM x 1/(avg # sectors/track) x 60 secs/1 min

Disk Access Time

- Given:
  - Rotational rate = 7,200 RPM
  - Average seek time = 9 ms
  - Avg # sectors/track = 400.
- Derived:
  - Tavg rotation = 1/2 x (60 secs/7200 RPM) x 1000 ms/sec = 4 ms
  - Tavg transfer = 60/7200 RPM x 1/400 secs/track x 1000 ms/sec = 0.02 ms
  - Taccess = 9 ms + 4 ms + 0.02 ms

Disk Access Time

- Important points:
  - Access time dominated by seek time and rotational latency.
  - First bit in a sector is the most expensive, the rest are free.
  - DRAM access time is about 60 ns / doubleword
    - 2,500 times slower than DRAM

Disk Access Time

- What is the time for 4KB random read with Cheetah?
- Tseek + Trotation + Ttransfer = 4ms + 2ms + 0.03 ms
- 15000 RPM ⇒ 1 rotation in 4ms ⇒ Trotation = 2ms
- Ttransfer = 4KB/125MB/s = 0.032ms

	Cheetah 15K.5	Barracuda
Capacity	300 GB	1 TB
RPM	15,000	7,200
Average Seek	4 ms	9 ms
Max Transfer	125 MB/s	105 MB/s
Platters	4	4
Cache	16 MB	16/32 MB
Connects via	SCSI	SATA

Disk Access Time

- What is the time for 4KB random read with Barracuda?
- Tseek + Trotation + Ttransfer = 9ms + 4.16ms + 0.038ms
- 7200RPM ⇒ 1 rotation in 8.33ms ⇒ Trotation = 4.16ms
- Ttransfer = 4KB/105MB/s = 0.038ms

	Cheetah 15K.5	Barracuda
Capacity	300 GB	1 TB
RPM	15,000	7,200
Average Seek	4 ms	9 ms
Max Transfer	125 MB/s	105 MB/s
Platters	4	4
Cache	16 MB	16/32 MB
Connects via	SCSI	SATA

Disk Scheduling

- Seeks are so expensive (milliseconds!)
- OS schedules requests that are queued waiting for the disk
  - **FCFS**
  - **SSTF**
  - **SCAN**
  - **C-SCAN**

Disk Scheduling

- **FCFS** (do nothing)
  - Schedule disk operations in order they arrive
  - Reasonable when load is low
- **SSTF** (shortest seek time first)
  - Select request with minimum seek time from current head position
  - A form of Shortest Job First (SJF) scheduling
  - Not optimal:
    - suppose cluster of requests at far end of disk ➜ starvation!

Disk Scheduling

- **SCAN** (Elevator Algorithm)
  - sweep back and forth
  - arm starts at one end of disk
  - moves to other end, servicing requests
  - movement reversed at end of disk
  - repeat
- **C-SCAN**
  - Like SCAN, but only go in one direction

Disk Scheduling

- In general, unless there are request queues, disk scheduling does not have much impact
  - Important for servers, less so for PCs
- Modern disks often do the disk scheduling themselves
  - Disks know their layout better than OS, can optimize better
  - Ignores, undoes any scheduling done by OS

THANK YOU!