I/O Devices - Disk pt. 1(ch. 36+37)

Operating Systems

Based on: Three Easy Pieces by Arpaci-Dusseaux

Moshe Sulamy

Tel-Aviv Academic College

Hard Disk Drives

- Main form of persistent data storage
- File system technology: predicated on their behavior

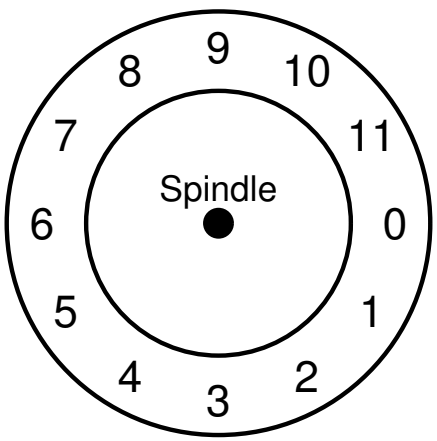
How do modern hard-disk drives store data?
What is the interface?

The Interface

- Consists of sectors (512-byte blocks)
  - Numbered 0 to $n - 1$ (the drive **address space**)
  - Each can be read or written
- Multi-sector operations are possible
  - Many file systems read or write 4KB at a time
  - Only guarantee: single 512-byte block write is **atomic**
  - i.e., will completely entirely or not at all
  - **Torn write**: only portion of a larger write complete
- Common assumption: sequential access is the fastest

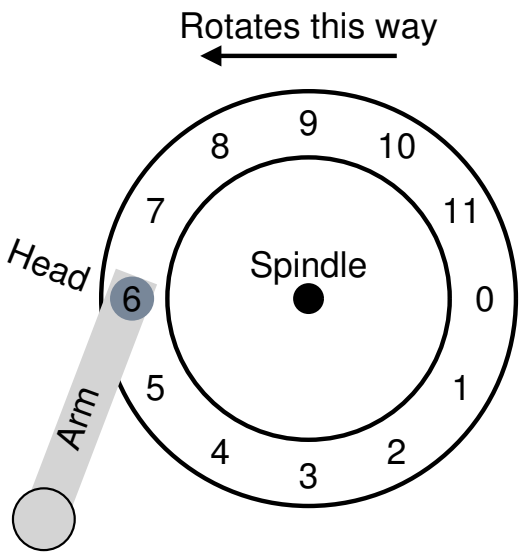
Basic Geometry

- A **platter**
  - Circular surface on which data is stored
  - Two sides, each called a **surface**
- A disk has one or more platters
  - Bound together around the **spindle**
    - Connected to a motor that spins the platters
    - Fixed rate of **rotations per minute (RPM)**

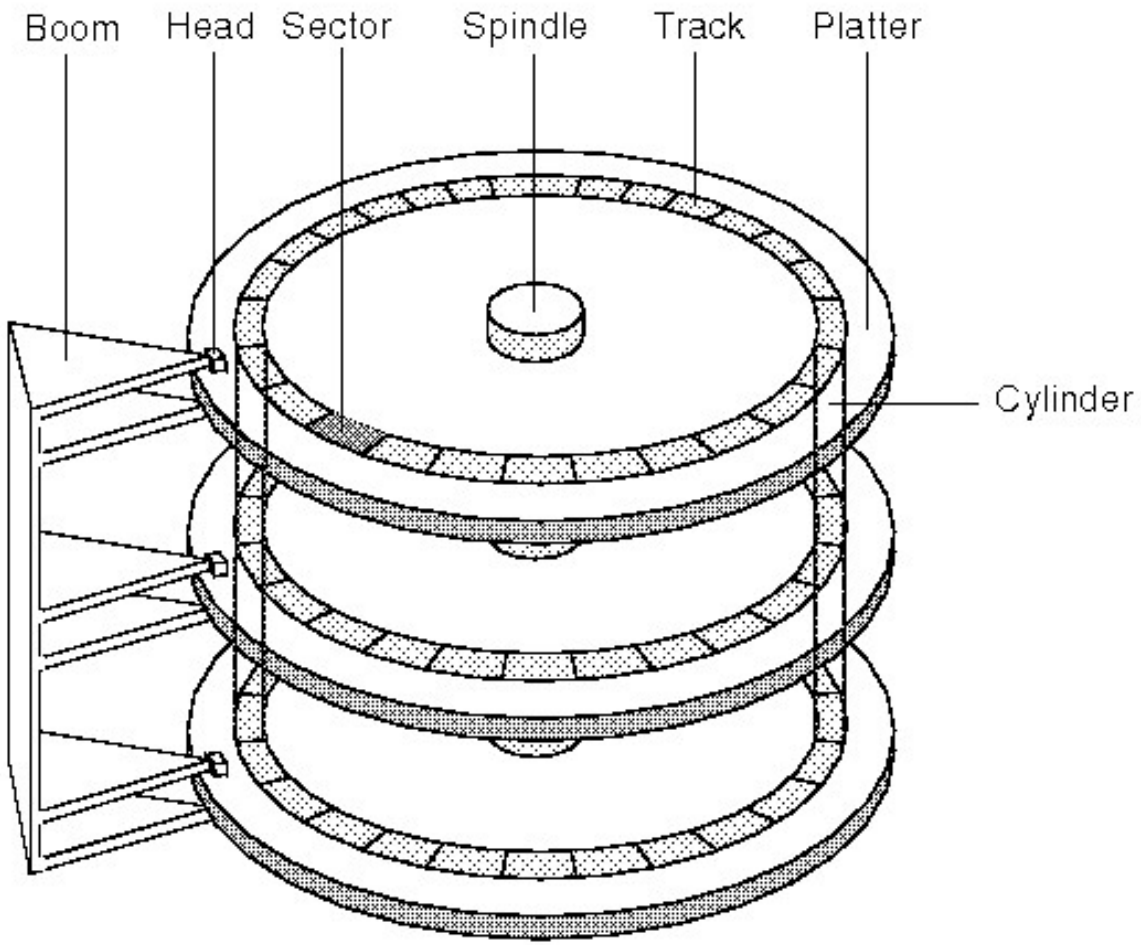


Basic Geometry

- Data is encoded in **tracks**
  - Concentric circles of sectors
  - Single surface contains thousands of tracks
- Read and write accomplished by **disk head**
  - One per surface
  - Attached to a **disk arm**
    - Moves across surface

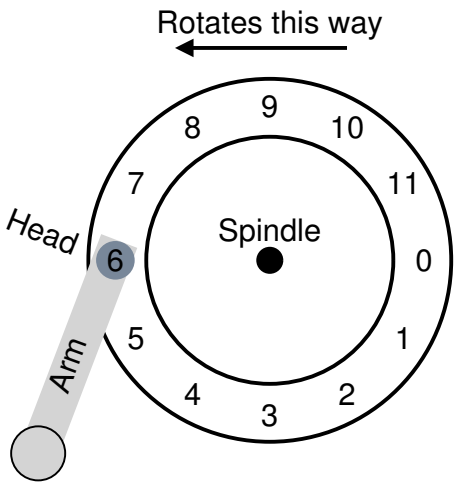


Basic Geometry



Single-track Latency

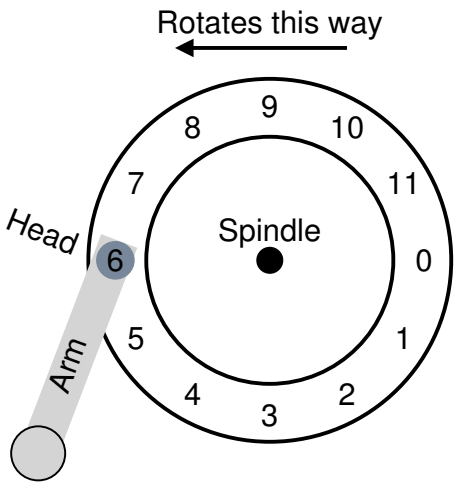
- Single track, with 12 sectors
- **Rotational delay**: wait for desired sector to reach disk head:



- Full rotational delay is R
  - Wait for sector 0?

Single-track Latency

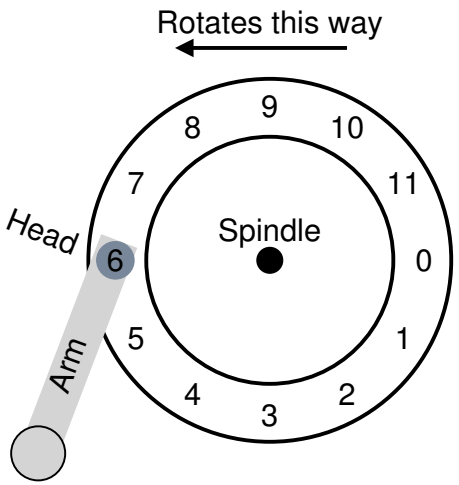
- Single track, with 12 sectors
- **Rotational delay**: wait for desired sector to reach disk head:



- Full rotational delay is R
  - Wait for sector 0? $\frac{R}{2}$
  - Worst-case request?

Single-track Latency

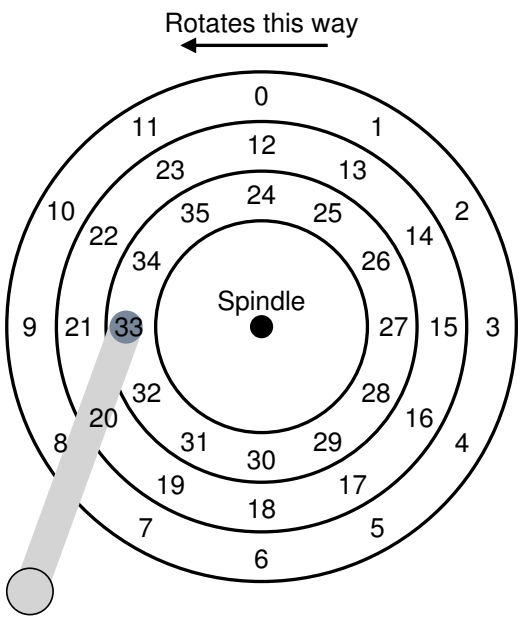
- Single track, with 12 sectors
- **Rotational delay**: wait for desired sector to reach disk head:

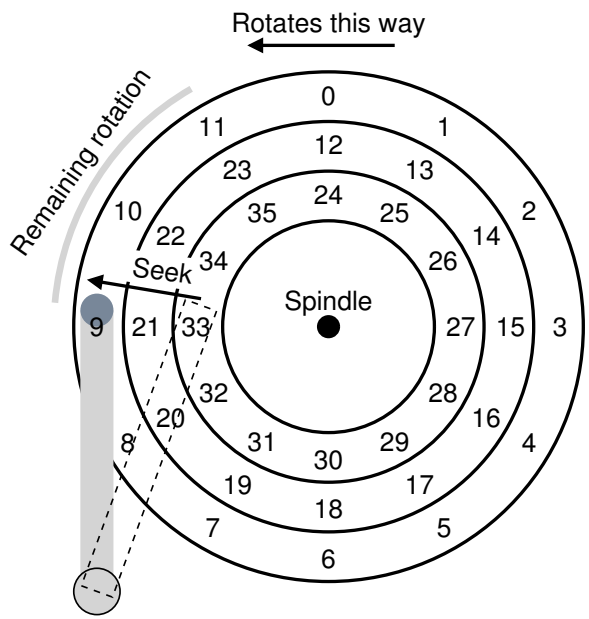


- Full rotational delay is R
  - Wait for sector 0? $\frac{R}{2}$
  - Worst-case request? sector 5 ($\frac{11R}{12}$)

Multiple Tracks

- **Seek**: move disk arm to the correct track
  - Costly disk operation, along with rotation
  - Acceleration: disk arm gets moving
  - Coasting: moving at full speed
  - Deceleration: arm slows down
  - Settling: head carefully positioned over correct track
  - **Settling time**: often quite significant, e.g., 0.5 to 2 ms



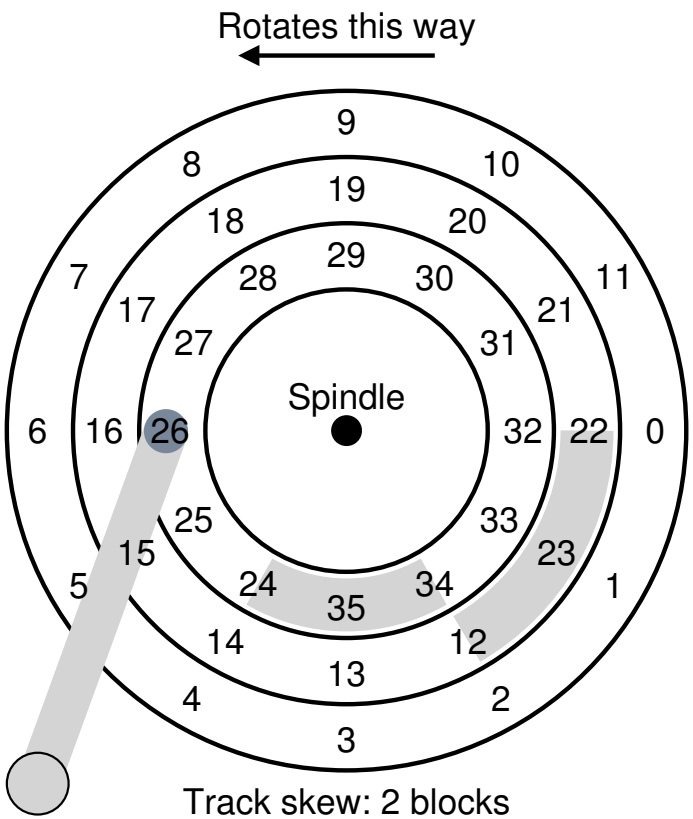


I/O Time

- Seek
- Wait for rotational delay
- **Transfer**: data is read from or written to surface

Other Details

- **Track skew**
  - Switching tracks → time to reposition the head
  - Without skew, desired next block already rotated → have to wait almost entire rotational delay



Other Details

- **Multi-zoned** disk drives
  - Outer tracks have more sectors than inner tracks
  - Disk is organized into multiple zones
  - Each zone has the same number of sectors per track
  - Outer zones have more sectors than inner zones

Other Details

- **Cache**
  - Hold data read from or written to disk (8 to 64 MB)
  - Quickly respond to requests
  - e.g., read all sectors on a track and cache in memory
  - **Write-through**
    - Acknowledge write when it's written to disk
  - **Writeback**
    - Acknowledge write when data is in cache
    - Faster but dangerous: consistency issues (order not guaranteed)

I/O Time

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

- Rate of I/O: $R_{I/O} = \frac{Size_{transfer}}{T_{I/O}}$

- **Random** workload
  - Small 4KB reads to random locations
- **Sequential** workload
  - Read 100MB of consecutive sectors

I/O Time

- Two example modern disks (Seagate):

	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

I/O Time

	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

- On the Cheetah:
  - $T_{seek} =$
  - $T_{rotation} =$
  - Random $T_{transfer} =$
  - Seq. $T_{transfer} =$
- On the Barracuda:
  - $T_{seek} =$
  - $T_{rotation} =$
  - Random $T_{transfer} =$
  - Seq. $T_{transfer} =$

I/O Time

	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

- On the Cheetah:
  - $T_{seek} = 4ms$
  - $T_{rotation} =$
  - Random $T_{transfer} =$
  - Seq. $T_{transfer} =$
- On the Barracuda:
  - $T_{seek} = 9ms$
  - $T_{rotation} =$
  - Random $T_{transfer} =$
  - Seq. $T_{transfer} =$

I/O Time

	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

- On the Cheetah:
  - $T_{seek} = 4ms$
  - $T_{rotation} = 2ms$
  - Random $T_{transfer} =$
  - Seq. $T_{transfer} =$
- On the Barracuda:
  - $T_{seek} = 9ms$
  - $T_{rotation} = 4.2ms$
  - Random $T_{transfer} =$
  - Seq. $T_{transfer} =$

I/O Time

	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

- On the Cheetah:
  - $T_{seek} = 4ms$
  - $T_{rotation} = 2ms$
  - Random $T_{transfer} = 30\mu s$
  - Seq. $T_{transfer} =$
- On the Barracuda:
  - $T_{seek} = 9ms$
  - $T_{rotation} = 4.2ms$
  - Random $T_{transfer} = 38\mu s$
  - Seq. $T_{transfer} =$

I/O Time

	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

- On the Cheetah:
  - $T_{seek} = 4ms$
  - $T_{rotation} = 2ms$
  - Random $T_{transfer} = 30\mu s$
  - Seq. $T_{transfer} = 800ms$
- On the Barracuda:
  - $T_{seek} = 9ms$
  - $T_{rotation} = 4.2ms$
  - Random $T_{transfer} = 38\mu s$
  - Seq. $T_{transfer} = 950ms$

I/O Time

- On the Cheetah:
  - $T_{seek} = 4ms$
  - $T_{rotation} = 2ms$
  - Random $T_{transfer} = 30\mu s$
  - Seq. $T_{transfer} = 800ms$
- On the Barracuda:
  - $T_{seek} = 9ms$
  - $T_{rotation} = 4.2ms$
  - Random $T_{transfer} = 38\mu s$
  - Seq. $T_{transfer} = 950ms$

	Cheetah 15K.5	Barracuda
$T_{I/O}$ Random		
$R_{I/O}$ Random		
$T_{I/O}$ Sequential		
$R_{I/O}$ Sequential		

I/O Time

- On the Cheetah:
  - $T_{seek} = 4ms$
  - $T_{rotation} = 2ms$
  - Random $T_{transfer} = 30\mu s$
  - Seq. $T_{transfer} = 800ms$
- On the Barracuda:
  - $T_{seek} = 9ms$
  - $T_{rotation} = 4.2ms$
  - Random $T_{transfer} = 38\mu s$
  - Seq. $T_{transfer} = 950ms$

	Cheetah 15K.5	Barracuda
$T_{I/O}$ Random	6 ms	13.2 ms
$R_{I/O}$ Random		
$T_{I/O}$ Sequential		
$R_{I/O}$ Sequential		

I/O Time

- On the Cheetah:
  - $T_{seek} = 4ms$
  - $T_{rotation} = 2ms$
  - Random $T_{transfer} = 30\mu s$
  - Seq. $T_{transfer} = 800ms$
- On the Barracuda:
  - $T_{seek} = 9ms$
  - $T_{rotation} = 4.2ms$
  - Random $T_{transfer} = 38\mu s$
  - Seq. $T_{transfer} = 950ms$

	Cheetah 15K.5	Barracuda
$T_{I/O}$ Random	6 ms	13.2 ms
$R_{I/O}$ Random	0.66 MB/s	0.31 MB/s
$T_{I/O}$ Sequential		
$R_{I/O}$ Sequential		

I/O Time

- On the Cheetah:
  - $T_{seek} = 4ms$
  - $T_{rotation} = 2ms$
  - Random $T_{transfer} = 30\mu s$
  - Seq. $T_{transfer} = 800ms$
- On the Barracuda:
  - $T_{seek} = 9ms$
  - $T_{rotation} = 4.2ms$
  - Random $T_{transfer} = 38\mu s$
  - Seq. $T_{transfer} = 950ms$

	Cheetah 15K.5	Barracuda
$T_{I/O}$ Random	6 ms	13.2 ms
$R_{I/O}$ Random	0.66 MB/s	0.31 MB/s
$T_{I/O}$ Sequential	806 ms	963 ms
$R_{I/O}$ Sequential		

I/O Time

- On the Cheetah:
  - $T_{seek} = 4ms$
  - $T_{rotation} = 2ms$
  - Random $T_{transfer} = 30\mu s$
  - Seq. $T_{transfer} = 800ms$
- On the Barracuda:
  - $T_{seek} = 9ms$
  - $T_{rotation} = 4.2ms$
  - Random $T_{transfer} = 38\mu s$
  - Seq. $T_{transfer} = 950ms$

	Cheetah 15K.5	Barracuda
$T_{I/O}$ Random	6 ms	13.2 ms
$R_{I/O}$ Random	0.66 MB/s	0.31 MB/s
$T_{I/O}$ Sequential	806 ms	963 ms
$R_{I/O}$ Sequential	125 MB/s	105 MB/s

I/O Time

- On the Cheetah:
  - $T_{seek} = 4ms$
  - $T_{rotation} = 2ms$
  - Random $T_{transfer} = 30\mu s$
  - Seq. $T_{transfer} = 800ms$
- On the Barracuda:
  - $T_{seek} = 9ms$
  - $T_{rotation} = 4.2ms$
  - Random $T_{transfer} = 38\mu s$
  - Seq. $T_{transfer} = 950ms$

	Cheetah 15K.5	Barracuda
$T_{I/O}$ Random	6 ms	13.2 ms
$R_{I/O}$ Random	0.66 MB/s	0.31 MB/s
$T_{I/O}$ Sequential	806 ms	963 ms
$R_{I/O}$ Sequential	125 MB/s	105 MB/s

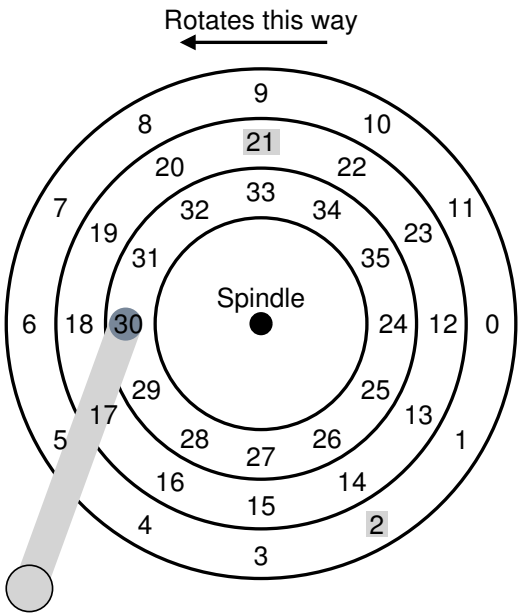
- **Use disks sequentially!**

Disk Scheduling

- **Disk scheduler**
  - OS examines requests and decides which to schedule next
  - Can make a good guess how long a job will take
    - By estimating seek and rotation delay
  - Greedily pick least time to service first

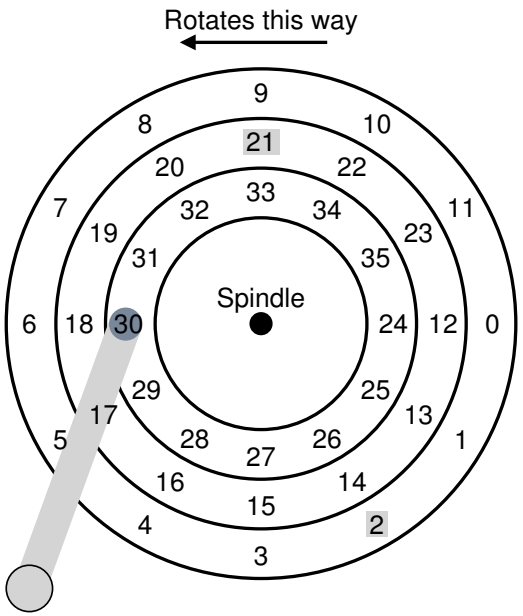
Disk Scheduling

- **Shortest Seek Time First (SSTF)**
  - Order I/O requests by track
  - Pick request on nearest track



Disk Scheduling

- **Shortest Seek Time First (SSTF)**
  - Order I/O requests by track
  - Pick request on nearest track



- Issue request to 21, then issue request to 2

Disk Scheduling

- Drive geometry not available to host OS
  - Sees an array of blocks
  - Solution?

Disk Scheduling

- Drive geometry not available to host OS
  - Sees an array of blocks
  - Solution? implement **nearest-block-first (NBF)**
- More fundamental problem?

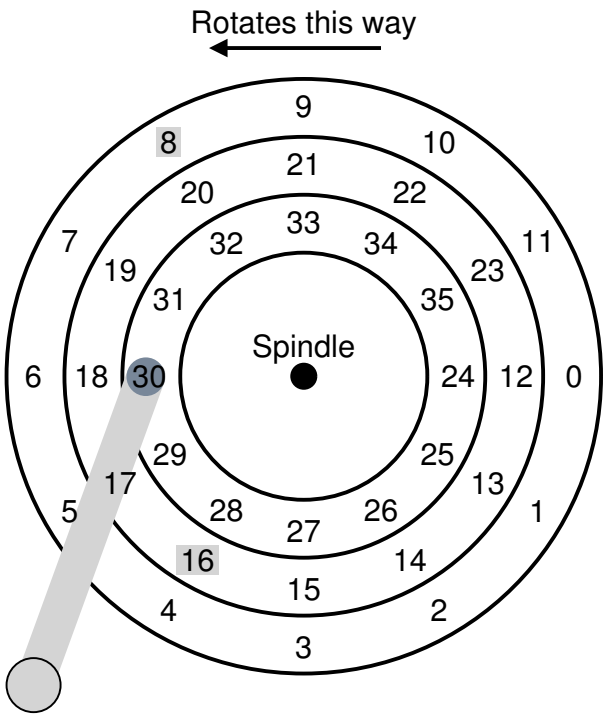
Disk Scheduling

- Drive geometry not available to host OS
  - Sees an array of blocks
  - Solution? implement **nearest-block-first (NBF)**
- More fundamental problem?
- **Starvation**
  - Steady stream of requests to inner track
  - Other tracks ignored completely

Disk Scheduling

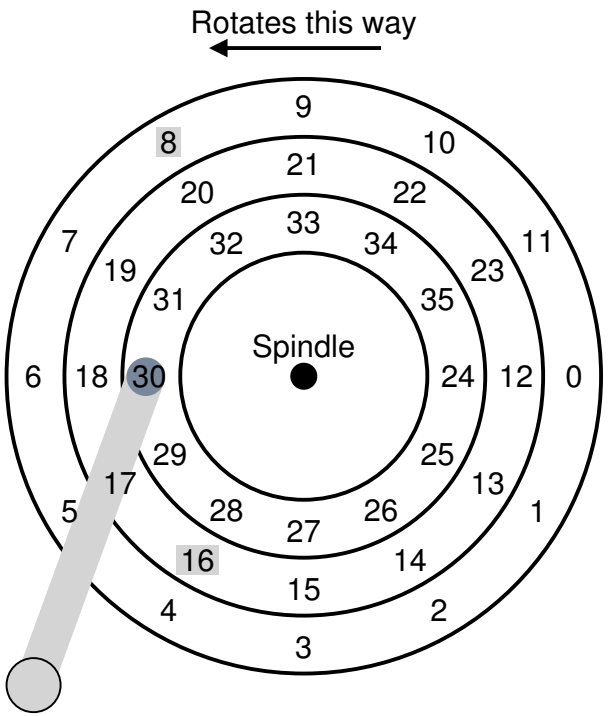
- **Elevator**
  - Service requests in order across the tracks (back and forth)
  - **Sweep**: single pass across the disk
    - Request of already-serviced track is queued until the next sweep
  - **F-SCAN**: freeze queue when doing a sweep
    - Prevents starvation of far-away requests
  - **C-SCAN**: sweep from outer-to-inner
    - Resets at outer track to begin again
    - Instead of both directions (favors middle tracks)
- Still problematic: ignores rotation

Disk Scheduling



- Schedule sector 16 or sector 8 next?

Disk Scheduling



- Schedule sector 16 or sector 8 next? it depends
  - Seek time much higher than rotational delay $\rightarrow$ SSTF
  - Seek faster than rotation $\rightarrow$ service request 8

Disk Scheduling

- Modern drives: seek and rotation times roughly equivalent
- **Shortest Positioning Time First (SPTF)**
  - Difficult to implement in OS
  - Usually performed inside a drive
  - OS picks best few requests and issues all to disk
- **I/O merging**
  - Series of requests - sectors 33, 8, then 34
  - OS merges 33 and 34 into a single two-block request
- **Work-conserving**
  - Wait before issuing I/O to disk
  - New and "better" request may arrive

Summary (Hard Disk Drives)

- 512-byte sectors
  - **Platter** with two **surfaces**, bound around the **spindle**
  - Fixed rate of **RPM**
  - Data encoded in **tracks**, read and write by **disk head**
- **Rotational delay**: wait for sector to reach head
- **Seek**: move disk arm to correct track
  - Acceleration $\rightarrow$ coasting $\rightarrow$ deceleration $\rightarrow$ settling
- **I/O time**: seek $\rightarrow$ wait for rotational delay $\rightarrow$ transfer
- **Cache** holds read/write data
  - **Write-through**: acknowledge on write to disk
  - **Writeback**: acknowledge when data is in cache
- Disk scheduling
  - **SSTF**, **NBF**, **Elevator** (**sweep**, **F-SCAN**, **C-SCAN**), **SPTF**
  - **I/O merging**: merge requests for consecutive sectors
  - **Work-conserving**: wait before issuing I/O to disk