Lecture 18: Disk Scheduling

Operating Systems

Content taken from: https://pages.cs.wisc.edu/~remzi/OSTEP/

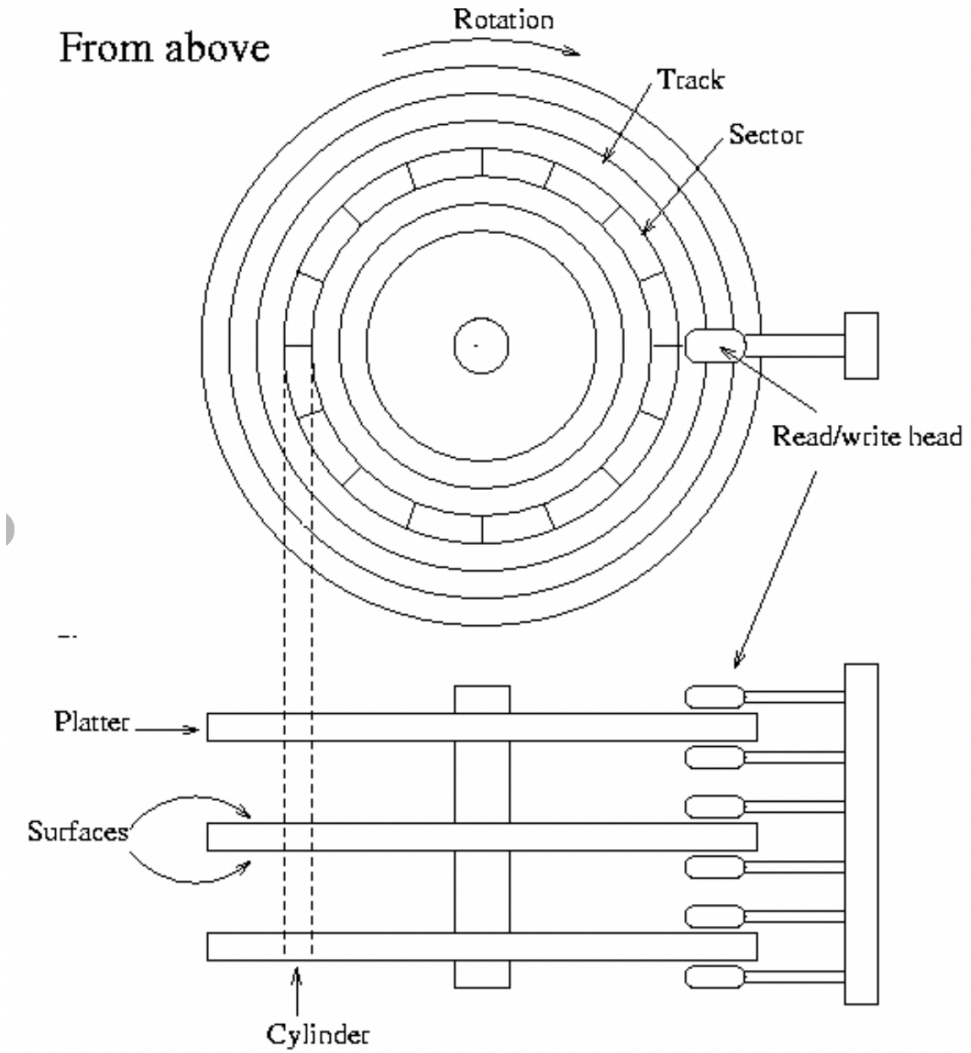
https://www.cse.iitb.ac.in/~mythili/os/

Last Lecture

- Simple/Canonical Device Model
- Canonical I/O Protocol
- Lower CPU overhead with Interrupts
- Use DMA for direct data transfer from memory to I/O device
- Device drivers
- Hard Disk

Hard Disk

- Interface: a set of 512-byte blocks (sectors), that can be read or written atomically
  - Sectors are numbered from 0 to N-1
- Internals: one or more circular platters, connected by a spindle, spinning at ~10K RPM (rotations per minute)
- Each platter has a disk head and arm
- A platter is divided into multiple tracks, and each track into 512-byte sectors

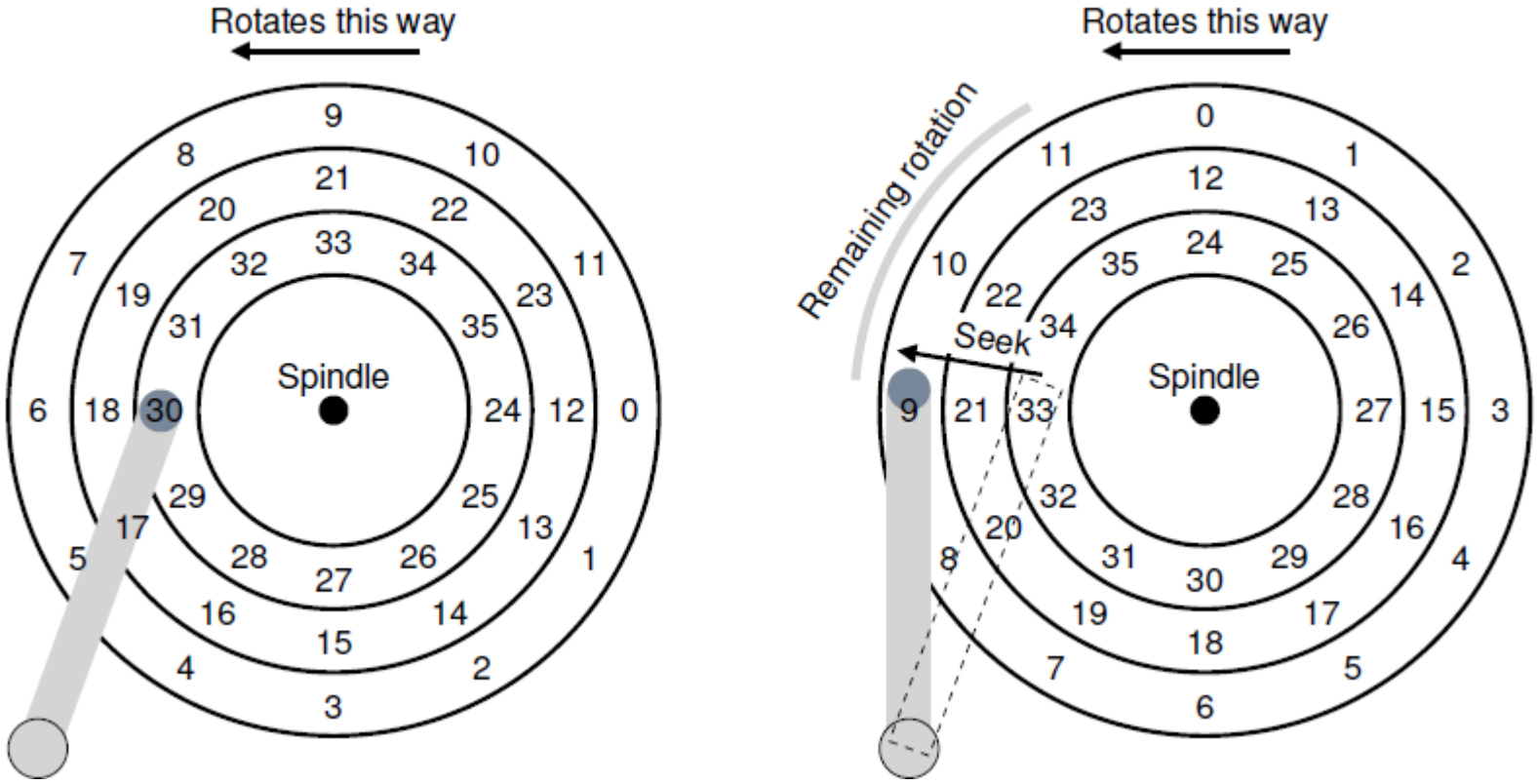


Top and Side View of Hard-disk

https://www.researchgate.net/figure/Top-and-Side-View-of-Hard-disk_fig1_334274392

Accessing a particular sector

- Suppose disk head at 30, need to access 11
- Seek to the correct track, wait for disk to rotate



Time Taken for I/O operation

- Time taken to read/write a block consists of
  - Seek time to get to the right track (few ms)
  - Rotational latency for disk to spin to correct sector on the track (few ms)
  - Transfer time to read sector (few tens microsec)

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

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

Let us solve a simple numerical

- Compute the rate of data transfer for each of the given disks for the following workloads:
  - **Random workload:** Issues small (**4 KB**) reads to random locations on the disk
  - **Sequential workload:** Reads **100 MB** of data in sequence (reads a large number of sectors consecutively from the disk without jumping around)

	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

Figure 37.5: **Disk Drive Specs: SCSI Versus SATA**

	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

Figure 37.5: **Disk Drive Specs: SCSI Versus SATA**

	Cheetah	Barracuda
$R_{I/O}$ Random	0.66 MB/s	0.31 MB/s
$R_{I/O}$ Sequential	125 MB/s	105 MB/s

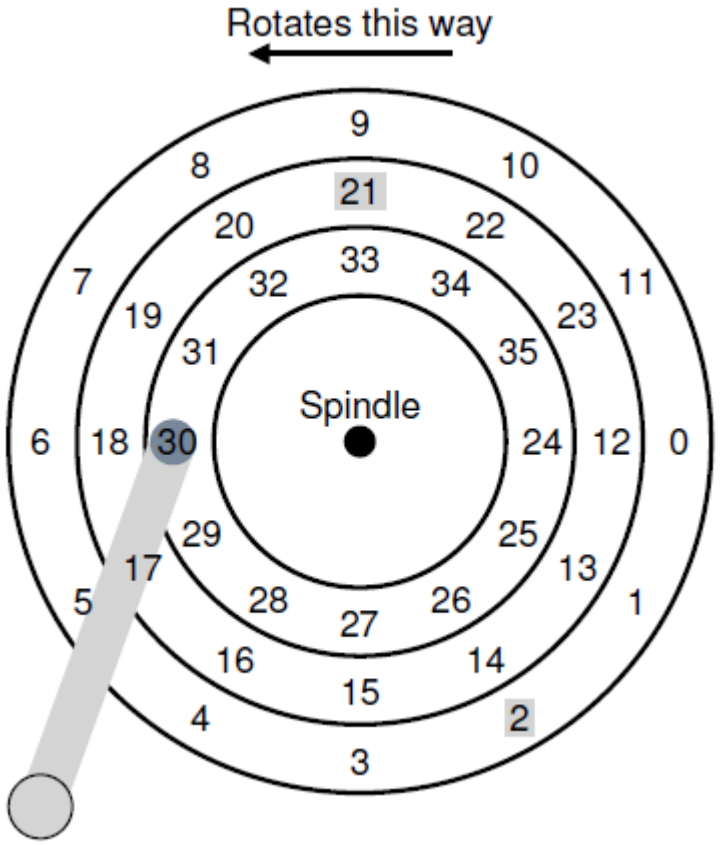
Figure 37.6: **Disk Drive Performance: SCSI Versus SATA**

Disk Scheduling

- Requests to disk are not served in FIFO, they are reordered with other pending requests
- Why? In order to read blocks in sequence as far as possible, to minimize seek time and rotational delay
- Disk scheduler will try to follow the principles of SJF (Shortest Job First)

Shortest Seek Time First (SSTF)

- Access block that we can seek to fastest
  - Go to 21 (one track away) before 2 (two tracks away)
- Problems:
  - Drive Geometry is not available to OS
    - OS implements **nearest-block-first (NBF)**
  - Starvation
    - Some requests that are far from current position of head may never get served

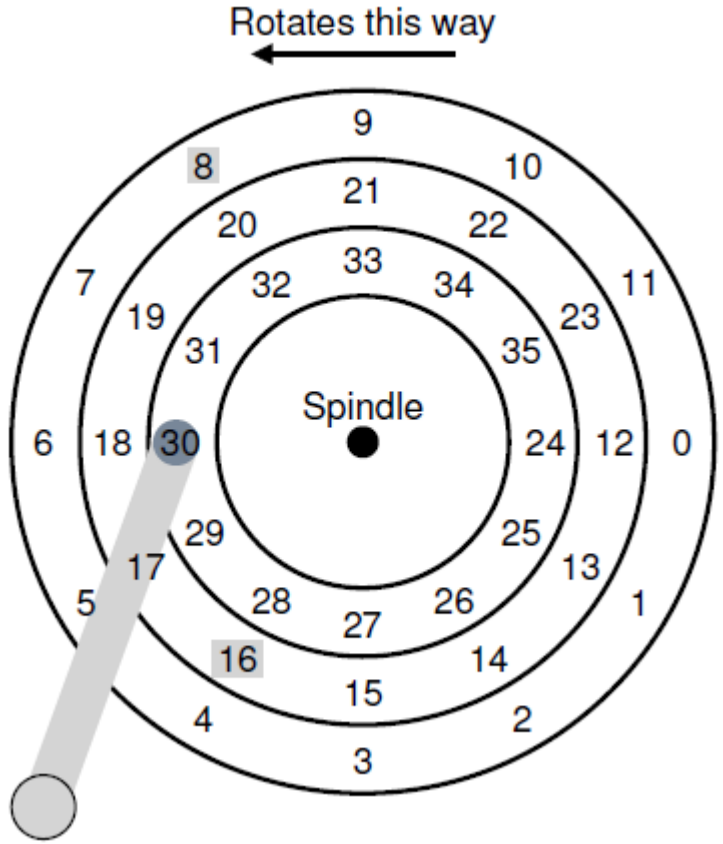


Elevator / SCAN

- Disk head does one sweep over tracks and serves requests that fall on the path
- Elevator/SCAN: sweep outer to inner, then inner to outer
- C-SCAN: sweep only one direction (say, outer to inner) and circle back, start again
  - Why? Sweeping back and forth favours middle tracks more
- F-SCAN: freeze queue while scanning
  - Why? Avoid starving far away requests

Shortest Positioning Time First (SPTF)

- Considers both seek time and rotational latency
  - Better to serve 8 before 16, even though seek time is higher
  - Why? 16 incurs a much higher rotational latency
- Problems:
  - OS does not have idea about the disk geometry



Other Scheduling Issues

- Where is disk scheduling performed on modern systems?
  - OS does some initial scheduling and sends the “best” few requests to the disk.
  - Disk then uses its internal knowledge of head position and detailed track layout information to service said requests using SPTF
- I/O merging
  - Imagine a series of requests to read blocks 33, then 8 and then 34.
  - OS can merge the requests for block 33 and 34 into a single two-block request to reduce overheads.
- How long should the system wait before issuing an I/O to disk?
  - Immediately issue the request to drive (work conserving)
  - Wait for a bit (non-work-conserving)