Universidade Estadual de Campinas
Instituto de Computação

MC504 Sistemas Operacionais

T26 Hard Disk Drives

Referência principal
Ch.37 of *Operating Systems: Three Easy Pieces* by Remzi and Andrea Arpaci-Dusseau (pages.cs.wisc.edu/~remzi/OSTEP/)

Discutido em classe em 29 de outubro de 2018

Arthur João Catto, PhD

2º semestre de 2018

Hard disk drives have been the main form of persistent data storage in computer systems for decades. . .

- How do modern hard disks store data?
- What is their interface?
- How is data actually laid out and accessed?
- How does disk scheduling affect performance?

The hard-disk drive interface

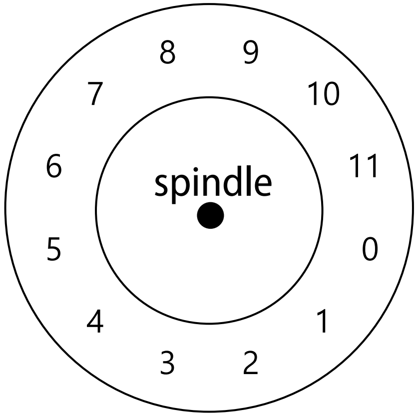
- The basic interface for all modern drives is straightforward.
- The drive consists of a large number of sectors (512-byte blocks), each of which can be read or written.
- The sectors are numbered from $\mathbf{0}$ to $n-1$ on a disk with n sectors.
- Thus, we can view the disk as an array of sectors; $\mathbf{0}$ to $n-1$ is the **address space** of the drive.

The hard-disk drive interface

- Multi-sector operations are possible.
  - Many file systems will read or write 4KB or more at a time.
  - However, the only guarantee is that a single 512- byte write is **atomic** (i.e., it will either complete in its entirety or it won't complete at all).
  - Thus, if an untimely power loss occurs, only a portion of a larger write may complete (sometimes called a **torn write**).
- There are some assumptions most clients of disk drives make, but that are not specified directly in the interface (an **unwritten contract**):
  - Accessing two blocks that are near one-another within the drive's address space is faster than accessing two blocks that are far apart.
  - Accessing blocks in a contiguous chunk (i.e., a sequential read or write) is the fastest access mode, much faster than any random access pattern.

The basic geometry of a hard disk

- Platter
  - A circular hard surface (aluminum coated with a thin magnetic layer)
  - Data is stored persistently by inducing magnetic changes to it.
  - Each platter has 2 sides, each of which is called a **surface**.

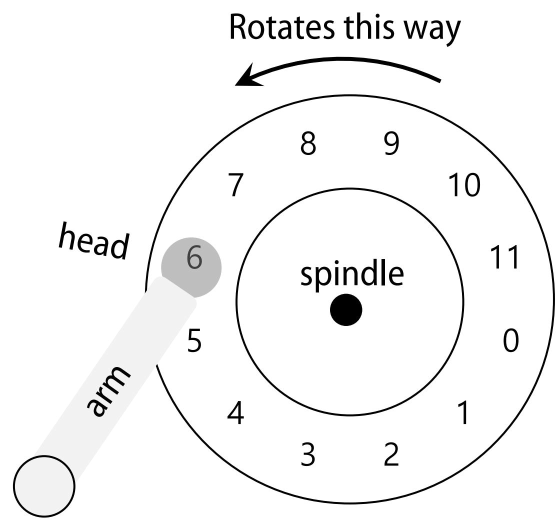


A Disk with Just One Single Track (12 sectors)

- Spindle
  - Spindle is connected to a motor that spins the platters around.
  - The rate of rotation today is typically 7,200 to 15,000 RPM.
  - At 10,000 RPM a full rotation takes about 6 ms.
- Track
  - Concentric circles of sectors
  - Data is encoded on each surface in a track.
  - A single surface contains many thousands and thousands of tracks.

The basic geometry of a hard disk

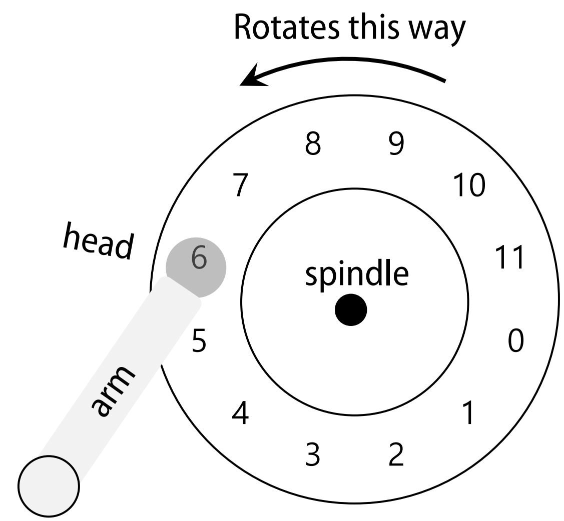
- Disk head
  - There is one head per surface of the drive
  - It is responsible for *reading* and *writing* the disk.
  - It is attached to a single **disk arm**, which moves across the surface.



A Single Track Plus A Head

Single track latency: Rotational Delay

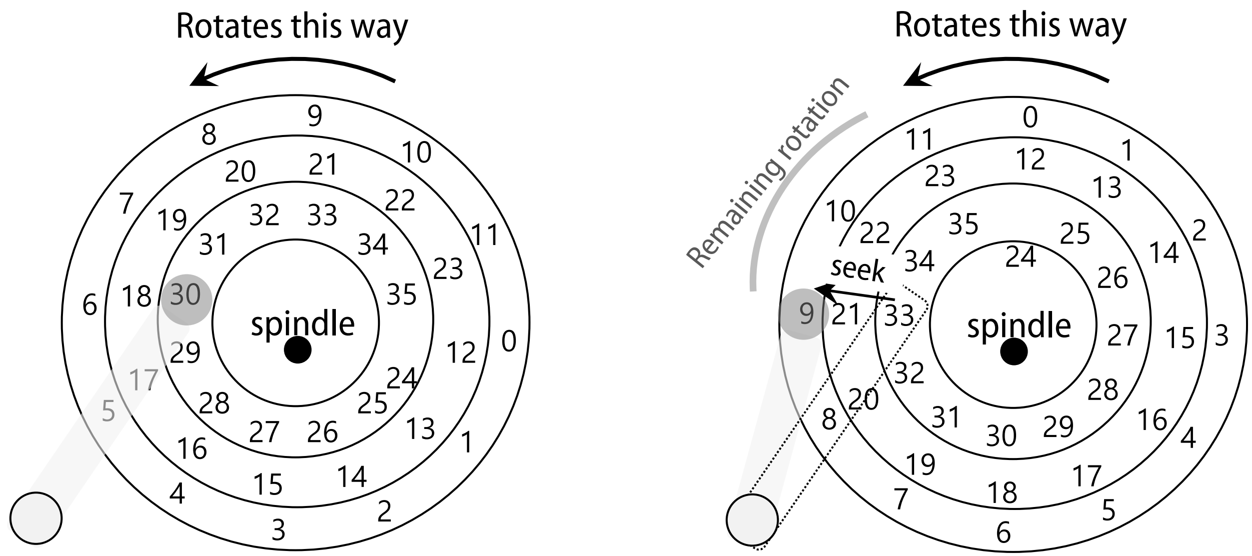
- Rotational delay
  - Time for the desired sector to reach the disk head
  - For example, if full rotational delay is R and we start at sector 6
    - Rotational delay to read sector 0 is $\frac{R}{2}$.
    - Rotational delay to read sector 5 is $R - 1$ (worst case).



A Single Track Plus A Head

Multiple Tracks: Seek Time

Three Tracks Plus A Head
(Right: With Seek)
(e.g., read to sector 11)



- **Seek**
  - Move the disk arm to the correct track
  - The time to move head to the track containing the desired sector is called **seek time**.
  - It is one of the most costly disk operations.

Phases of Seek

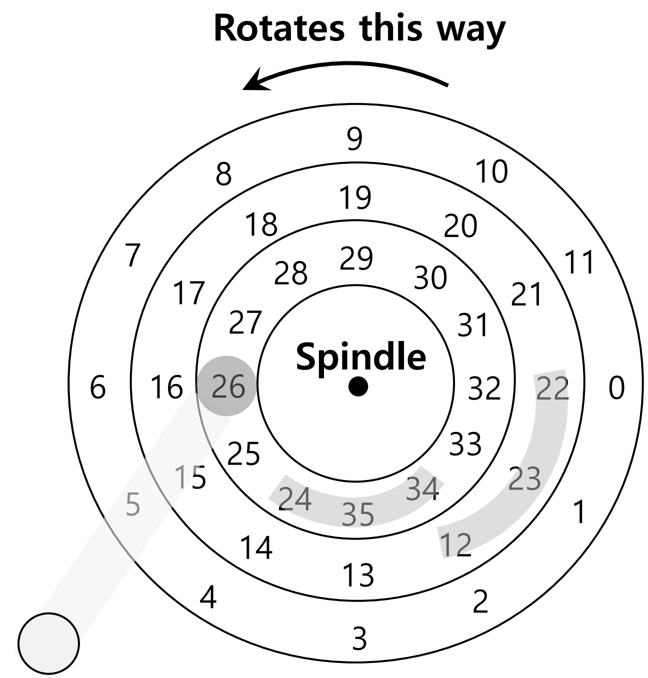
- Acceleration → Coasting → Deceleration → Settling
  - **Acceleration**: The disk arm gets moving.
  - **Coasting**: The arm is moving at full speed.
  - **Deceleration**: The arm slows down.
  - **Settling**: The head is *carefully positioned* over the correct track.
    - The settling time is often quite significant, e.g., 0.5 to 2ms.

Phases of I/O

- Seek → Rotational delay → Transfer
- **Transfer**: The final phase of I/O
  - Data is either actually *read from* or *written to* the surface.

Track Skew

- Used in some hard disks to make sure that sequential reads can be properly serviced *even when crossing track boundaries*.
- *Without track skew*, the head would be moved to the next track but the desired next block might have already rotated under the head.



Three Tracks: Track Skew Of 2

Cache or Track Buffer

- The cache holds data read from or written to the disk.
  - Allows the drive to *quickly respond* to requests.
  - Small amount of memory (usually around 8 or 16 MB)

Write on cache

- **Writeback** (immediate reporting)
  - Acknowledge that a write has completed when it has put the data in its memory.
  - Faster but dangerous.
- **Write through**
  - Acknowledge that a write has completed only after the data has actually been written to disk.

I/O Time: Doing The Math

- I/O time
  - $T_{I/O} = T_{seek} + T_{rotation} + T_{transfer}$
- Rate of I/O
  - $R_{I/O} = \frac{Size_{Transfer}}{T_{I/O}}$

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

I/O Time: Doing The Math

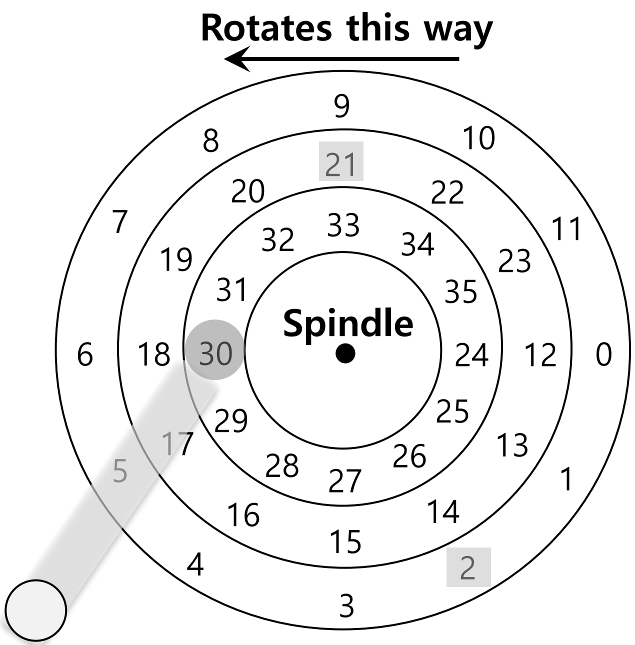
- **Random workload**: Issue 4KB read to random locations on the disk
- **Sequential workload**: Read 100MB consecutively from the disk

Disk Drive Performance: SCSI vs SATA, Random vs Sequential

		Cheetah 15K.5	**Barracuda**
T_{seek}		4 ms	9 ms
$T_{rotation}$		2 ms	4.2 ms
Random	$T_{transfer}$	30 μs	38 μs
	$T_{I/O}$	6 ms	13.2 ms
	$R_{I/O}$	0.66 MB/s	0.31 MB/s
Sequential	$T_{transfer}$	800 ms	950 ms
	$T_{I/O}$	806 ms	963.2 ms
	$R_{I/O}$	125 MB/s	105 MB/s

Disk Scheduling

- **Disk Scheduler** decides which I/O request to schedule next.
- **SSTF** (Shortest Seek Time First)
  - Order the queue of I/O request by track
  - Pick requests on the nearest track to complete first



SSTF: Scheduling Request 21 and 2

Issue the request to 21 → issue the request to 2

SSTF is not a panacea.

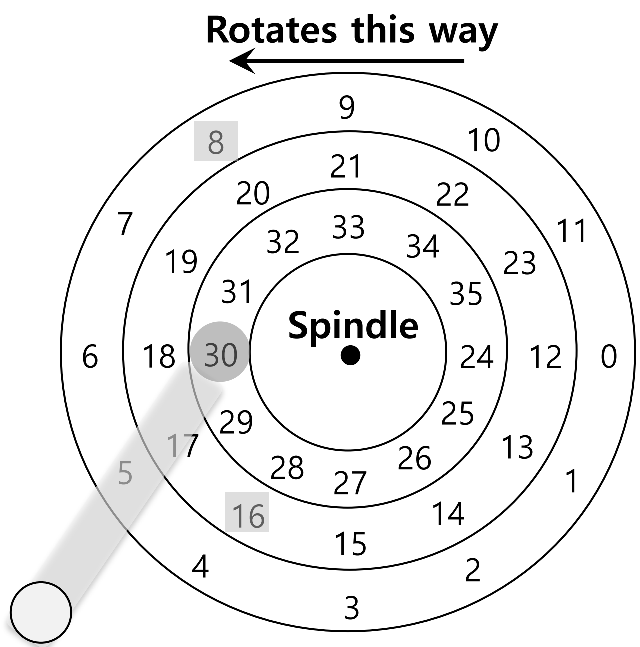
- **Problem 1**: The drive geometry is not available to the host OS
  - Solution: OS can simply implement Nearest-block-first (NBF)

- **Problem 2**: Starvation
  - If there were a steady stream of request to the inner track, request to other tracks would then be ignored completely.

Elevator (a.k.a. SCAN or C-SCAN)

- Move across the disk servicing requests in order across the tracks.
  - **Sweep**: A single pass across the disk
    - If a request comes for a block on a track that has already been services on this sweep of the disk, it is queued until the next sweep.
  - **F-SCAN**
    - Freeze the queue to be serviced when it is doing a sweep
    - Avoid starvation of far-away requests
  - **C-SCAN** (Circular SCAN)
    - Sweep from outer-to-inner, and then inner-to-outer, etc.

How to account for Disk rotation costs?



SSTF: Sometimes Not Good Enough

- If rotation is faster than seek : request 16 → request 8
- If seek is faster than rotation : request 8 → request 16

On modern drives, both seek and rotation are roughly equivalent:
Thus, SPTF (Shortest Positioning Time First) is useful.

I/O merging

- **Reduce the number of request** sent to the disk and lowers overhead
  - E.g., read blocks 33, then 8, then 34:
    - The scheduler merge the request for blocks 33 and 34 *into a single two-block request*.