Principles of Operating Systems

CS 446/646

5. Input/Output

a. Overview of the O/S Role in I/O

b. Principles of I/O Hardware

c. I/O Software Layers

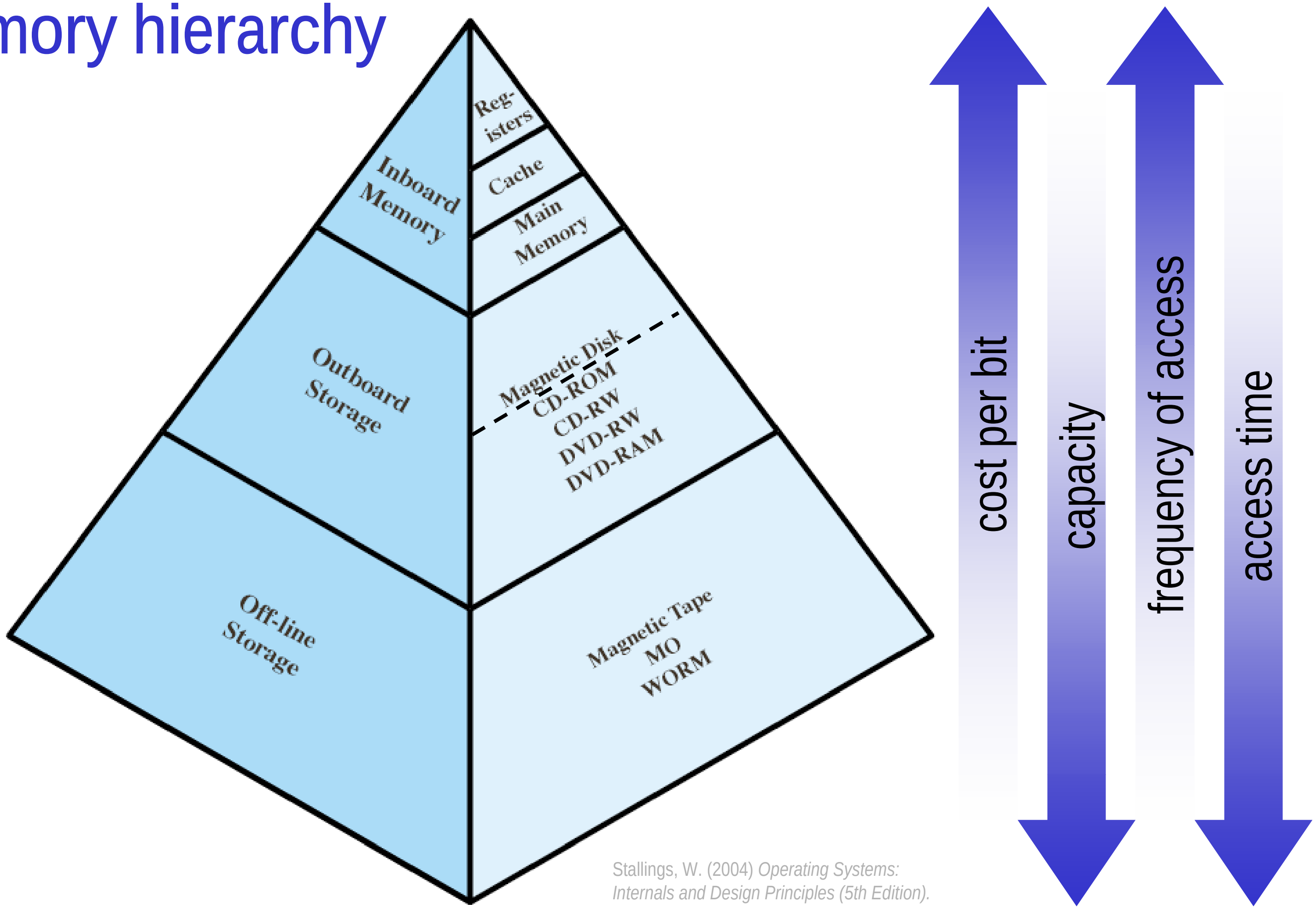
d. **Disk Management**

- ✓ Physical disk characteristics
- ✓ Disk formatting
- ✓ Disk scheduling

5.d Disk Management

Physical disk characteristics

- The memory hierarchy



Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

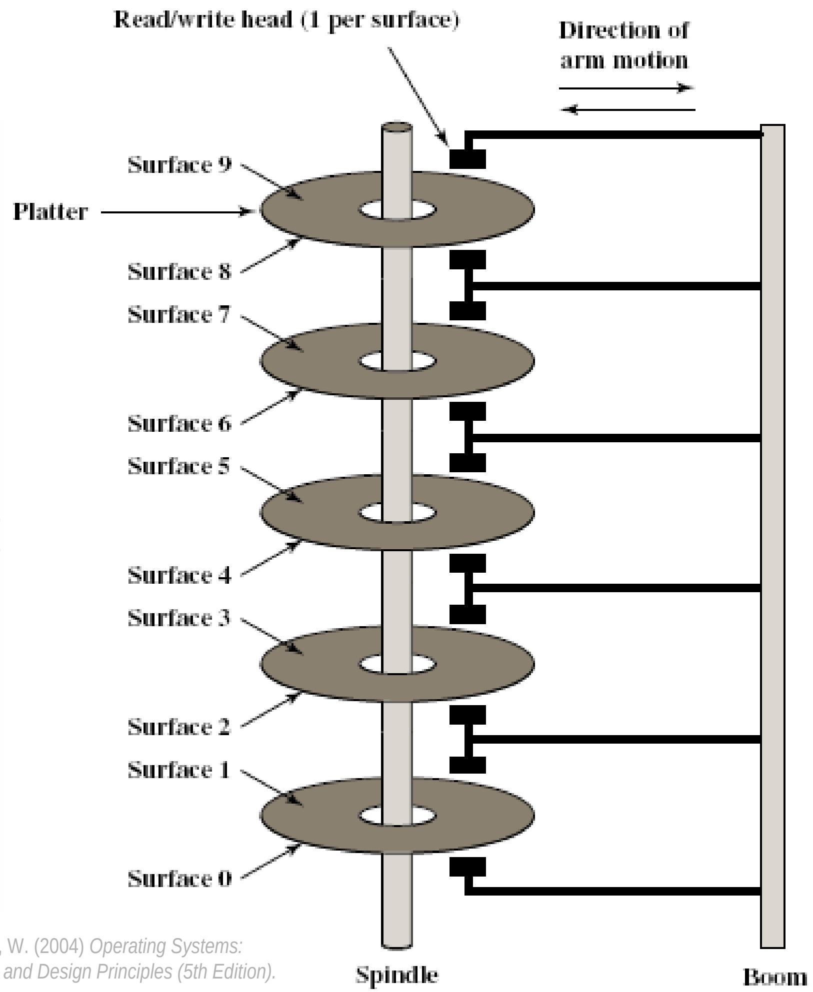
The memory hierarchy

5.d Disk Management

Physical disk characteristics

- Rigid ("hard") magnetic disks
  - ✓ remain today the most important secondary memory (although the gap between CPU and disk performance has increased)
  - ✓ diameter shrunk from 50 cm down to 12 or 3 cm (notebooks)
  - ✓ "Winchester" disks are sealed
  - ✓ components of a disk drive:
    - one or several aluminum platters stacked vertically
    - platters have magnetizable coating on both sides
    - one pair of read/write movable heads per platter surface (heads hover on air cushion, don't make contact)
    - all heads mechanically fixed so they move together and are all at same distance from center

5.d Disk Management
Physical disk characteristics

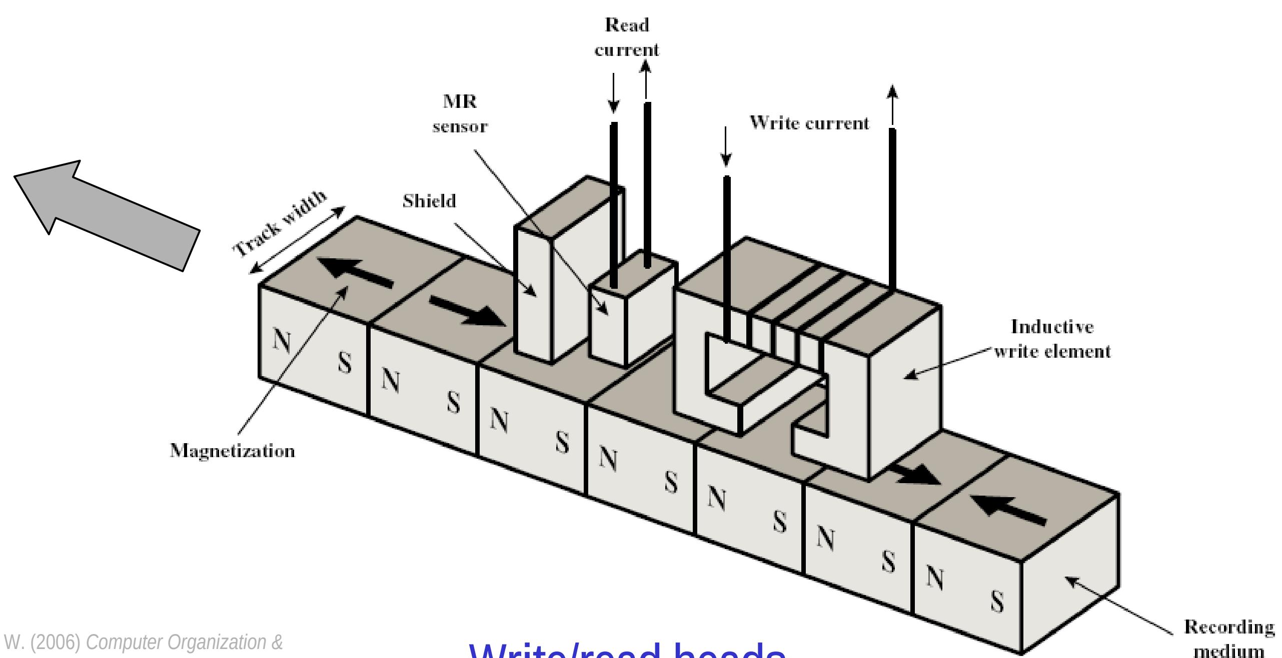


Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

Components of a disk drive

5.d Disk Management
Physical disk characteristics

- In modern systems, read and write heads are separate
  - ✓ the write head is an induction coil: produces a magnetic field
  - ✓ the read head is a magnetoresistive (MR) sensor: resistance depends on magnetic field, thus generates variable voltage



Stallings, W. (2006) *Computer Organization & Architecture: Designing for Performance (7th Edition).*

Write/read heads

5.d Disk Management

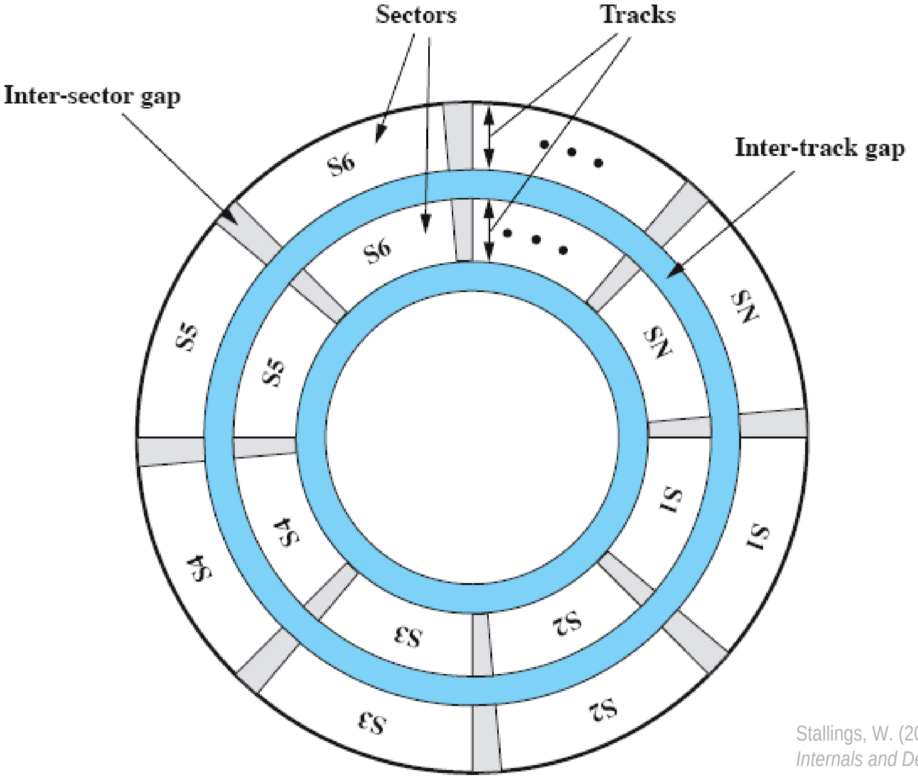
Disk formatting

- Data organization and formatting
  - ✓ after manufacturing, there is no information on the disk: just a blank slate (continuous surface of magnetizable metal oxide)
  - ✓ before a disk can be used, each platter must receive a **low-level format** ("physical format") done by code in I/O controller:
    - series of concentric **tracks** (not grooves)
    - each tracks contains **sectors**, separated by short gaps
  - ✓ then the disk may be **partitioned**
  - ✓ finally, each partition receives a **high-level format** ("logical"):
    - boot sector, free storage map, file allocation table, etc.

    → *we'll see more of this in the File System chapter*

5.d Disk Management
Disk formatting

- A disk is addressed as a 1-D array of logical blocks
  - ✓ translation between logical block # and track # + sector #

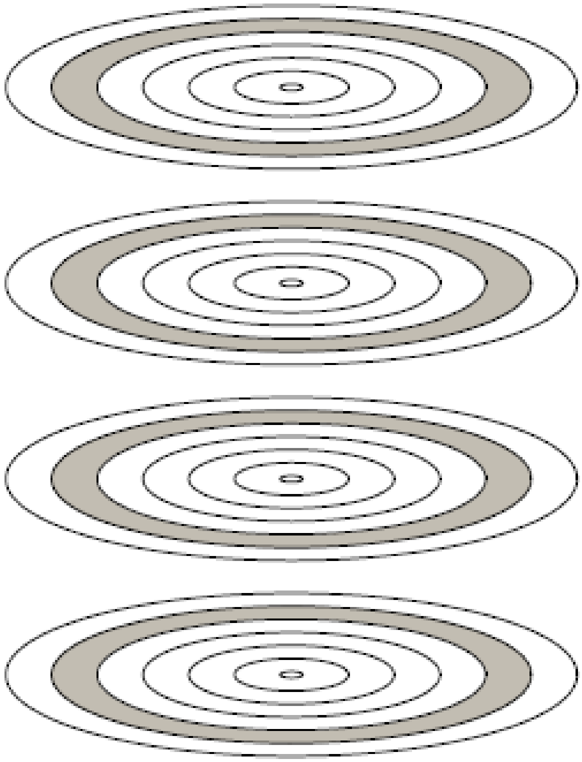


Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

Disk data layout

5.d Disk Management
Disk formatting

- Vertically aligned tracks on multiple platters are called "cylinders"



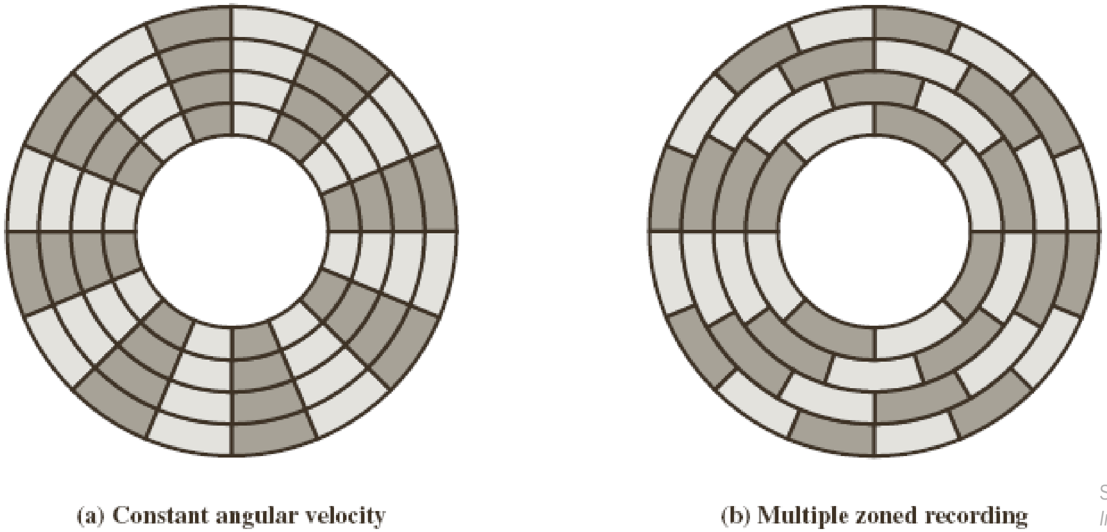
Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

Tracks and cylinders

5.d Disk Management

Disk formatting

- **Disk layout methods**
  - ✓ **constant angular velocity**: pie-shaped sectors, same number per track → simple but wasted space on the long outer tracks
  - ✓ **multiple zone recording**: fixed-length sectors, variable number per track → greater data density but more complicated access



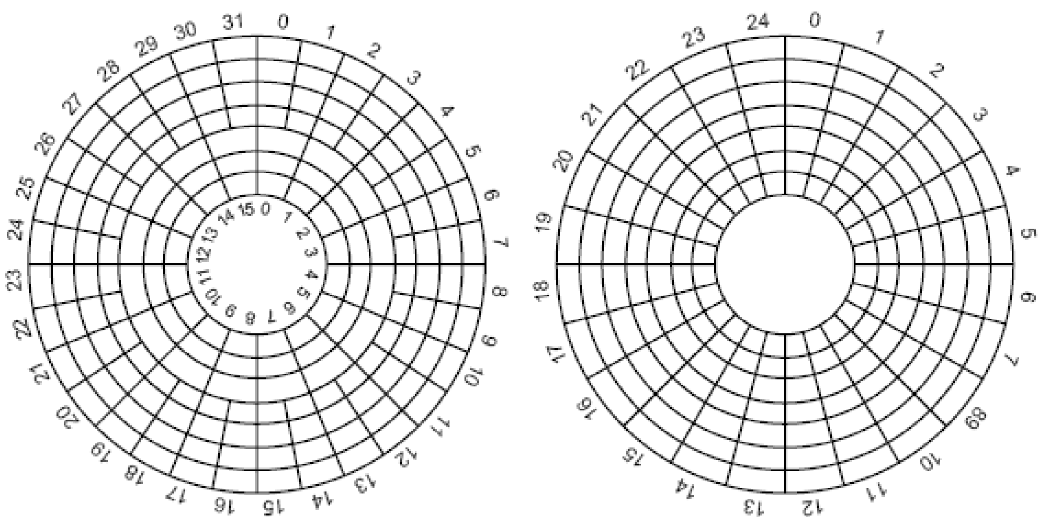
Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

Comparison of disk layout methods

5.d Disk Management

Disk formatting

- **Virtual disk geometry**
  - ✓ most disks are physically MZR but may still present a simpler, virtual CAV geometry to the O/S
  - ✓ the O/S driver uses cylinder, track, sector coordinates (*x*, *y*, *z*) which are remapped into zones by the I/O controller



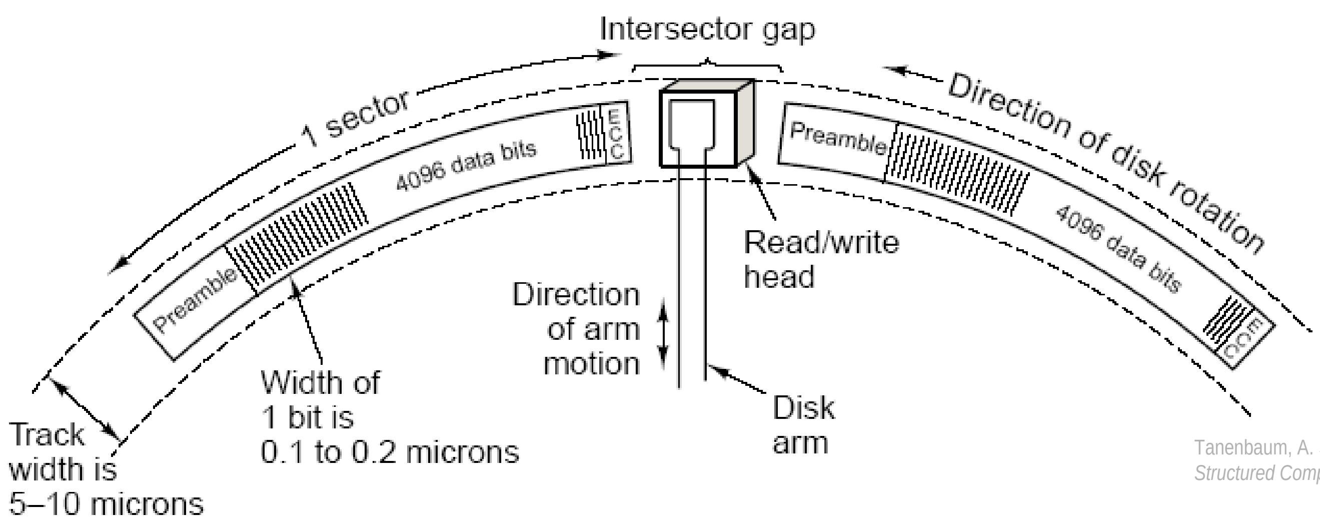
Tanenbaum, A. S. (2001)
Modern Operating Systems (2nd Edition).

Physical geometry vs. virtual geometry

5.d Disk Management

Disk formatting

- Tracks are divided into fixed-length sectors
  - each sector typically contains
    - 512 bytes of data
    - preceded by a preamble (for head synchronization)
    - followed by an error-correcting code (ECC)



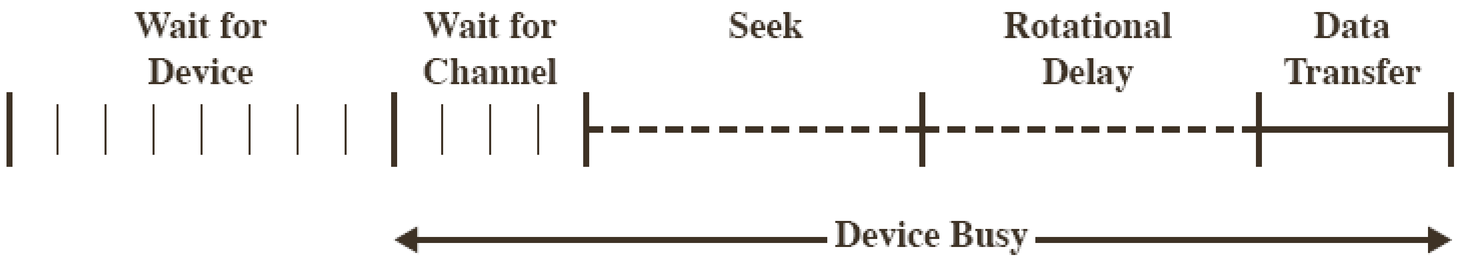
Tanenbaum, A. S. (2006) *Structured Computer Organization (5th Edition).*

A portion of a disk track showing two sectors

5.d Disk Management

Disk scheduling

- Disk performance parameters
  - **seek time**: time it takes to position the head at the track
  - **rotational delay**: time it takes for the beginning of the sector to reach the head
  - **access time** = seek time + rotational delay
  - **transfer time**: time required for sector data transfer



Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

Timing of a disk I/O transfer

5.d Disk Management
Disk scheduling

- **Disk performance parameters**
  - ✓ average seek time typically < 10 ms (thanks to small diameter)
  - ✓ rotational speed $r \approx 7{,}500$ rpm = 1r / 8 ms $\rightarrow$ 4 ms rot. delay
  - ✓ transfer time $T = b / rN$ with b / N = transferred bytes / track

Parameter	IBM 360-KB floppy disk	WD 18300 hard disk
Number of cylinders	40	10601
Tracks per cylinder	2	12
Sectors per track	9	281 (avg)
Sectors per disk	720	35742000
Bytes per sector	512	512
Disk capacity	360 KB	18.3 GB
Seek time (adjacent cylinders)	6 msec	0.8 msec
Seek time (average case)	77 msec	6.9 msec
Rotation time	200 msec	8.33 msec
Motor stop/start time	250 msec	20 sec
Time to transfer 1 sector	22 msec	17 μsec

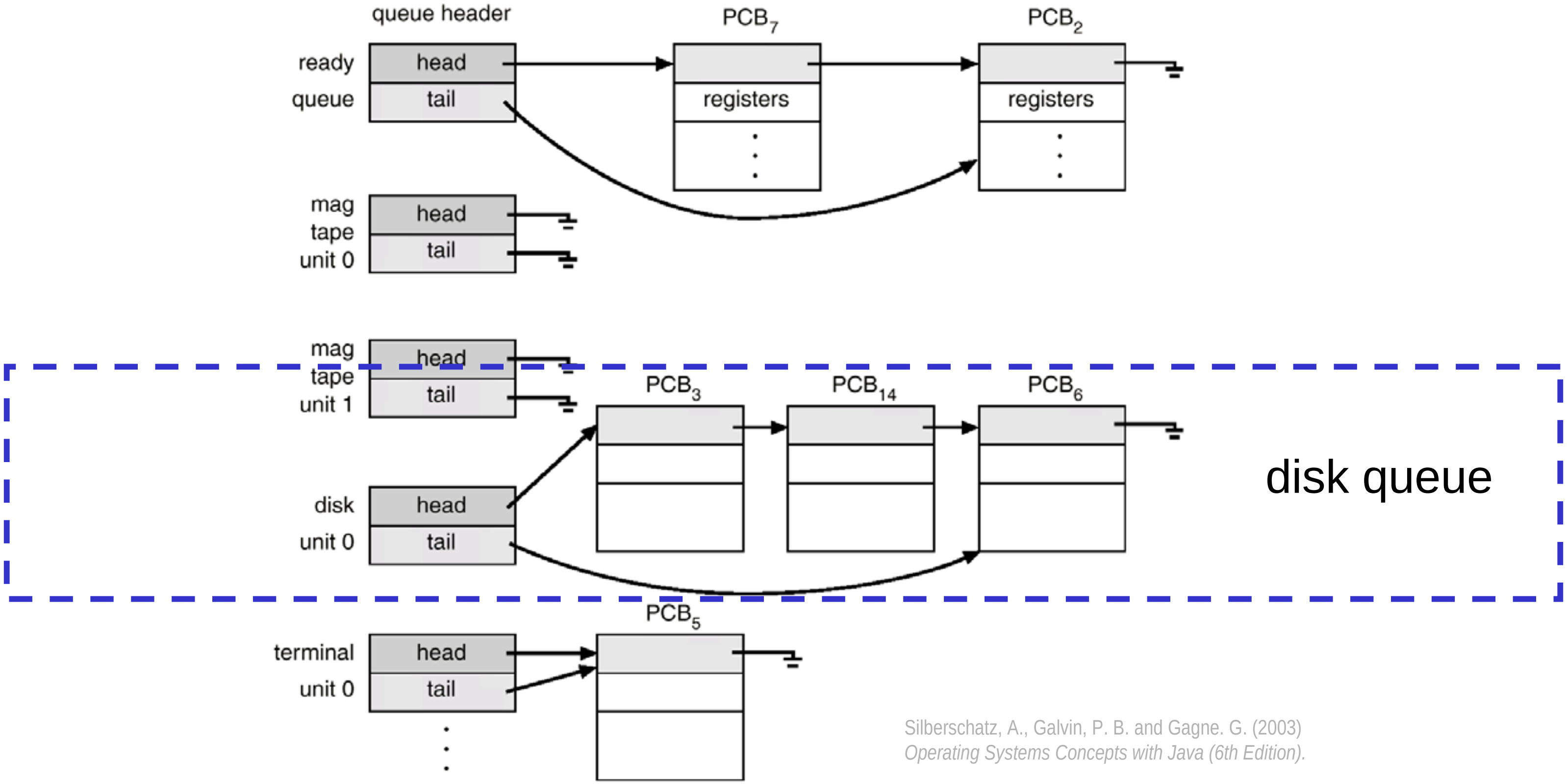
Two opposites on the historical scale of disk parameters

Tanenbaum, A. S. (2001)
Modern Operating Systems (2nd Edition).

5.d Disk Management

Disk scheduling

- Additional waiting time for device availability
  - ✓ processes blocked for I/O are put into device-specific queues



Silberschatz, A., Galvin, P. B. and Gagne. G. (2003)
Operating Systems Concepts with Java (6th Edition).

Various I/O device queues

5.d Disk Management
Disk scheduling

- Why disk scheduling matters: a timing comparison
  - ✓ total average service time

    $$T_{service} = T_{seek} + T_{rotational} + T_{transfer}$$

    $$= T_{seek} + 1/2r + b/rN$$

  - ✓ assume T_{seek} = 4 ms, r = 7,500 rpm, 500 sectors per track × 512 bytes per sector → $T_{transfer}$ = 0.016 ms / sector
  - ✓ first case: reading 2,500 randomly scattered sectors

    $$T_{service} = 2{,}500 \times (4\text{ ms} + 4\text{ ms} + 0.016\text{ ms}) = 20\text{ seconds}$$

  - ✓ first case: reading 2,500 contiguous sectors (in 5 tracks)

    $$T_{service} = 4\text{ ms} + 5 \times 4\text{ ms} + 2{,}500 \times 0.016\text{ ms} = 64\text{ ms}$$

  → *the order of sector access requests can make a big difference!*

5.d Disk Management
Disk scheduling

- Overview of disk scheduling policies
  - **kernel-level scheduling**: based on requestor process
    - control of scheduling outside of disk management software
    - not intended to optimize disk utilization
    - main objective is process priorities defined by the O/S
    - or following a blind, generic policy such as FIFO (no starvation) or LIFO (locality)
  - **driver-level scheduling**: based on requested item
    - goal is to optimize disk utilization
    - the disk-specific software has expertise on how requests should be ordered

5.d Disk Management

Disk scheduling

- Overview of disk scheduling policies
  - ✓ kernel-level (process) vs. driver-level (request) scheduling

Name	Description	Remarks
	Selection according to requestor	
RSS	Random scheduling	For analysis and simulation
FIFO	First in first out	Fairest of them all
PRI	Priority by process	Control outside of disk queue management
LIFO	Last in first out	Maximize locality and resource utilization
	Selection according to requested item	
SSTF	Shortest service time first	High utilization, small queues
SCAN	Back and forth over disk	Better service distribution
C-SCAN	One way with fast return	Lower service variability
N-step-SCAN	SCAN of *N* records at a time	Service guarantee
FSCAN	N-step-SCAN with *N* = queue size at beginning of SCAN cycle	Load sensitive

Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*

5.d Disk Management

Disk scheduling

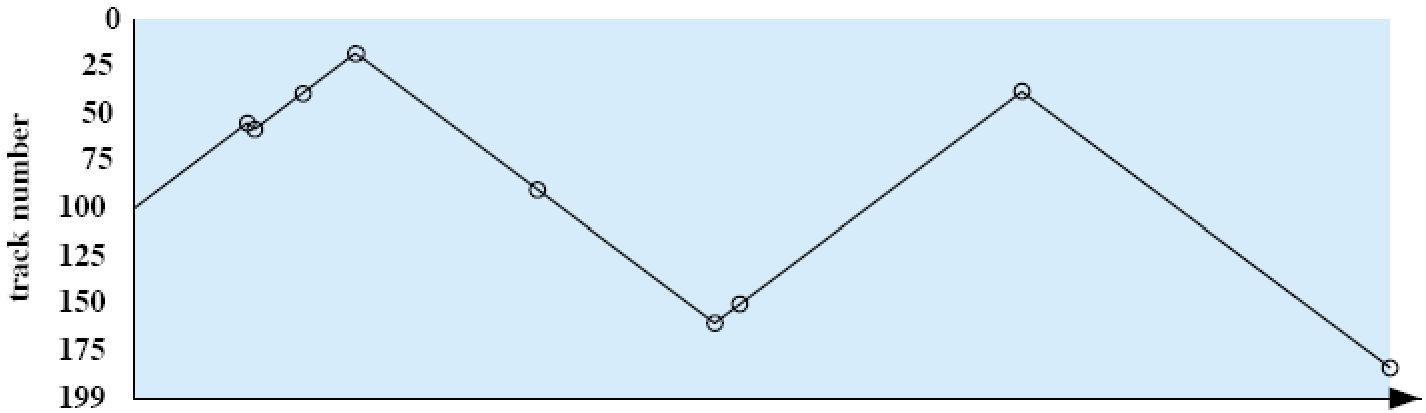
- Comparing performance of scheduling policies
  - ✓ assume disk with 200 tracks
  - ✓ assume sequence of requested tracks in order received by disk scheduler: 55, 58, 39, 18, 90, 160, 150, 38, 184
  - ✓ assume disk head initially located at track #100
  - ✓ we will compare FIFO, SSTF, SCAN, C-SCAN

5.d Disk Management

Disk scheduling

- First-In-First-Out (FIFO)
  - ✓ requests are processed in arrival order
  - ✓ fair and no risk of starvation
  - ✓ ok if few processes and requests cluster file sectors (locality)
  - ✓ generally bad, though, as interleaving causes random seek jumps and waste of time

Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*



(a) FIFO

(a) FIFO (starting at track 100)

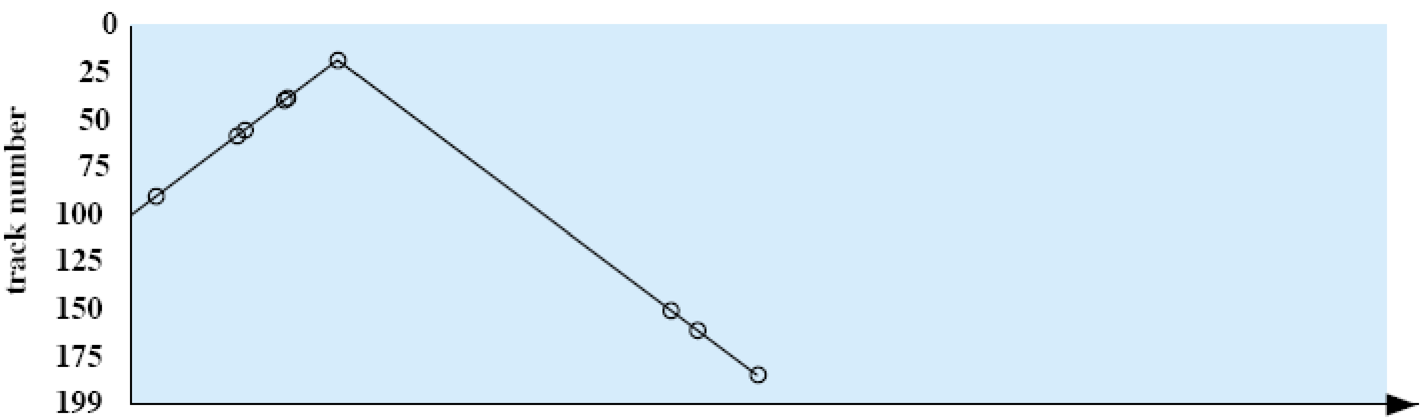
Next track accessed	Number of tracks traversed
55	45
58	3
39	19
18	21
90	72
160	70
150	10
38	112
184	146
Average seek length	55.3

5.d Disk Management

Disk scheduling

- **Shortest Service (Seek) Time First (SSTF)**
  - ✓ select the request that requires the least arm movement, i.e., the shortest seek time
  - ✓ much better than random or FIFO, however greater risk of starvation: requests in remote disk area may remain unfulfilled as long as there are shorter ones

Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*



(b) SSTF

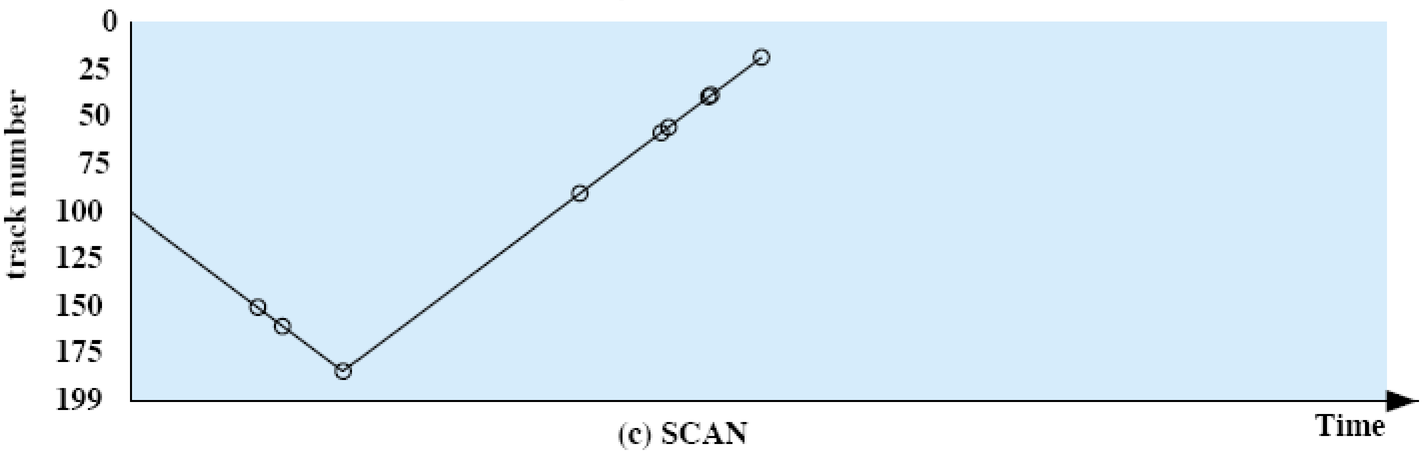
(b) SSTF
(starting at track 100)

Next track accessed	Number of tracks traversed
90	10
58	32
55	3
39	16
38	1
18	20
150	132
160	10
184	24
Average seek length	27.5

5.d Disk Management
Disk scheduling

- **Scan or "elevator" algorithm (SCAN)**
  - ✓ to prevent starvation, the arm moves in one direction only and satisfies requests "en route"
  - ✓ arm direction is reversed when reaching the last track (innermost or outermost)
  - ✓ . . . or as soon as reaching last request (LOOK: the variant implemented in Linux)

Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*



(c) SCAN

(c) SCAN (starting at track 100, in the direction of increasing track number)

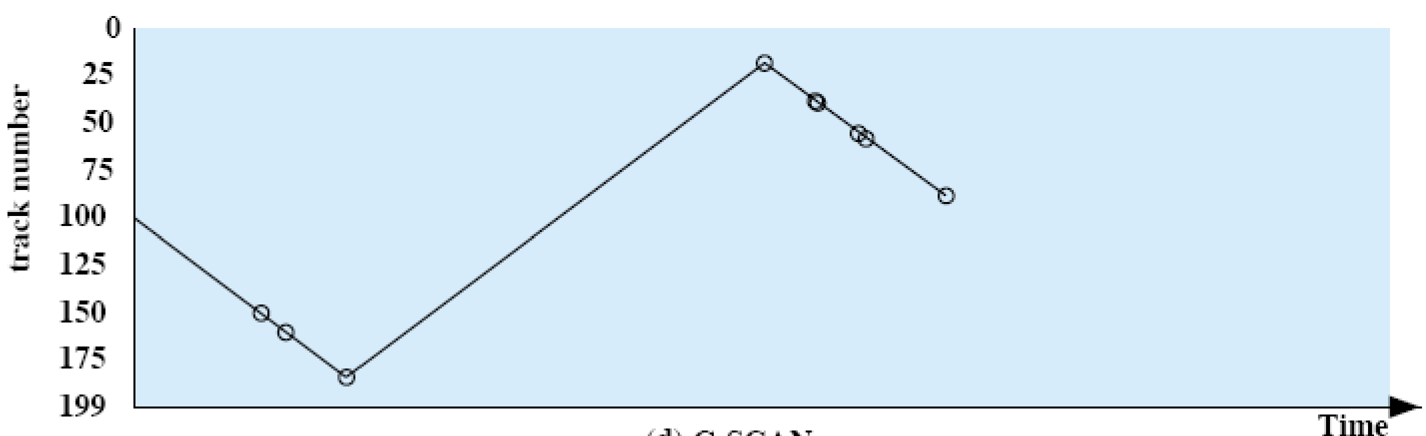
Next track accessed	Number of tracks traversed
150	50
160	10
184	24
90	94
58	32
55	3
39	16
38	1
18	20
Average seek length	27.8

5.d Disk Management

Disk scheduling

- Circular scan (C-SCAN)
  - ✓ same as SCAN except the arm direction of movement is never reversed
  - ✓ this reduces the maximum delay experienced by new requests that arrived at the opposite end of the disk

Stallings, W. (2004) *Operating Systems: Internals and Design Principles (5th Edition).*



(d) C-SCAN

(d) C-SCAN (starting at track 100, in the direction of increasing track number)	
Next track accessed	**Number of tracks traversed**
150	50
160	10
184	24
18	166
38	20
39	1
55	16
58	3
90	32
Average seek length	35.8

Principles of Operating Systems
CS 446/646

5. Input/Output

a. Overview of the O/S Role in I/O

b. Principles of I/O Hardware

c. I/O Software Layers

d. **Disk Management**

- ✓ Physical disk characteristics
- ✓ Disk formatting
- ✓ Disk scheduling

Principles of Operating Systems

CS 446/646

5. Input/Output

a. Overview of the O/S Role in I/O

b. Principles of I/O Hardware

c. I/O Software Layers

d. Disk Management

Principles of Operating Systems
CS 446/646

0. Course Presentation
1. Introduction to Operating Systems
2. Processes
3. Memory Management
4. CPU Scheduling
5. Input/Output
6. **File System**
7. **Case Studies**