

Jin-Soo Kim
(jinsoo.kim@snu.ac.kr)

Systems Software &
Architecture Lab.

Seoul National University

Spring 2020

Hard Disk Drives (HDDs)



Three Pieces

- **Virtualization**

- Virtual CPUs
- Virtual memory

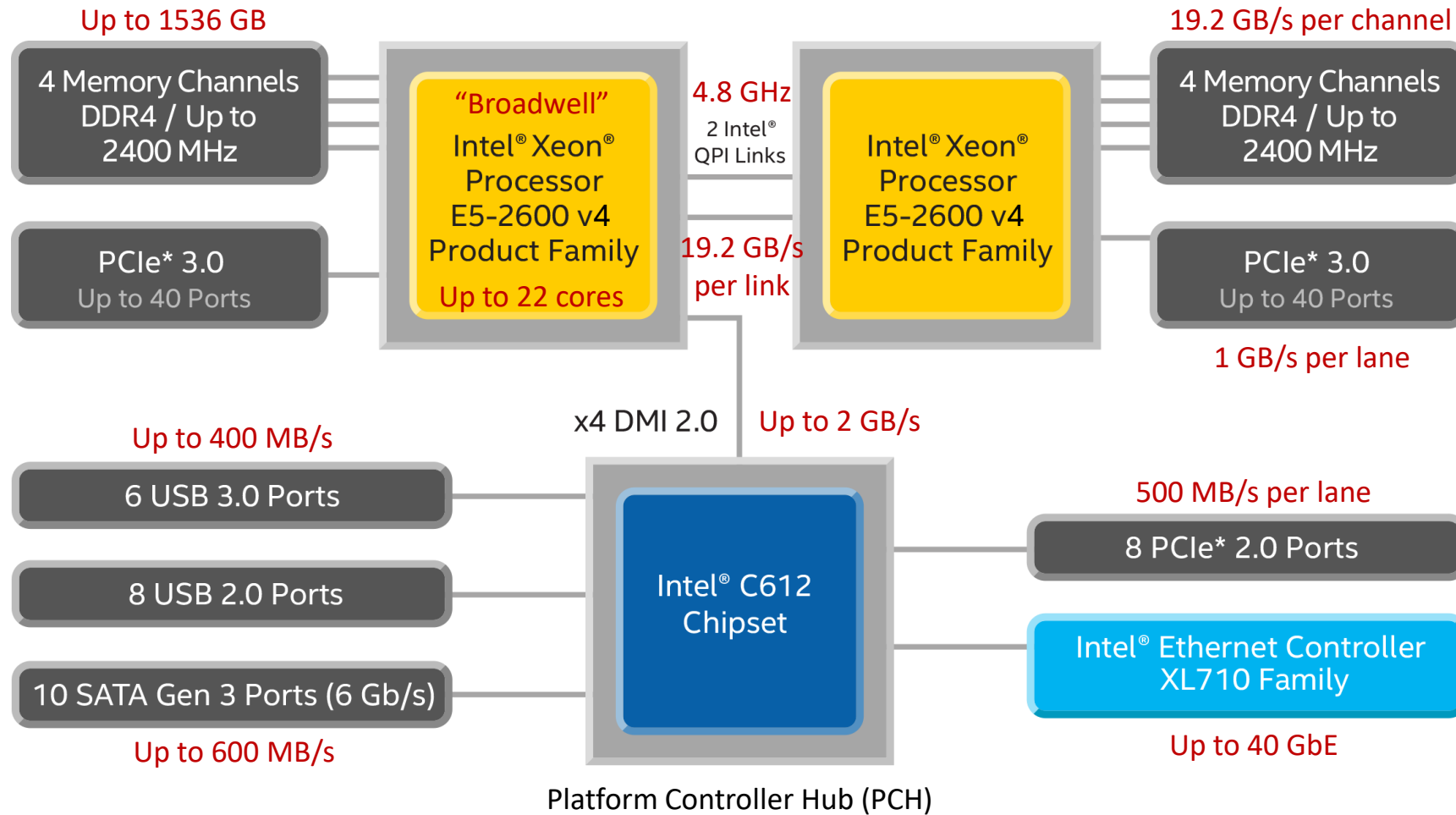
- **Concurrency**

- Threads
- Synchronization

- **Persistence**

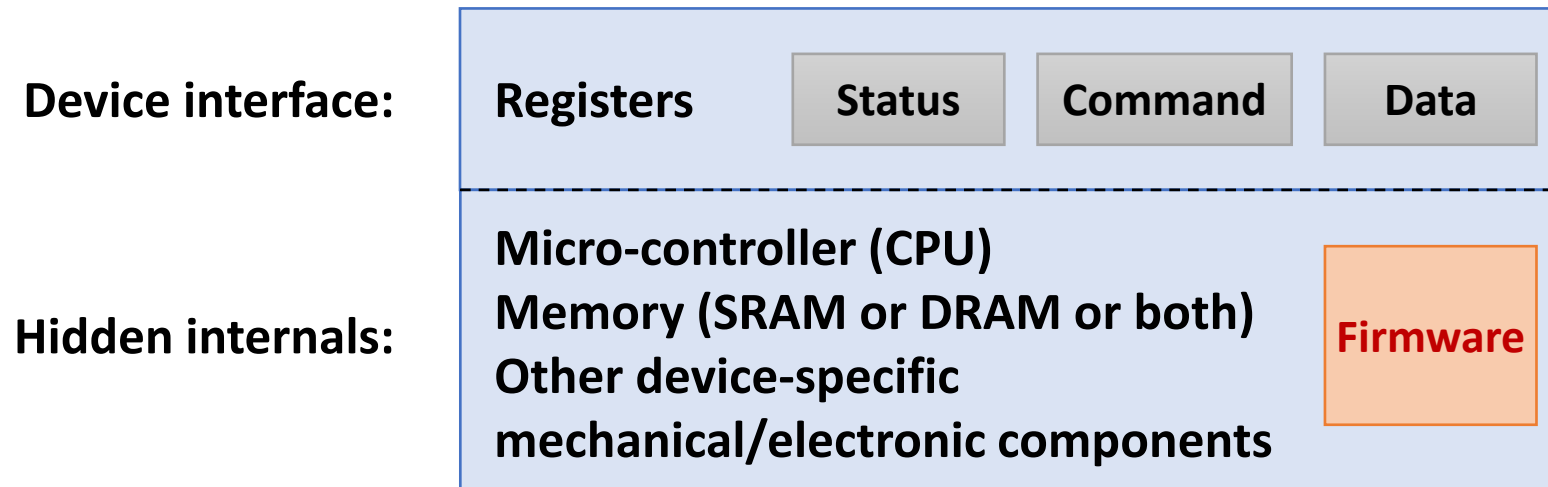
- How to make information persist, despite computer crashes, disk failures, or power outages?
- Storage
- File systems

Modern System Architecture

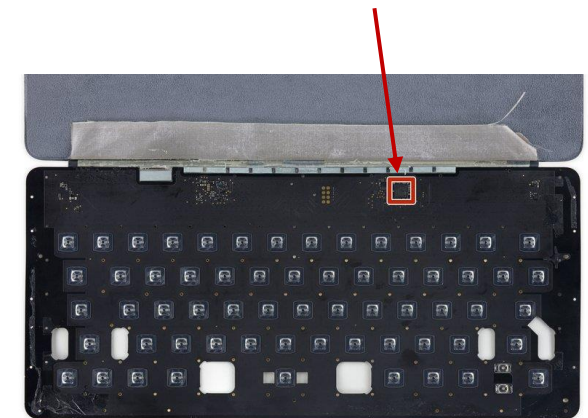


A Typical I/O Device

- **Control:** Special instructions (e.g., in & out in x86) vs. memory-mapped I/O (e.g., load & store)
- **Data transfer:** Programmed I/O (PIO) vs. DMA
- **Status check:** Polling vs. Interrupts



72MHz 32-bit
ARM Cortex-M3

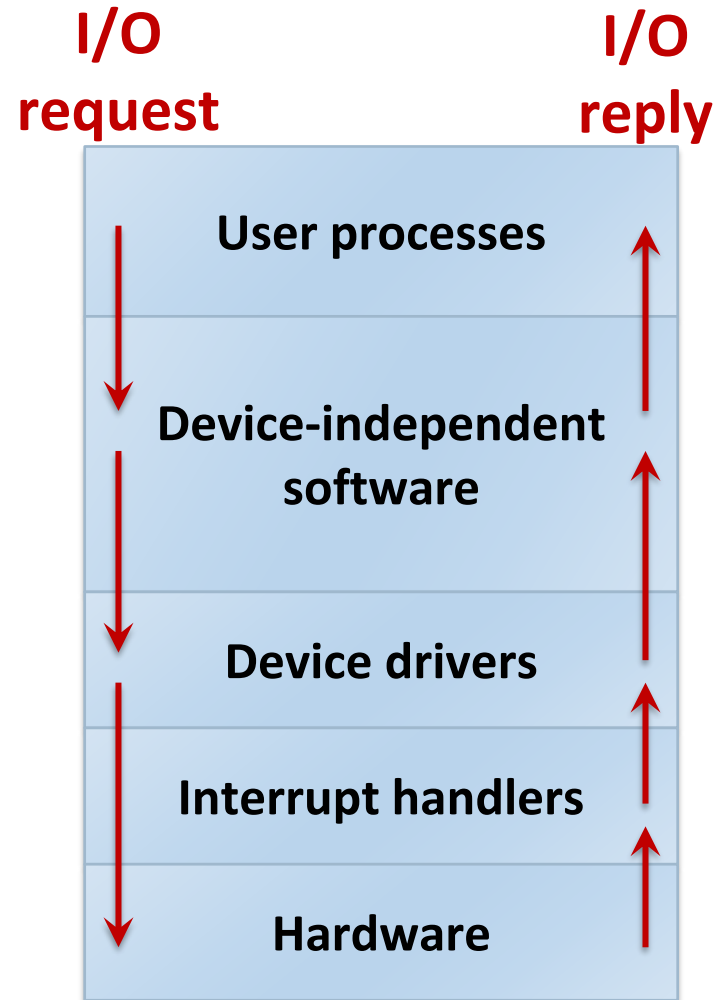


Source: <https://www.ifixit.com/Teardown/Smart+Keyboard+Teardown/53052>

Classifying I/O Devices

- device
 - Stores information in fixed-size blocks, each one with its own address
 - Typically, 512B or 4KB per block
 - Can read or write each block independently
 - Disks, tapes, etc.
- **Character device**
 - Delivers or accepts a stream of characters
 - Not addressable and no seek operation supported
 - Printers, networks, mouse, keyboard, etc.

I/O Stack



Make I/O call, format I/O, spooling

Naming, protection, blocking, buffering, allocation

Set up device registers, check status

Wake up driver when I/O completed

Perform I/O operation

Device Drivers

- **Device-specific code to control each I/O device**
 - Require to define a well-defined model and a standard interface
- **Implementation**
 - Statically linked with the kernel
 - Selectively loaded into the system during boot time
 - Dynamically loaded into the system during execution (especially for hot pluggable devices)
- **Variety is a challenge**
 - Many, many devices
 - Each has its own protocol

OS Reliability



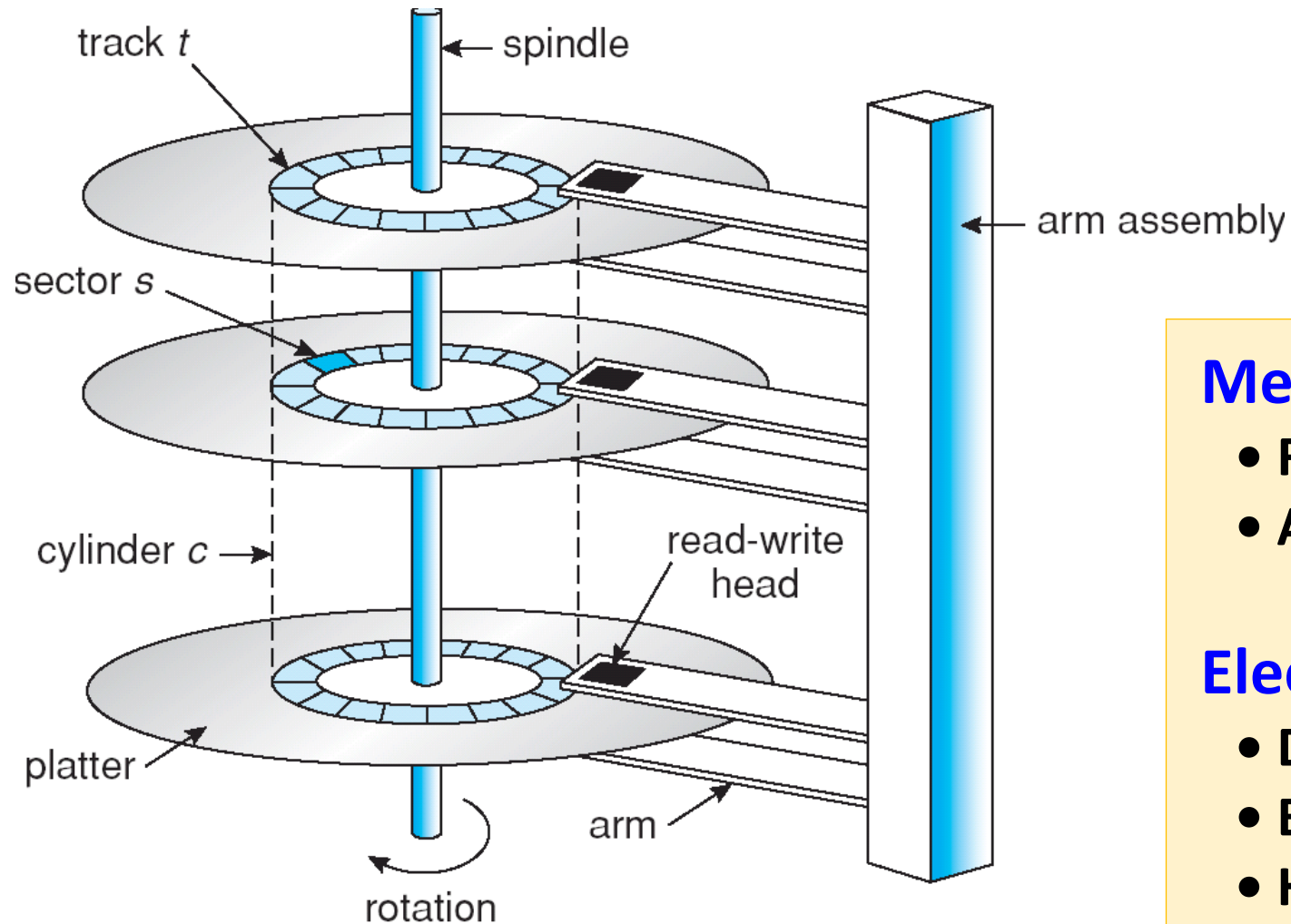
OS Reliability and Device Drivers

- Reliability remains a crucial, but unresolved problem
 - 5% of Windows systems crash every day
 - Huge cost of failures: stock exchange, e-commerce, etc.
 - Growing “unmanaged systems”: digital appliances, CE devices
- OS extensions are increasingly prevalent
 - 70% of Linux kernel code
 - Over 35,000 drivers with over 120,000 versions on WinXP
 - Written by less experienced programmer
- Extensions are a leading cause of OS failure
 - Drivers cause 85% of WinXP crashes
 - Drivers are 7 times buggier than the kernel in Linux

Secondary Storage

- Anything that is outside of “primary memory”
 - Does not permit direct execution of instructions or data retrieval via machine load/instructions
 - Abstracted as an array of sectors
 - Each sector is typically 512 bytes or 4096 bytes
- HDD (Hard Disk Drive) Characteristics
 - It's large: 100 GB or more
 - It's cheap: 8TB SATA3 hard disk costs 270,000won
 - It's persistent: data survives power loss
 - It's slow: milliseconds to access

HDD Architecture



Mechanical

- Rotating disks
- Arm assembly

Electronics

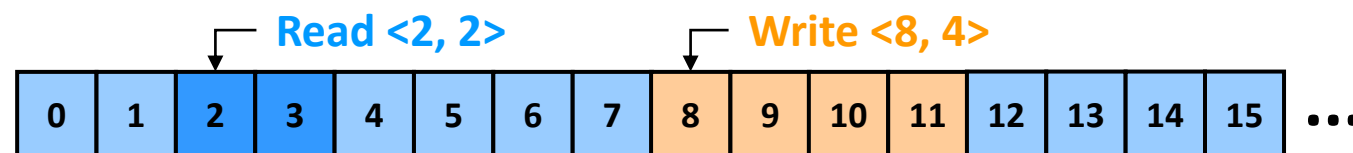
- Disk controller
- Buffer
- Host interface

A Modern HDD

- **Seagate Barracuda ST8000DM004 (8TB)**
 - 8 Heads, 4 Discs
 - 63 sectors/track, 16,383 cylinders
 - Avg. track density: 540K TPI (tracks/inch)
 - Avg. areal density: 1203 Gbits/sq.inch
 - Spindle speed: 5400 rpm (11.1ms / rotation)
 - Internal cache buffer: 256 MB
 - Average latency: 6.0 ms (previously, average seek time was 15.0 ms)
 - Max. I/O data transfer rate: 600 MB/s (SATA3)
 - Max. data transfer rate: 190 MB/s
 - Power-on to ready: < 15.0 sec

Interfacing with HDDs

- Cylinder-Head-Sector (CHS) scheme
 - Each block is addressed by <Cylinder #, Head #, Sector #>
 - The OS needs to know all disk “geometry” parameters
- Logical block addressing (LBA) scheme
 - First introduced in SCSI
 - Disk is abstracted as a logical array of blocks [0, ..., N-1]
 - Address a block with a “logical block address (LBA)”
 - Disk maps an LBA to its physical location
 - Physical parameters of a disk are hidden from OS



HDD Performance Factors

- **Seek time (T_{seek})**
 - Moving the disk arm to the correct cylinder
 - Depends on the cylinder distance (not purely linear cost)
 - Average seek time is roughly one-third of the full seek time
- **Rotational delay ($T_{rotation}$)**
 - Waiting for the sector to rotate under head
 - Depends on rotations per minute (RPM)
 - 5400, 7200 RPM common, 10K or 15K RPM for servers
- **Transfer time ($T_{transfer}$)**
 - Transferring data from surface into disk controller, sending it back to the host

HDD Performance Comparison

	Cheetah 15K.5	Barracuda
Capacity	300 GB	1 TB
RPM	15,000	7,200
Avg. Seek	4 ms	9 ms
Max Transfer	125 MB/s	105 MB/s
Platters	4	4
Cache	16MB	16/32 MB
Interface	SCSI	SATA
Random Read (4 KB)	$T_{seek} = 4\text{ms}$ $T_{rotation} = 60 / 15000 / 2 = 2\text{ms}$ $T_{transfer} = 4\text{KB} / 125\text{MB} = 32\mu\text{s}$ $R_{I/O} \approx 4\text{KB} / 6\text{ms} = 0.66 \text{ MB/s}$	$T_{seek} = 9\text{ms},$ $T_{rotation} = 60 / 7200 / 2 = 4.2\text{ms}$ $T_{transfer} = 4\text{KB} / 105\text{MB} = 37\mu\text{s}$ $R_{I/O} \approx 4\text{KB} / 13.2\text{ms} = 0.31 \text{ MB/s}$
Sequential Read (100 MB)	$T_{transfer} = 100\text{MB} / 125\text{MB} = 0.8\text{s}$ $R_{I/O} \approx 100\text{MB} / 0.8\text{s} = 125 \text{ MB/s}$	$T_{transfer} = 100\text{MB} / 105\text{MB} = 0.95\text{s}$ $R_{I/O} \approx 100\text{MB} / 0.95\text{s} = 105 \text{ MB/s}$

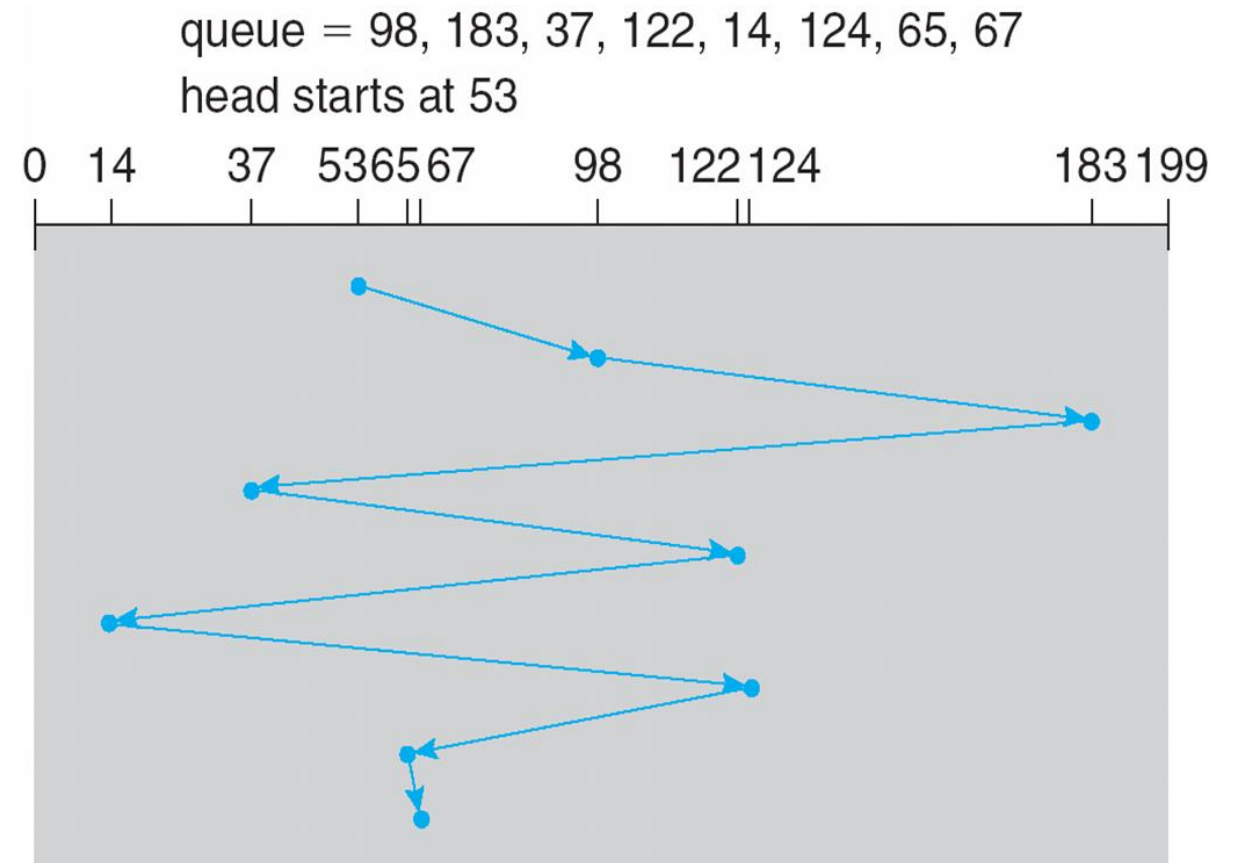
Disk Scheduling

- **Given a stream of I/O requests, in what order should they be served?**
 - Much different than CPU scheduling
 - Seeks are so expensive
 - Position of disk head relative to request position matters more than length of a job
- **Work conserving schedulers**
 - Always try to do work if there's work to be done
- **Non-work-conserving schedulers**
 - Sometimes, it's better to wait instead if system anticipates another request will arrive

FCFS

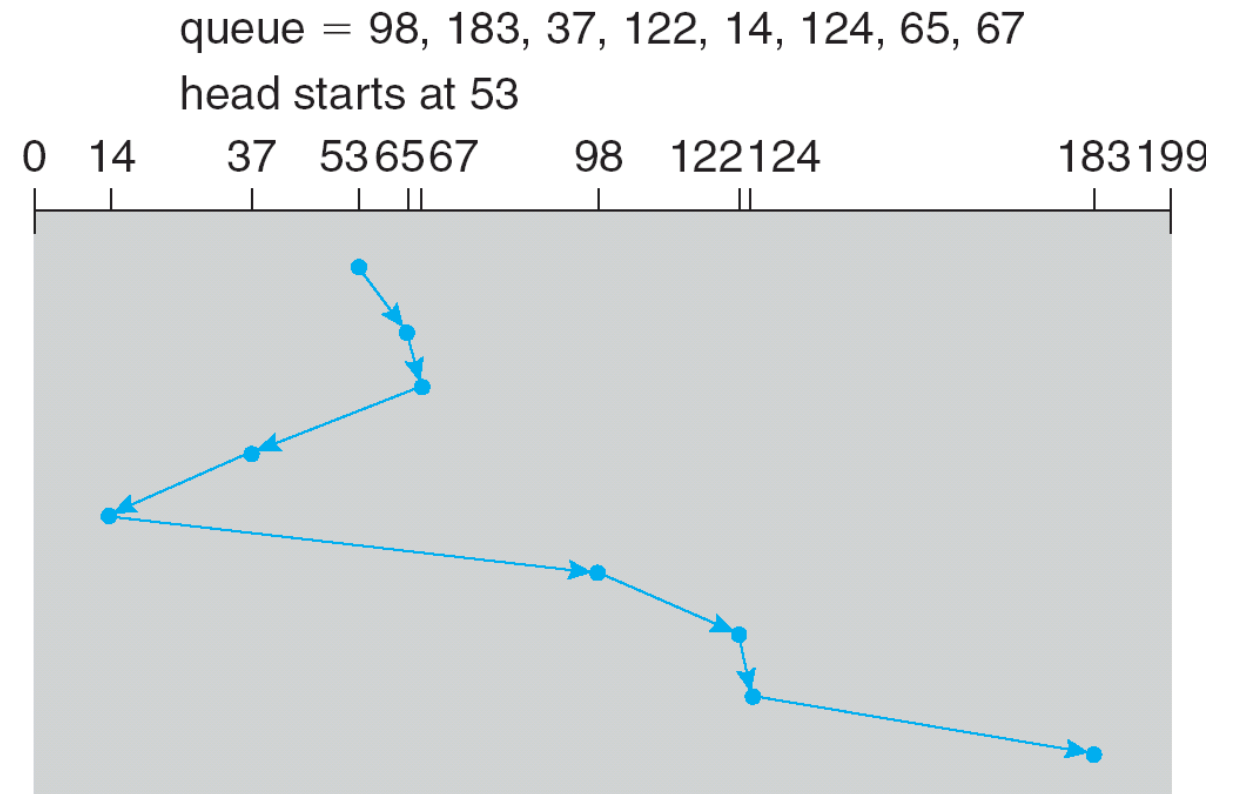
- First-Come First-Served (= do nothing)

- Reasonable when load is low
- Long waiting times for long request queues



SSTF

- Shortest Seek Time First
 - Minimizes arm movement (seek time)
 - Unfairly favors middle blocks
 - May cause starvation
- Nearest-Block-First (NBF) when the drive geometry is not available to the host OS



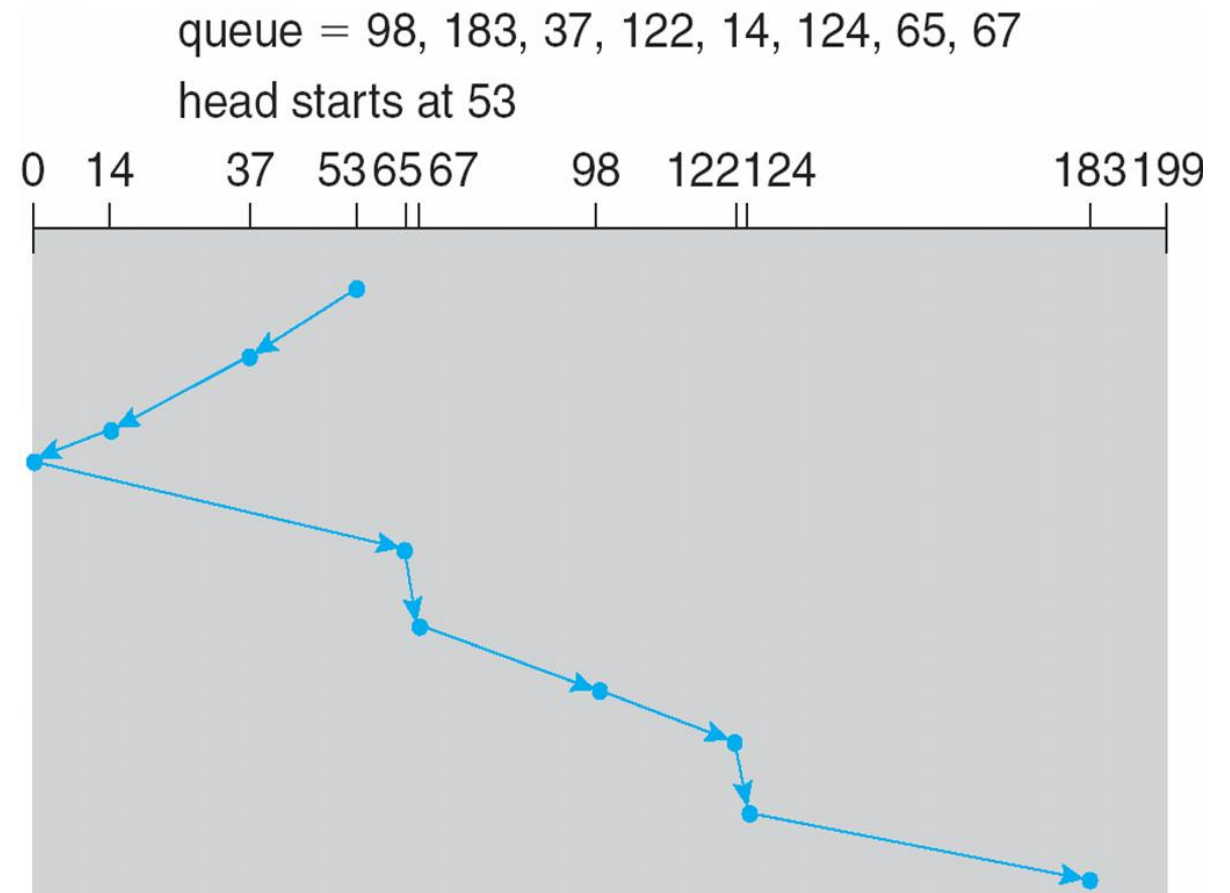
SCAN

■ SCAN

- Service requests in one direction until done, then reverse
- Skews wait times non-uniformly
- Favors middle blocks

■ F-SCAN

- Freezes the queue when it is doing a sweep
- Avoids starvation of far-away requests



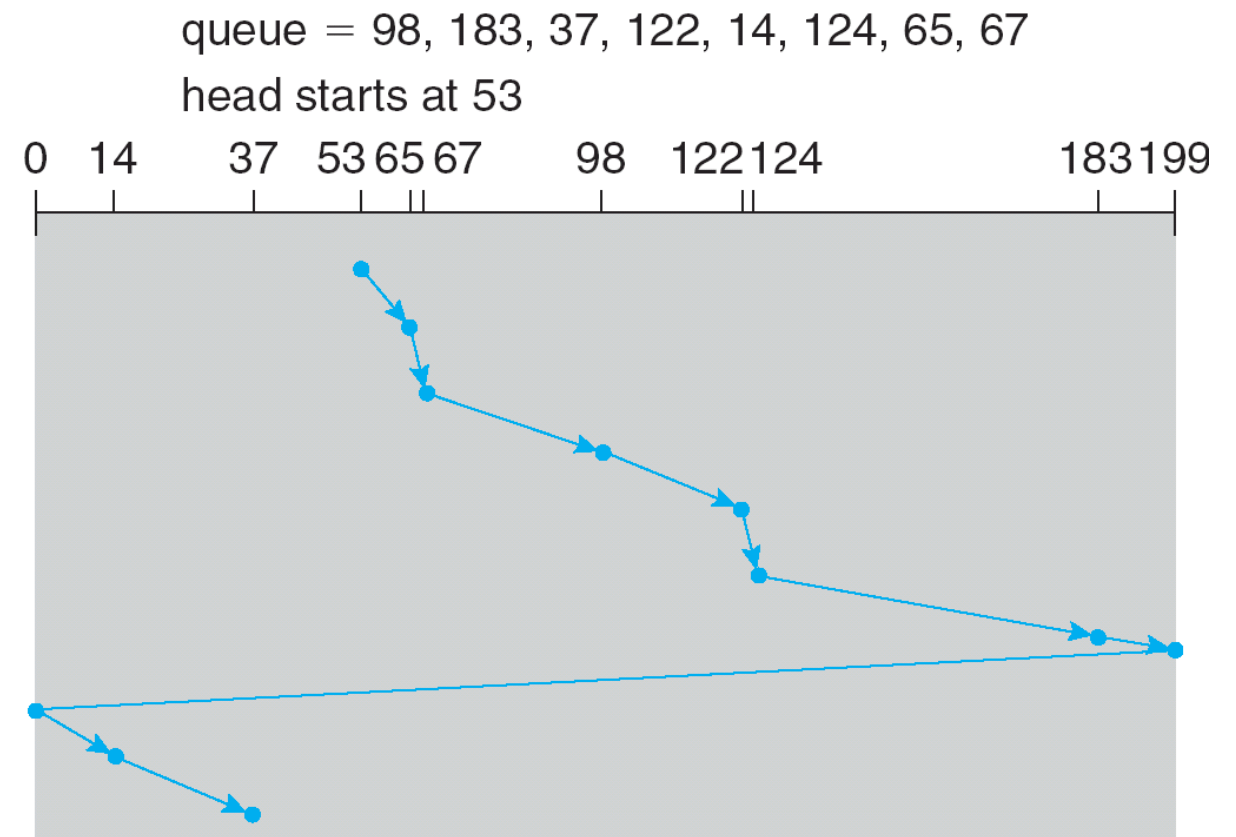
C-SCAN

- Circular SCAN

- Like SCAN, but only goes in one direction (e.g., typewriter)
- Uniform wait times

- SCAN and C-SCAN are referred to as the “ ” algorithm

- Both do not consider rotation



Modern Disk Scheduling

- I/O scheduler in the host OS
 - Improve overall disk throughput
 - Merge requests to reduce the number of requests
 - Sort requests to reduce disk seek time
 - Prevent starvation
 - Provide fairness among different processes
- Disk drive
 - Disk has multiple outstanding requests
 - e.g., SATA NCQ (Native Command Queueing): up to 32 requests
 - Disk schedules requests using its knowledge of head position and track layout
 - e.g., SPTF (Shortest Positioning Time First): consider rotation as well

Summary

- HDD is a block device
- Modern HDD interface is based on LBA (Logical Block Addressing)
 - SATA, SAS
- Modern disks support command queueing and scheduling
- “Unwritten contract” of HDDs
 - Sequential accesses are much better than random accesses
 - Distant LBAs lead to longer seek time
 - Data written is equal to data issued (no write amplification)
 - Media does not wear down
 - Storage devices are passive with little background activity