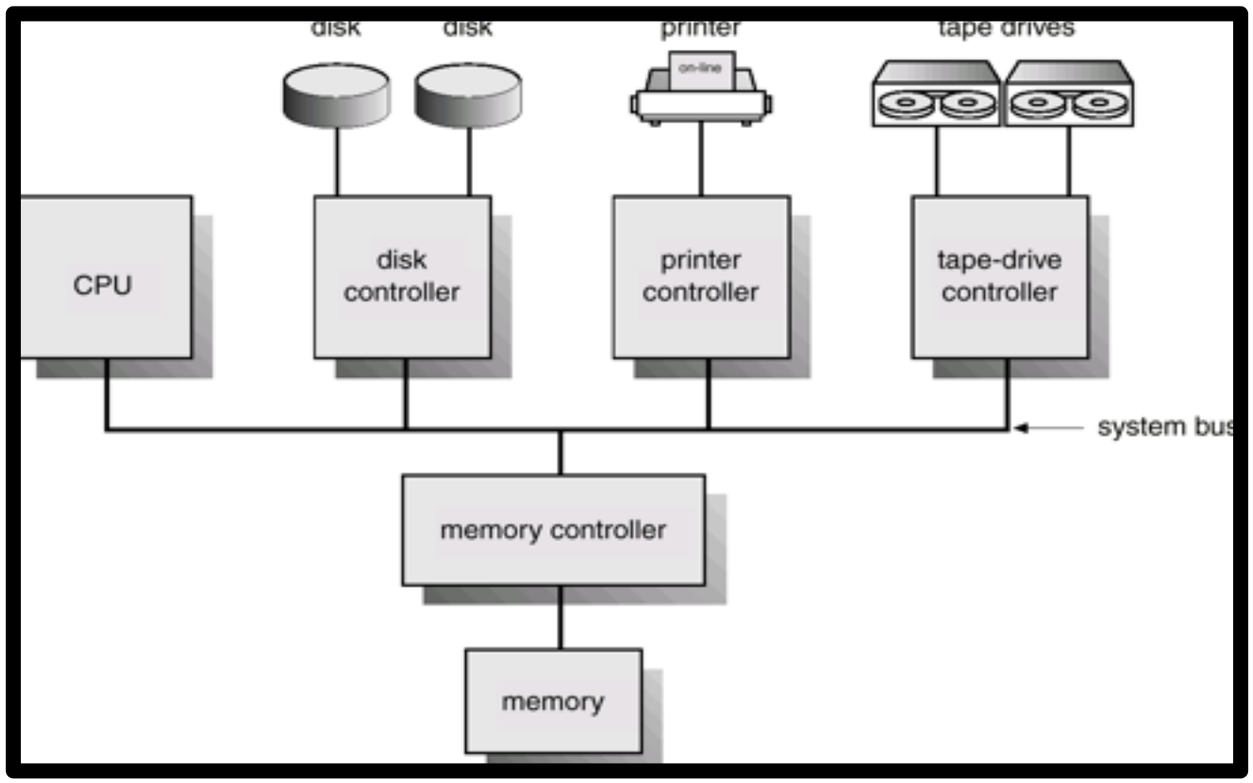
I/O Systems and Storage Devices

John Criswell
University of Rochester

I/O Device Controllers

- I/O devices have both mechanical component & electronic component
- The electronic component is the device controller
  - It contains control logic, command registers, status registers, and on-board buffer space



Device-Controller-Software Relationship

Application

User Mode Software

High-Level OS Kernel Component

Device Driver

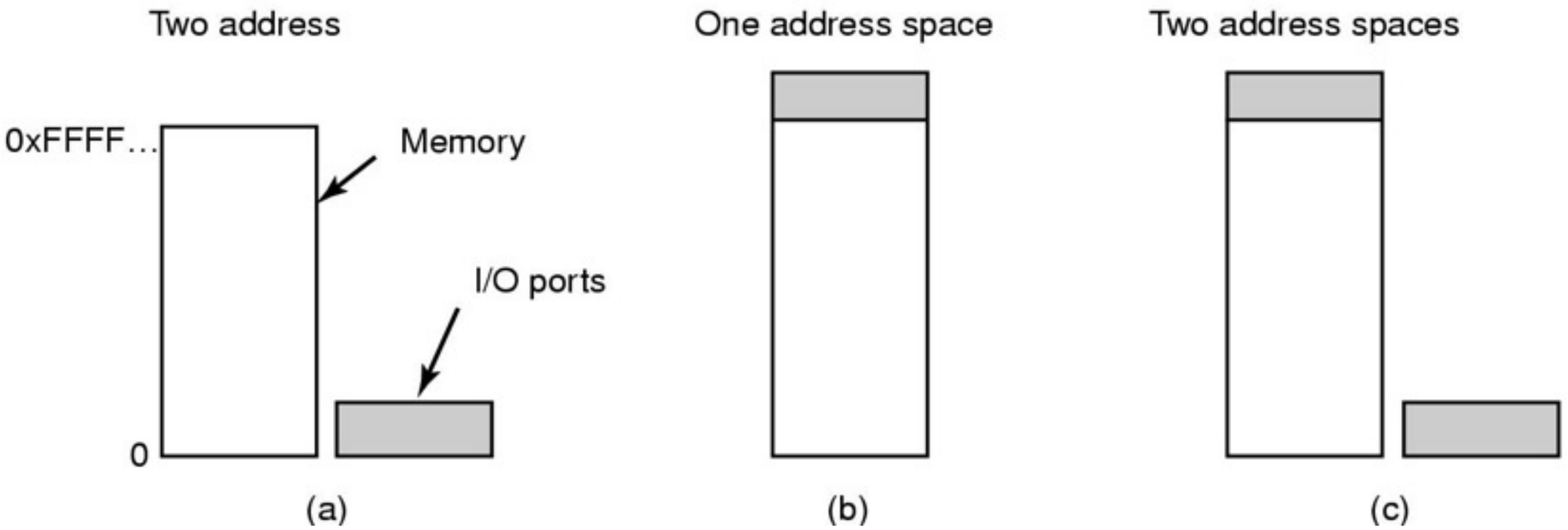
Kernel Mode Software

Device Controller

Device

Hardware

I/O Ports & Memory-Mapped I/O



- Separate I/O and memory space
  - Special I/O commands (e.g., IN/OUT)
- Memory-mapped I/O

How is I/O accomplished?

- Polling-based
  - CPU spins and polls the I/O until it completes
- Periodic polling
  - Continuous polling consumes too much CPU
  - Instead, we poll periodically - saving CPU overhead, may not react immediately to hardware events
- Interrupt-driven
  - CPU initiates I/O and then does something else
  - CPU notified when the I/O is done (interrupts)

Interrupt Handlers

1. Save registers of the old process
2. Set up context for interrupt service procedure
    1. Switch from user space to kernel space: MMU, stack, …
3. Run service procedure; when safe, re-enable interrupts
4. Run scheduler to choose the new process to run next
5. Set up context (MMU, registers) for process to run next
6. Start running the new process

Cost

What is the cost? Is it a big deal?

For Gigabit Ethernet, each packet arrives once every 12us.

How could you speed up interrupt handling?

Speeding Up Interrupt Handling

- Let OS share address space with application
  - No TLB flushes since page table remains the same
- Shadow registers
- Lazily save processor state
  - Floating point registers
  - Vector registers
  - Debug registers

Interrupt Vectors

- Intel Pentium processor event-vector table

  0: divide by zero

  6: invalid opcode

  11: segment not present

  12: stack fault

  14: page fault

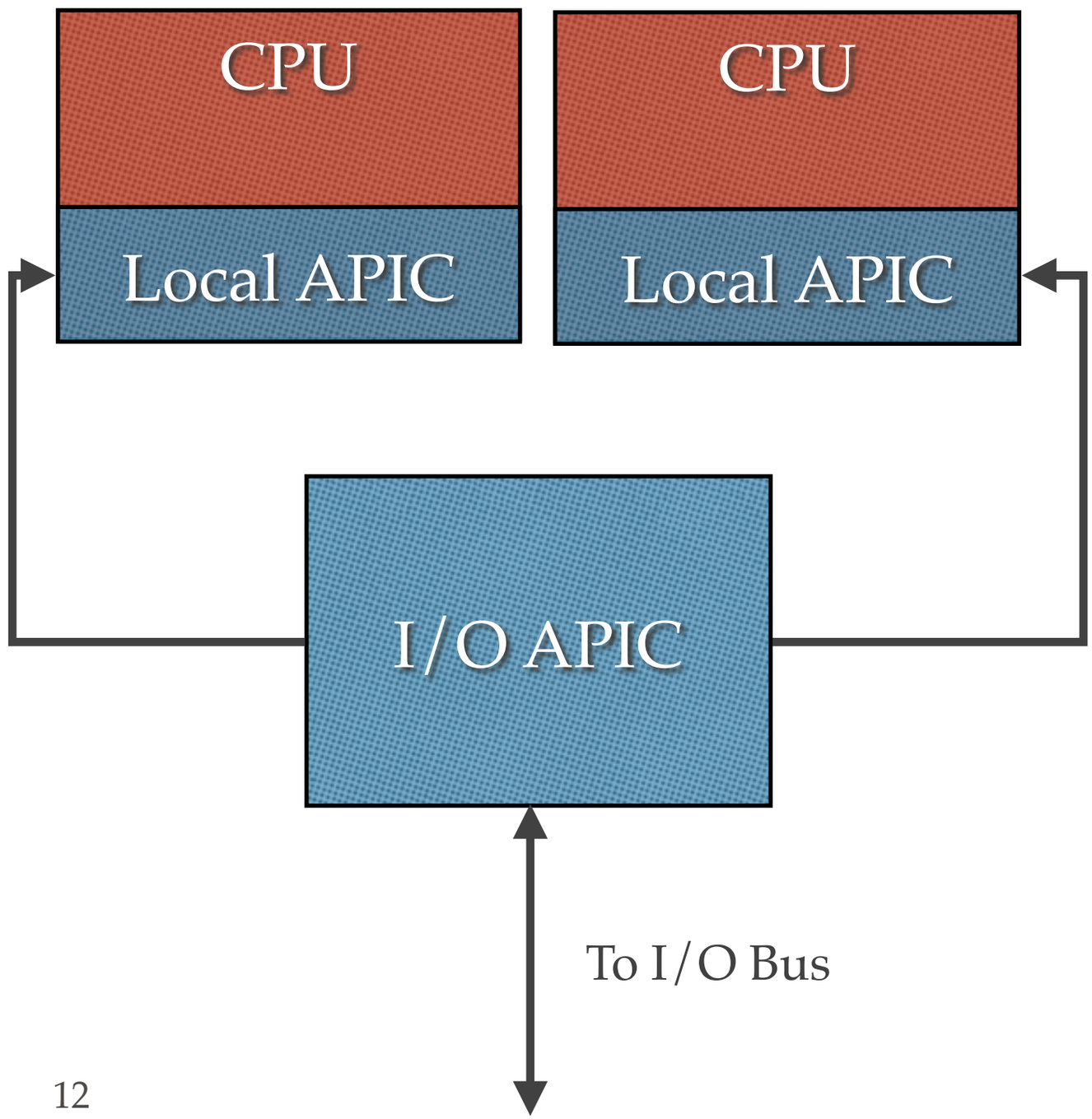
  …31: non-maskable

  32-255: maskable interrupts

Which core gets an interrupt?

Advanced Programmable Interrupt Controller (APIC)

- Local APIC
  - Timer interrupt
- I/O APIC
  - Routes interrupts to different processors
  - Balances interrupts among different processors



Device Driver Reliability

- Device driver is the device-specific part of the kernel-space I/O software; it also includes interrupt handlers
- Device drivers must run in kernel mode
  - Device driver crash brings down whole system
- Device drivers are probably the buggiest part of the OS

How could we make systems more reliable if device drivers continue to be buggy?

Improving Device Driver Reliability

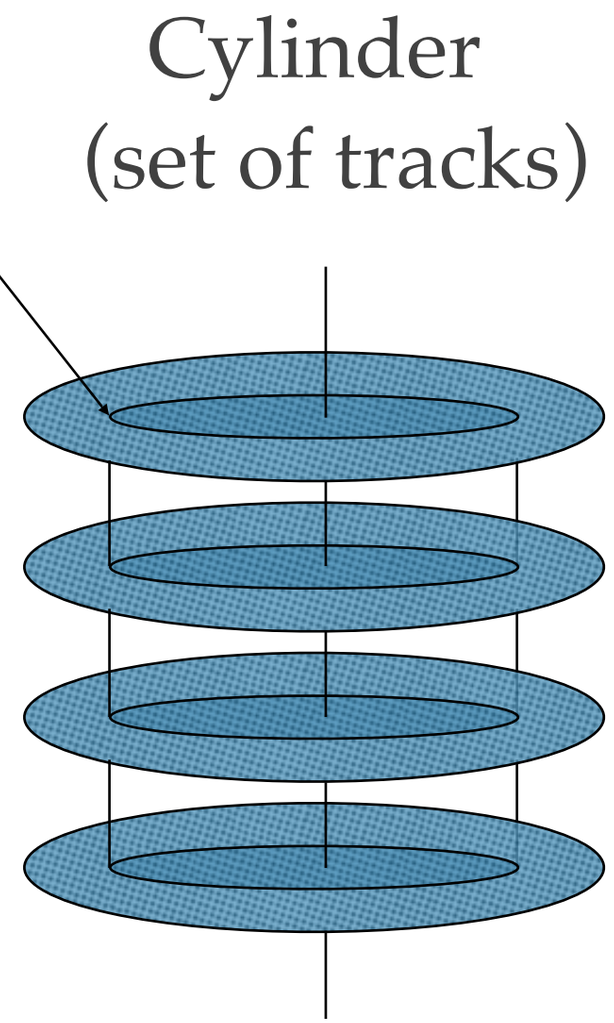
- Run most of the device driver code at user level
  - What is the cost?
- Restrict and limit device driver operations in the kernel
  - Separate address space via MMU
  - Compiler instrumentation (XFI)
  - Type-safe programming languages

To the disk drive, Robin!

Disk Drive – Mechanical Parts

Multi-surface Disk

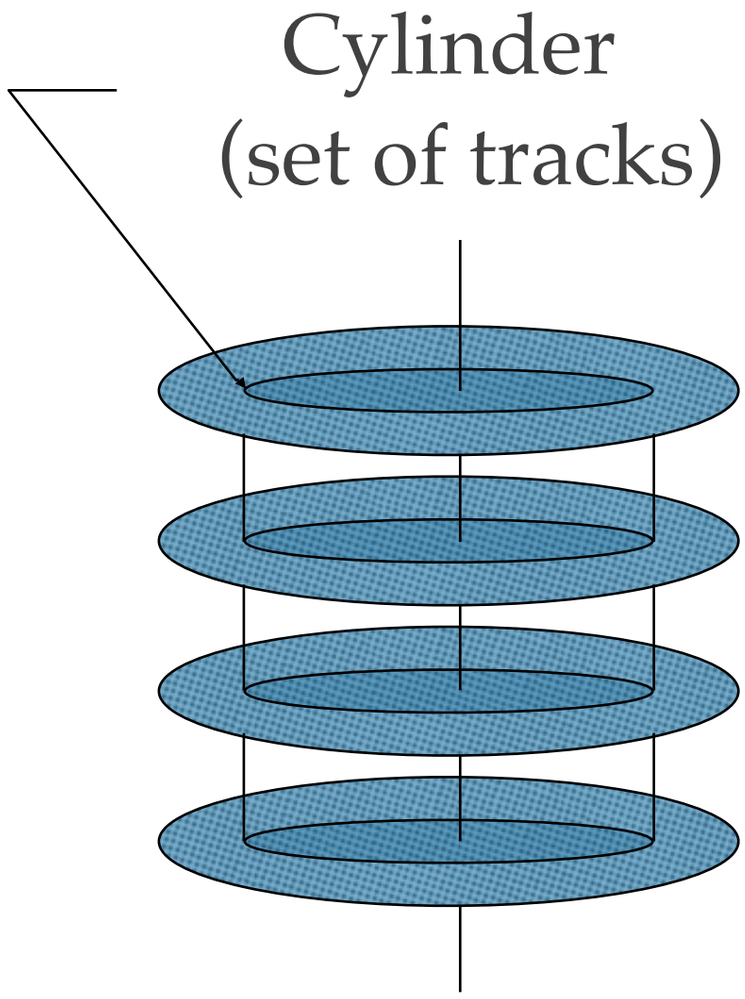
Disk Surface



Cylinders

Disk Structure

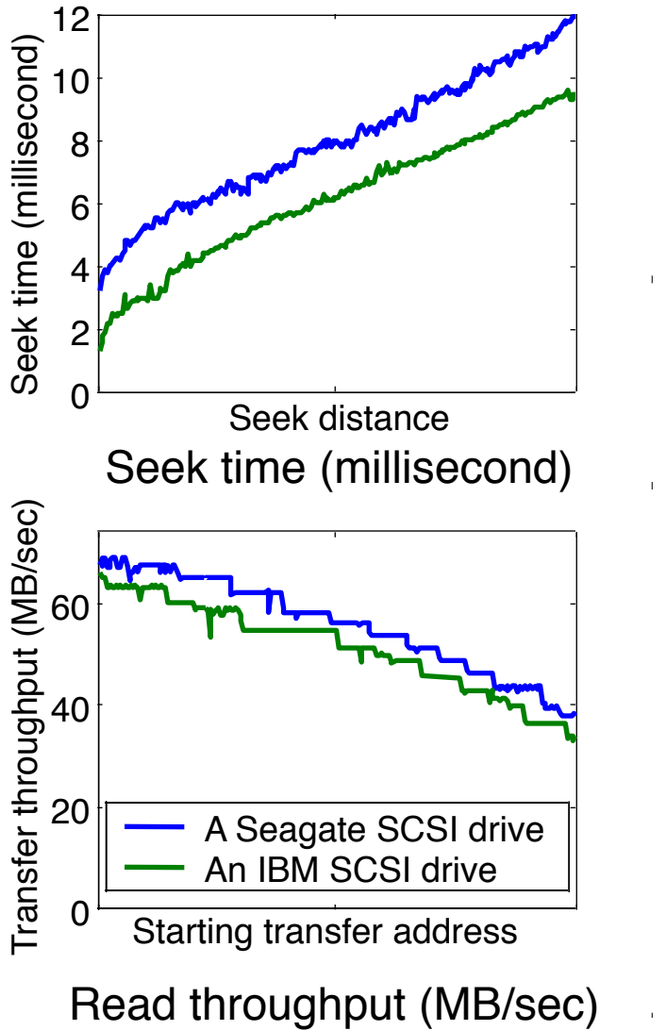
- Disk drives are addressed as large 1-dimensional arrays of logical blocks, where the logical block is the smallest unit of transfer.



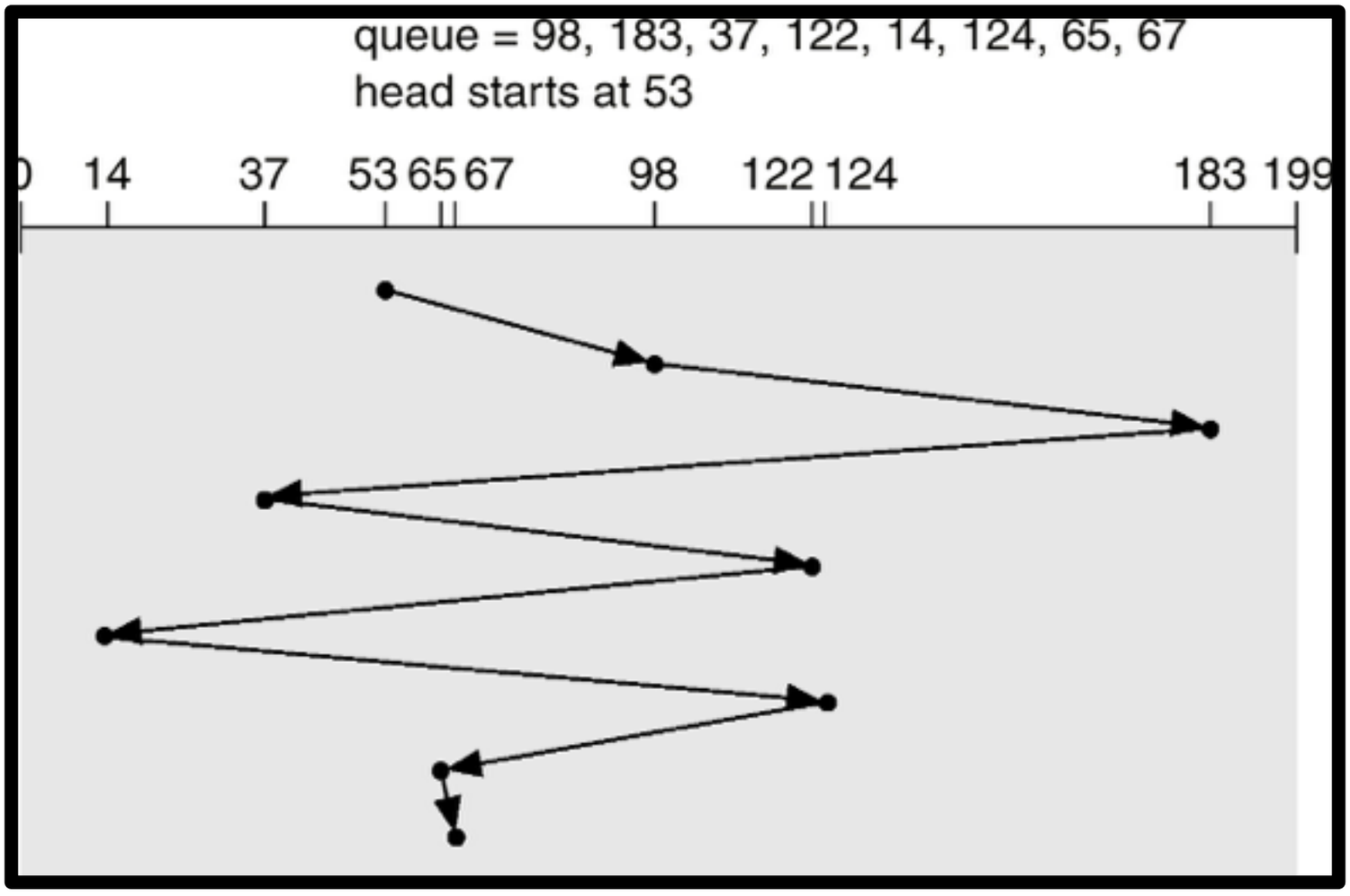
- The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially.
  - Sector 0 is the first sector of the first track on the outermost cylinder.
  - Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost.

Disk Performance Characteristics

- A disk operation has three major components
  - Seek – moving the heads to the cylinder containing the desired sector
  - Rotation – rotating the desired sector to the disk head
  - Transfer – sequentially moving data to or from disk



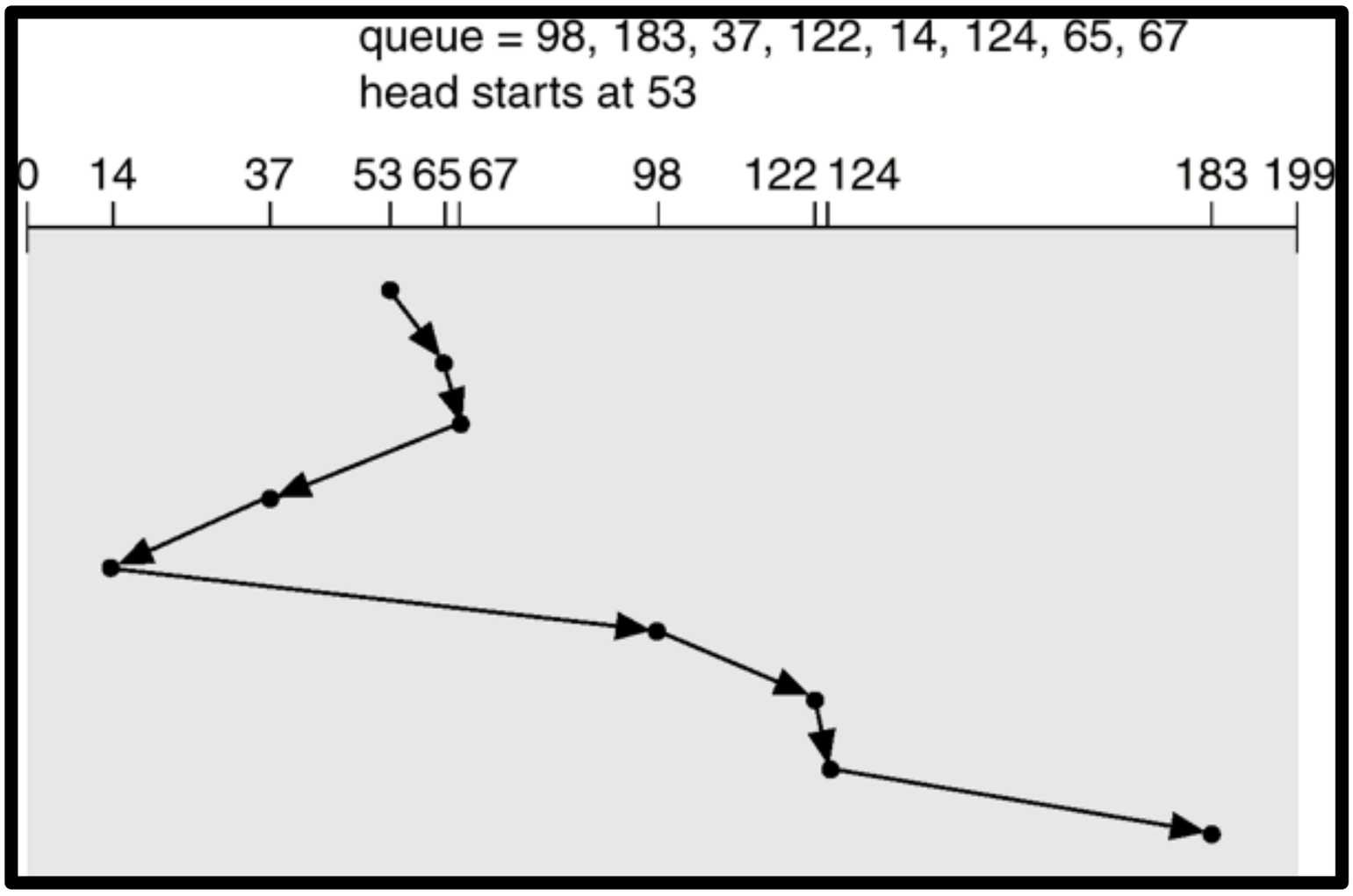
FCFS (First-Come-First-Serve)



- Illustration shows the total head movement is 640.
- Starvation?

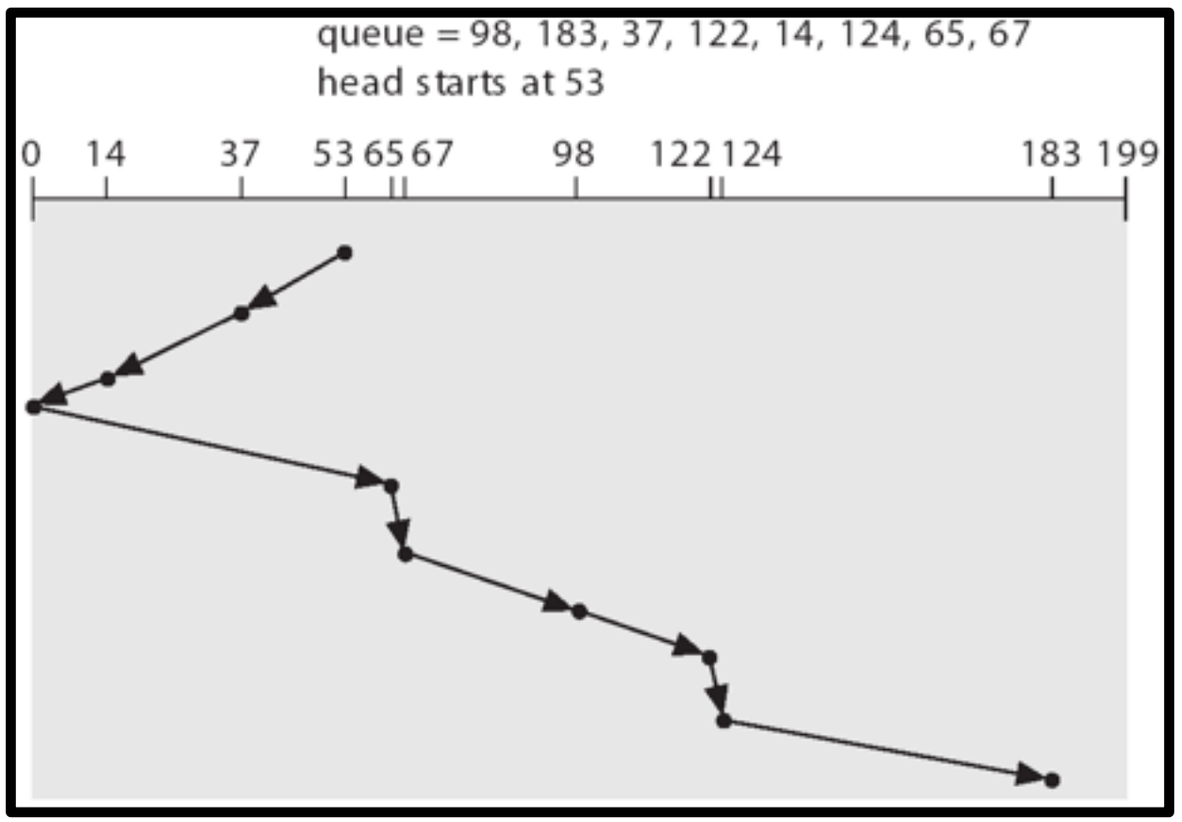
SSTF (Shortest-Seek-Time-First)

- Selects the request with the minimum seek time from the current head position.
- SSTF scheduling is a form of SJF scheduling.
- Illustration shows the total head movement is 236.



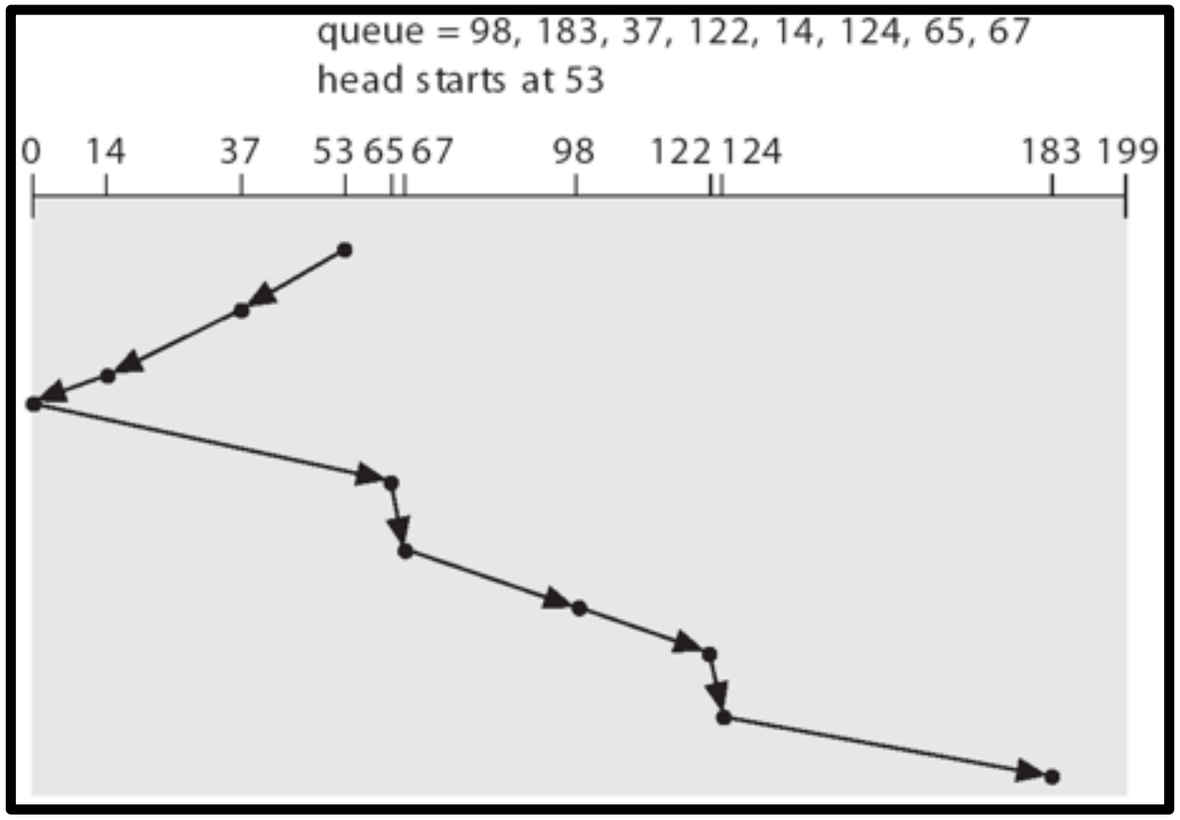
SCAN

- The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end, where the head movement is reversed and servicing continues.
- Sometimes called the ***elevator algorithm***.
- Illustration shows the total head movement is 208.



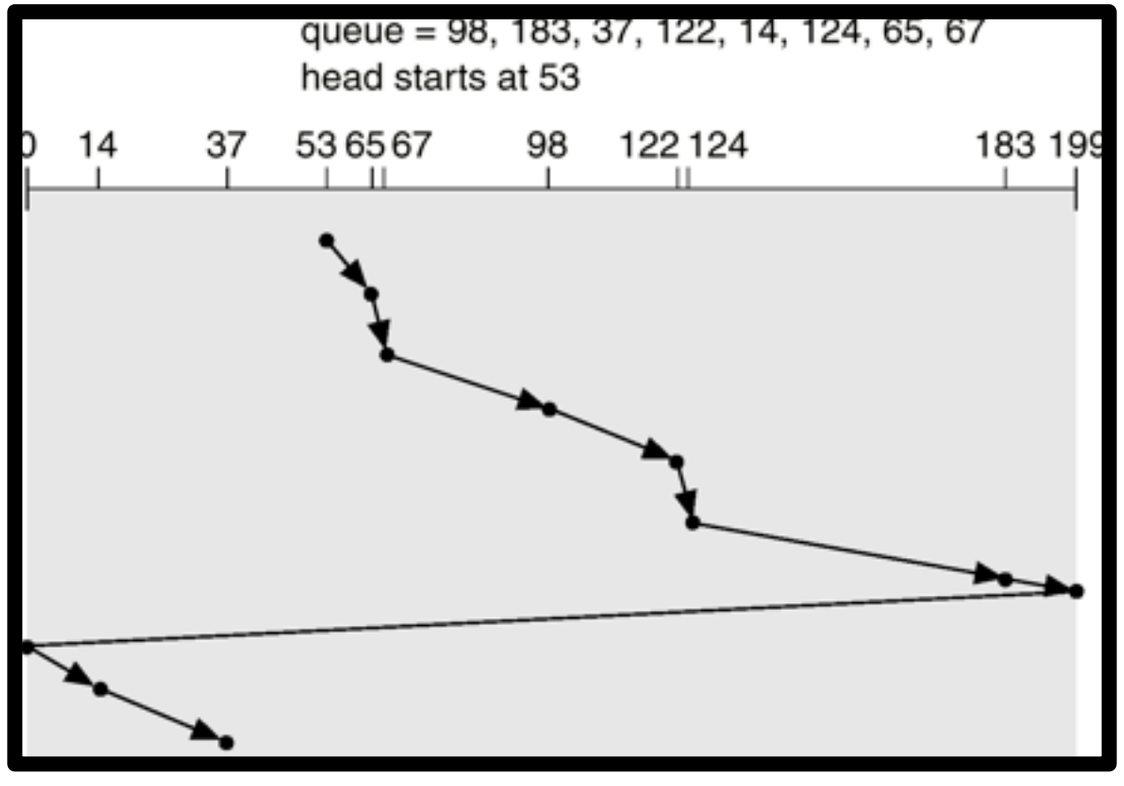
SCAN

- Disk arm moves back and forth along disk
  - Reads data as it goes
- Sometimes called the ***elevator algorithm***.
- Illustration shows the total head movement is 208.



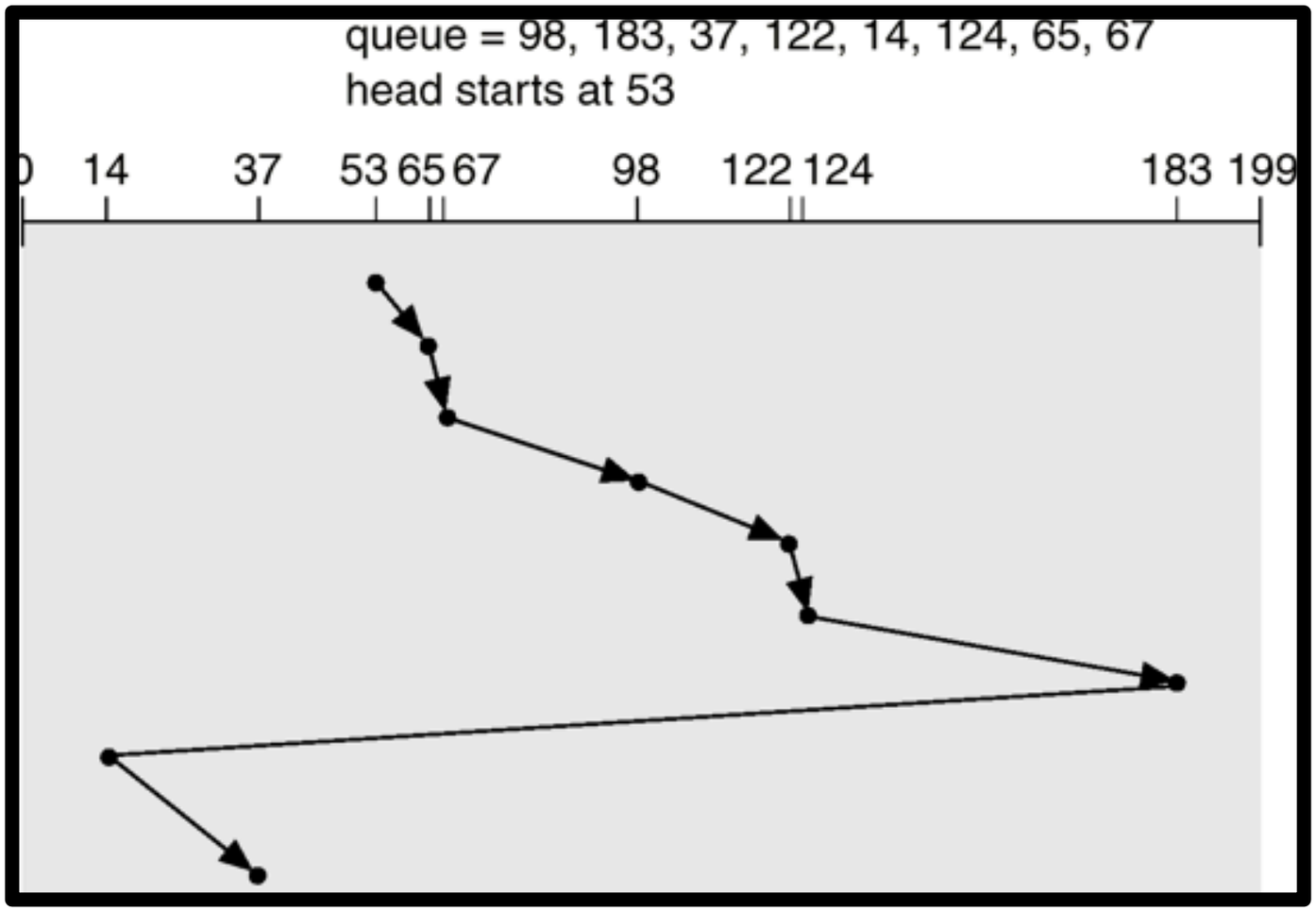
C-SCAN (Circular-SCAN)

- Provides a more uniform wait time than SCAN.
- The head moves from one end of the disk to the other, servicing requests as it goes. When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip.



C-LOOK

- Variation of C-SCAN
- Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk.



Deadline Scheduling in Linux

- A regular elevator-style scheduler similar to C-LOOK
- Additionally, all I/O requests are put into a FIFO queue with an expiration time (e.g., 500ms)
- When the head request in the FIFO queue expires, it will be executed next (even if it is not next in line according to C-LOOK).
- A mix of performance and fairness.

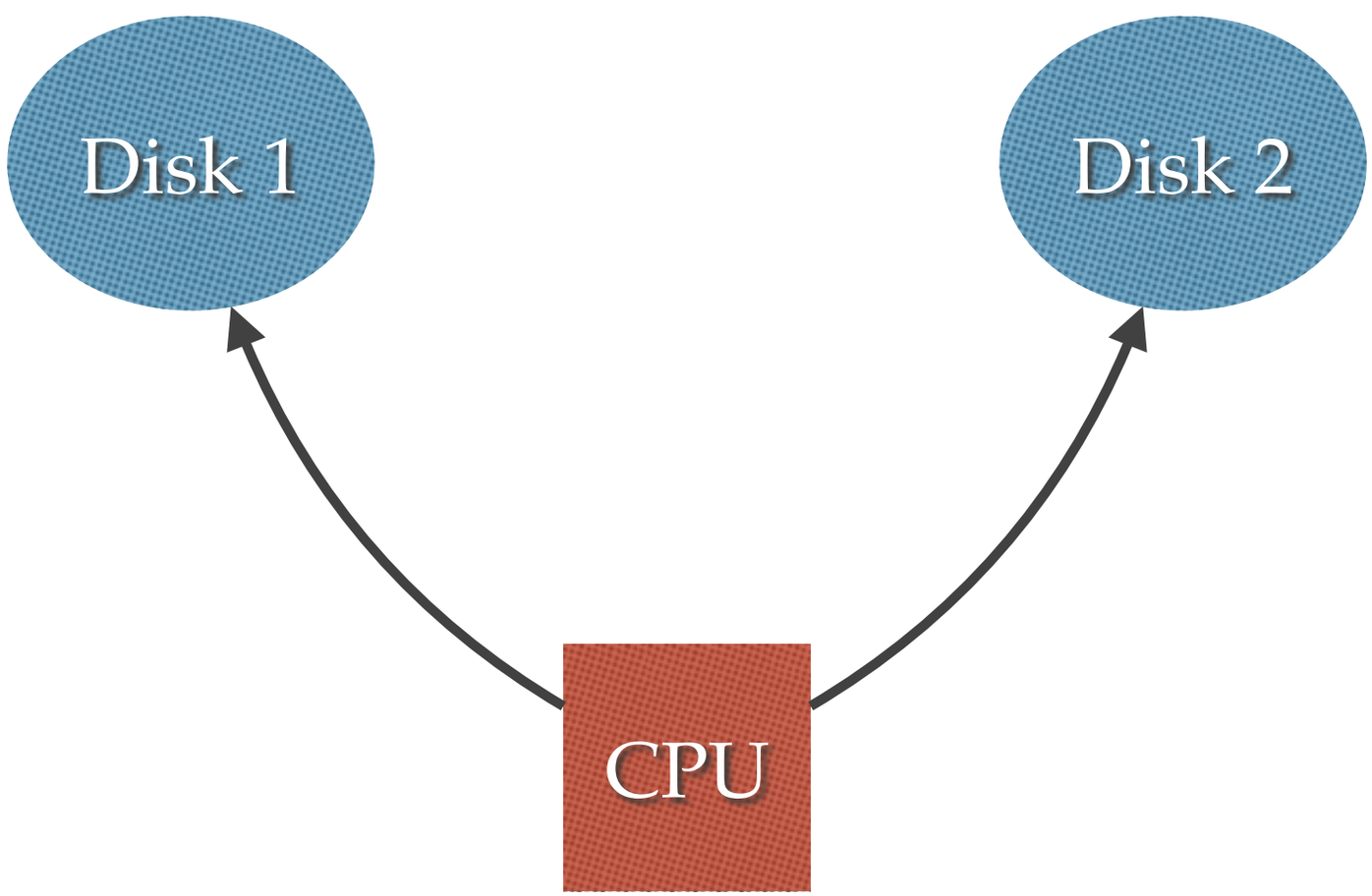
RAIDers of the Lost Data

Disk Concurrency

- "The only thing better than one guitar is two guitars."
  - Maurice Ravel
- The only thing better than one disk is two disks.
  - David Patterson, Garth A. Bison, and Randy Katz

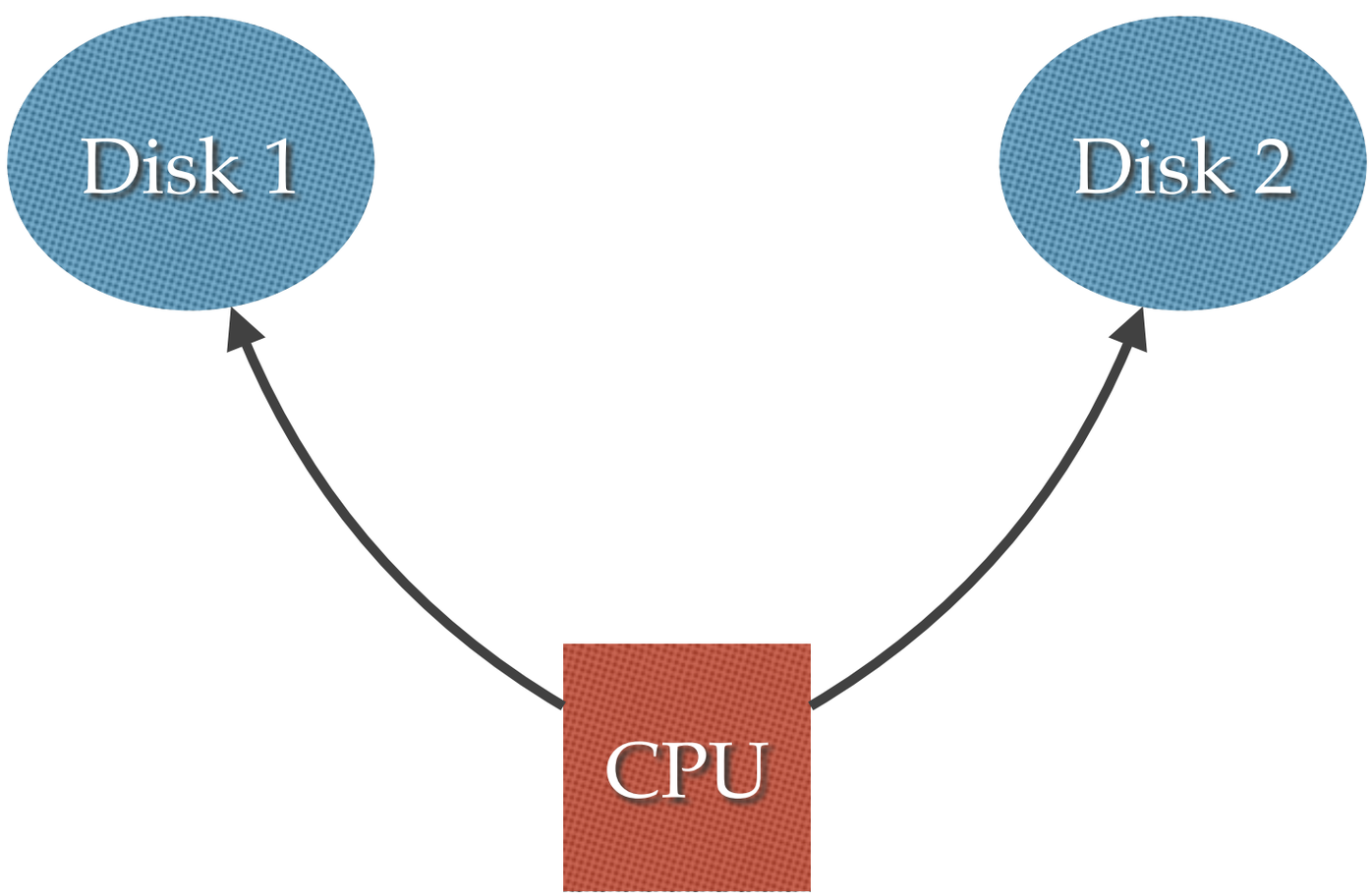
Two Disks: Disk Striping

- Blocks divided into subblocks
- Subblocks stored on different disks



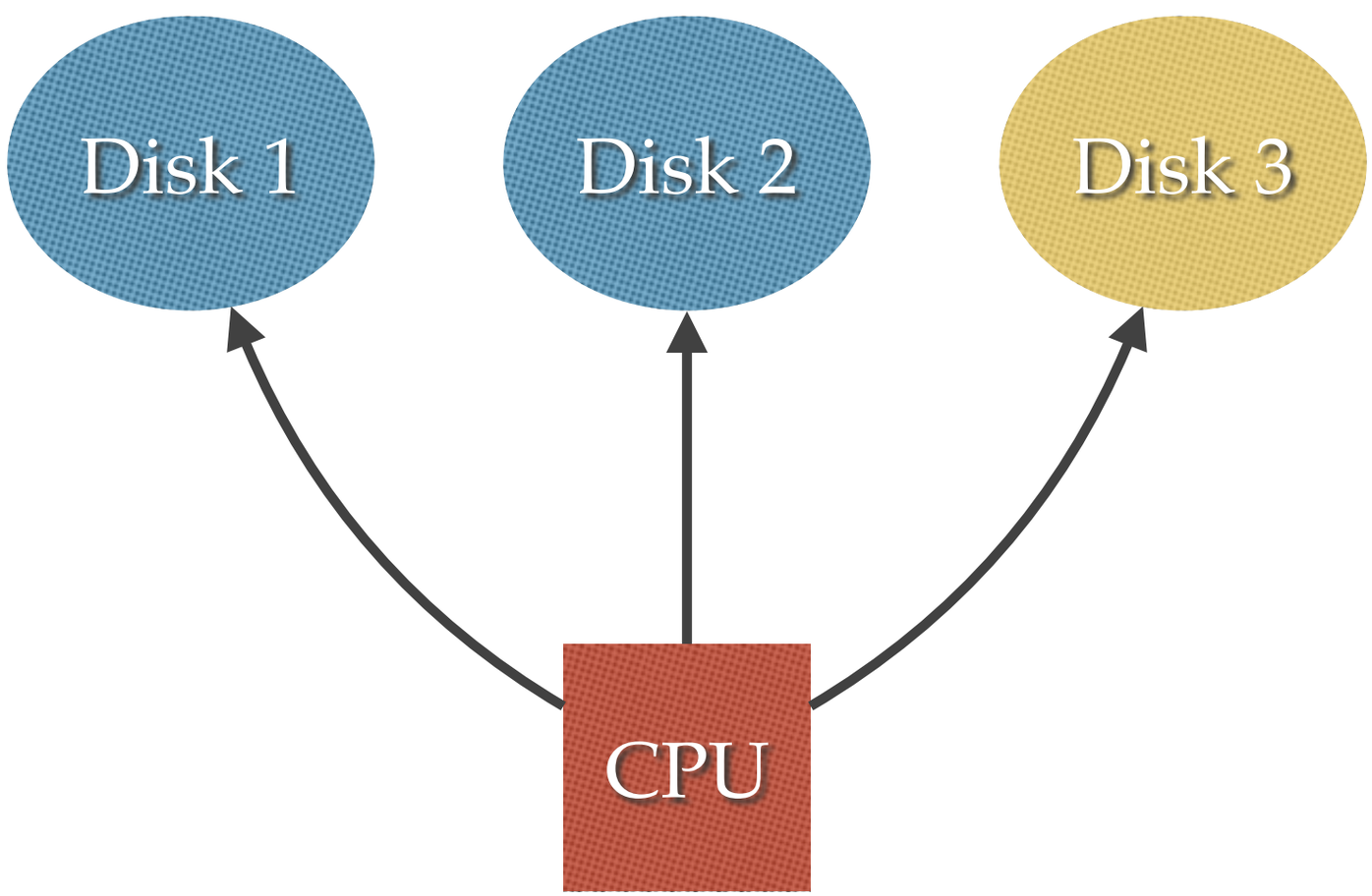
Two Disks: Mirroring

- Make a copy of each block on each disk
- Provides redundancy



Multiple Disks: Parity Block

- Have one disk contain parity bits of blocks on other devices
- Provides redundancy without full copy



Exploiting Concurrency

- RAID: Redundant Arrays of Independent Disks
  - RAID 0: data striping at block level, no redundancy
  - RAID 1: mirrored disks (100% overhead)
  - RAID 2: bit-level striping with parity bits, synchronized writes
  - RAID 3: data striping at the bit level with parity disk, synchronized writes
  - RAID 4: data striping at block level with parity disk
  - RAID 5: scattered parity
  - RAID 6: handles multiple disk failures

Disk Management

- Formatting
  - Header: sector number etc.
  - Footer/tail: ECC codes
  - Gap
  - Initialize mapping from logical block number to defect-free sectors
- Logical disk partitioning
  - One or more groups of cylinders
  - Sector 0: master boot record loaded by BIOS firmware, which contains partition information
  - Boot record points to boot partition

Swap Space Management

- Part of file system?
  - Requires navigating directory structure
  - Disk allocation data structures
- Separate disk partition
  - No file system or directory structure
  - Optimize for speed rather than storage efficiency
  - When is swap space created?

Credits

- Parts of the lecture slides contain original work of Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Andrew S. Tanenbaum, and Gary Nutt. The slides are intended for the sole purpose of instruction of operating systems at the University of Rochester. All copyrighted materials belong to their original owner(s).