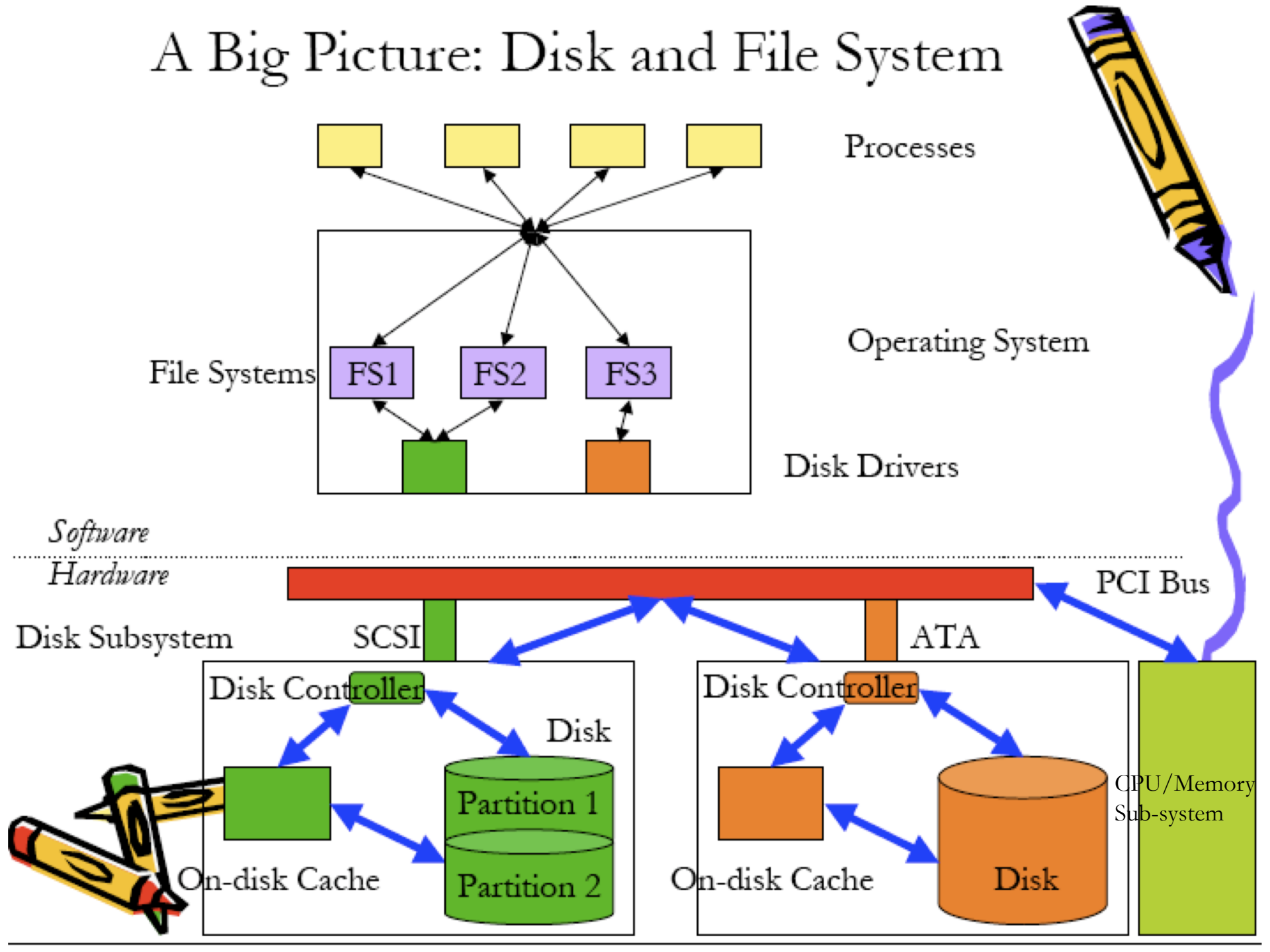
Operating Systems Principles

I/O Management (2) – Disks and File Systems

Slides are adapted from Prof. Bhuvan Urgaonkar

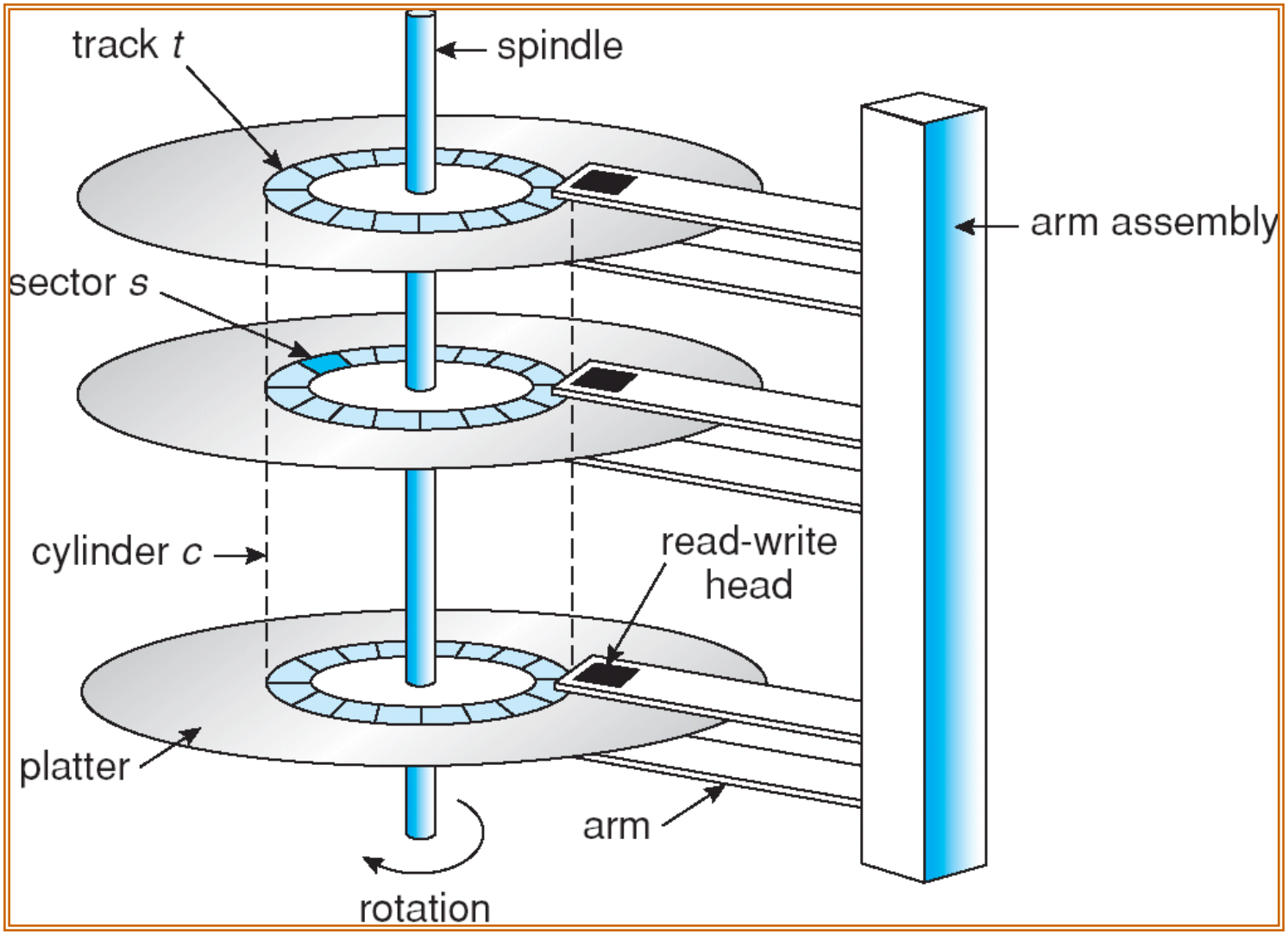
A Big Picture: Disk and File System



Disks and File Systems

- We will focus on certain salient aspects of disk IO management as a case study of IO management
- First, let's understand some basics of the IO device itself
- Then, we will study an important OS part that manages disks – the file system
  - We won't get into the device driver in this course in any detail

Moving-head Disk Mechanism




© explainthatstuff.com 2010
Some rights reserved

1. Actuator (compact electric motor that moves the read-write arm).
2. Read-write arm swings read-write head back and forth across platter.
3. Central spindle allows platter to rotate at high speed.
4. Magnetic platter stores information in binary form.
5. Plug connections link hard drive to circuit board in personal computer.
6. Read-write head is a tiny magnet on the end of the read-write arm.
7. Circuit board on underside controls the flow of data to and from the platter.
8. Flexible connector carries data from circuit board to read-write head and platter.
9. Small spindle allows read-write arm to swing across platter.

Overview of Mass Storage Structure

- Magnetic disks provide bulk of secondary storage of modern computers
- Drives rotate at 7200 to 15000 times per minute (RPM)
  - 120 to 240 times per second
- **Positioning Time** (**random-access time**)
  - Seek time: time to move disk arm to desired cylinder
  - Rotational Latency: time for desired sector to rotate under the disk head
- Transfer Time: Time to read data
- Request service time = Positioning time + Data Transfer Time

Overview of Mass Storage Structure

- Drive attached to computer via **I/O bus**
  - Busses vary, including **EIDE, ATA, SATA, USB, Fibre Channel, SCSI**
  - **Host controller** in computer uses bus to talk to **disk controller** built into drive or storage array

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.

Growing Popularity of Flash Disks

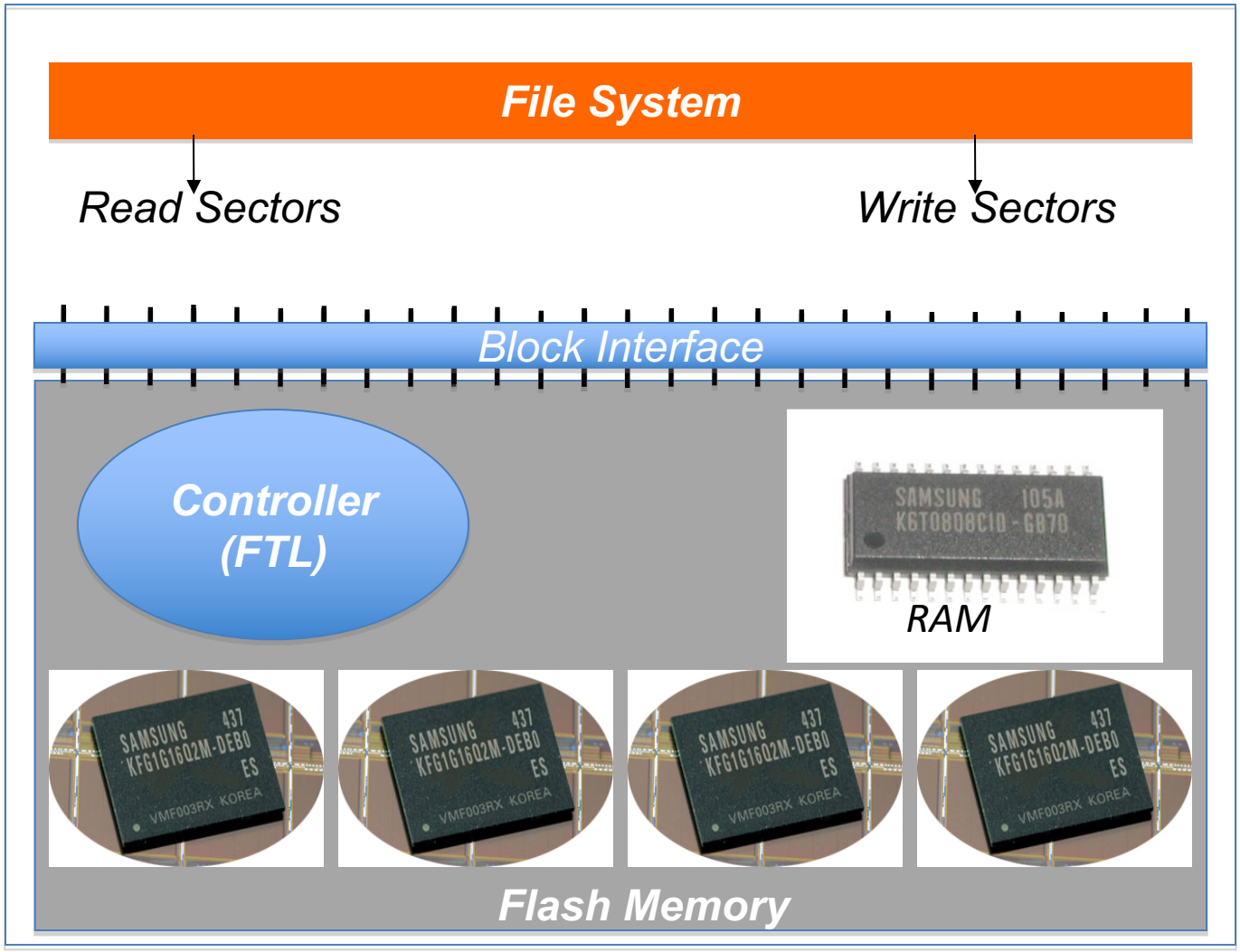
- **"The NAND market has grown faster than any technology in the history of semiconductors, exceeding $11 billion in 2006, only a decade after its introduction "** – Jim Handy, Objective Analysis
- Embedded storage: PDAs, mobile phones, digital cameras, digital music players
- Desktop storage: MacBook Air, One Laptop Per Child (OLPC), game consoles, Intel's X25-E Extreme SATA Solid-State Drive
- Enterprise scale storage: Texas Memory System's RamSan-500, Fusion-io's ioDrive, Symmetrix DMX-4 from EMC


TOSHIBA
Solid State Drive
128GB

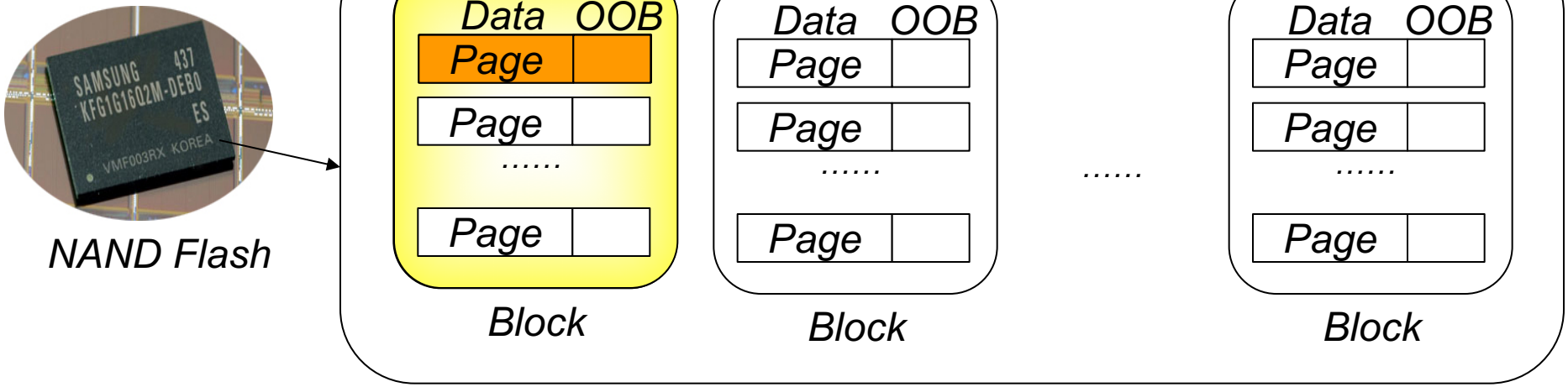
Why Flash is Likely to be Around

- Benefits over magnetic hard drives
  - Semi-conductor technology, no mechanical parts
  - Offers lower and more predictable access latencies
    - Microseconds (45us Reads / 200us Writes) vs milliseconds for magnetic
  - Lower power consumption
  - Higher robustness to vibrations and temperature
- Dropping prices
  - Some projections suggest comparable with magnetic sooner or later
  - Likely to offer significant cost advantage over other emerging technologies (e.g., PCM) for a decade or more

Flash Solid State Drive (SSD)

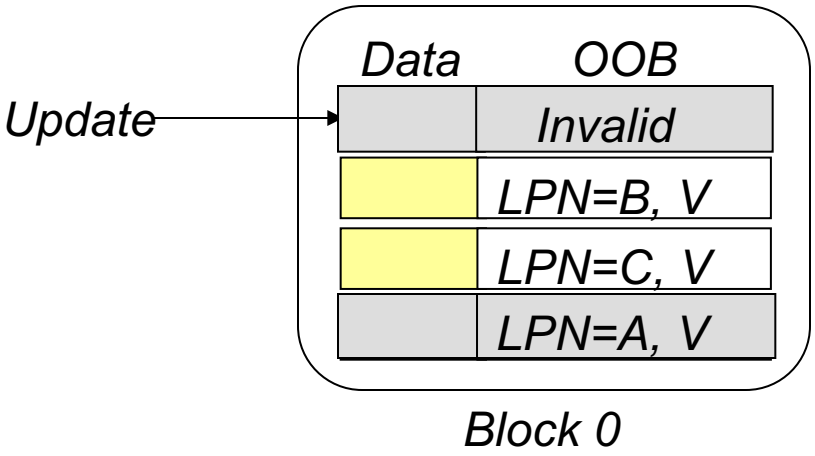


Basics of NAND Flash Memory



- Three operations: read, write, erase
- Reads and writes are done at the granularity of a **page** (2KB or 4KB)
- Erases are done at the granularity of a **block**
  - Block: A collection of physically contiguous pages (64 or 128)
  - Block erase is the **slowest operation** requiring about 2ms
- **Writes can only be done on erased pages**

Out-of-Place Updates



LPN	PPN (PBN, Offset)
A	(0, 3)
B	(0, 1)
C	(0, 2)

Flash Mapping Table

- Over-writes on the same location (page) are expensive
- Updates are written to a free page
- OOB area
  - Keeps valid/free/invalid status
  - Stores LPN, used to reconstruct mapping table upon power failure

Garbage Collection

- Reclaims invalid pages
- Typically, called when free space falls below a *threshold*
- Victim block selection
  - Small # valid pages (reduce copying overhead)
  - Small # overall erases (wear level)

Flash Translation Layer (FTL)

- Flash Translation Layer
  - Emulates a normal block device interface
  - Hides the presence of erase operation/erase-before-write
  - Address translation, garbage collection, and wear-leveling
- Address Translation
  - Mapping table present in small RAM within the flash device

Disk Scheduling

- Ordering of requests issued to disk
  - In controller: order requests executed by disk arm
  - Important: Done at multiple places in s/w as well
    - E.g., Device driver may order requests sent to controller
- Typical goal: Minimize seek time (our focus)
  - Seek time dependent on seek distance
- More advanced
  - Incorporate rotational latency as well
  - Incorporate notions of fairness

Disk Scheduling (Cont.)

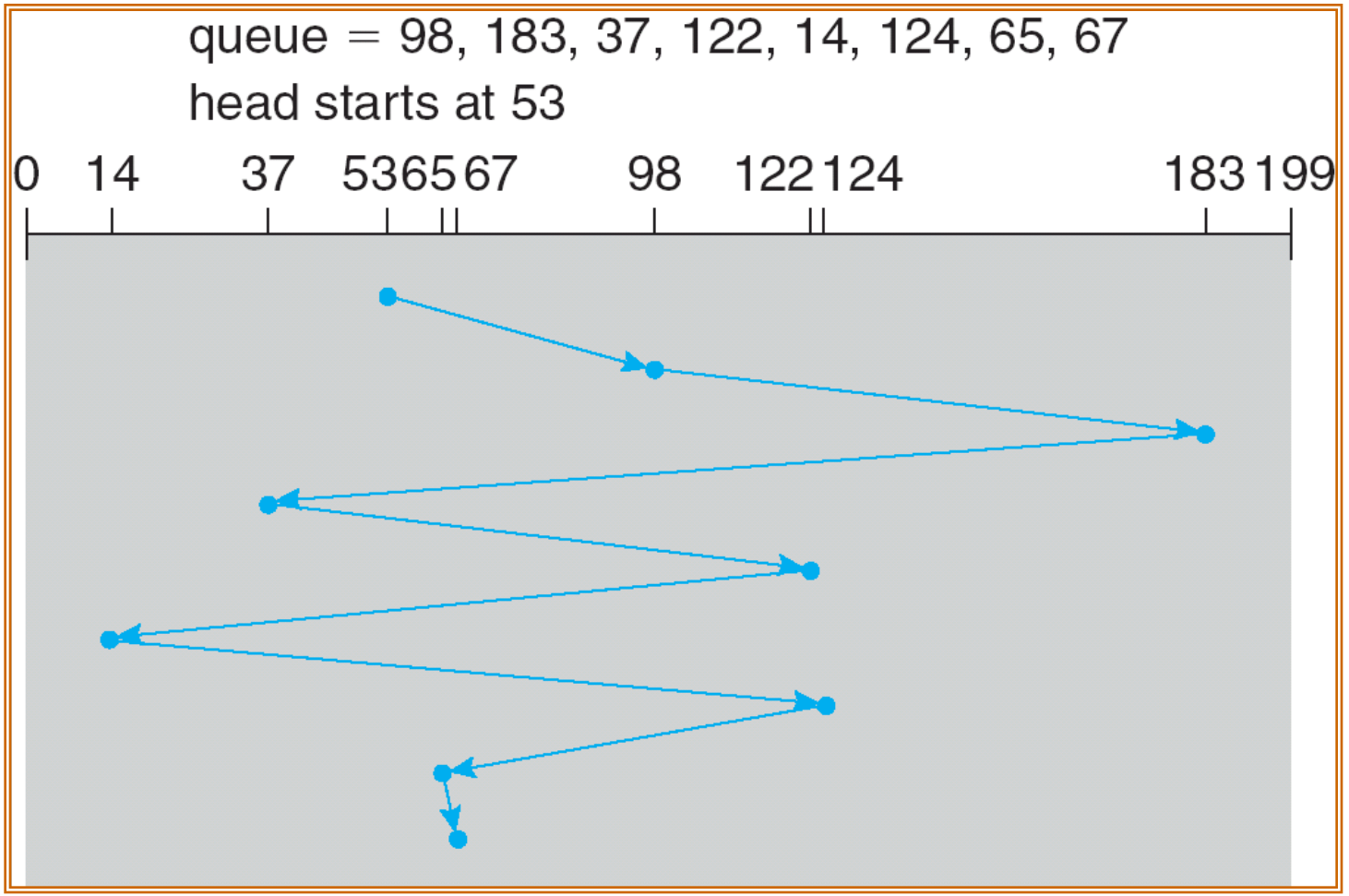
- Several algorithms exist to schedule servicing of disk I/O requests.
- We illustrate them with a request queue (0-199).

98, 183, 37, 122, 14, 124, 65, 67

Head pointer 53

FCFS

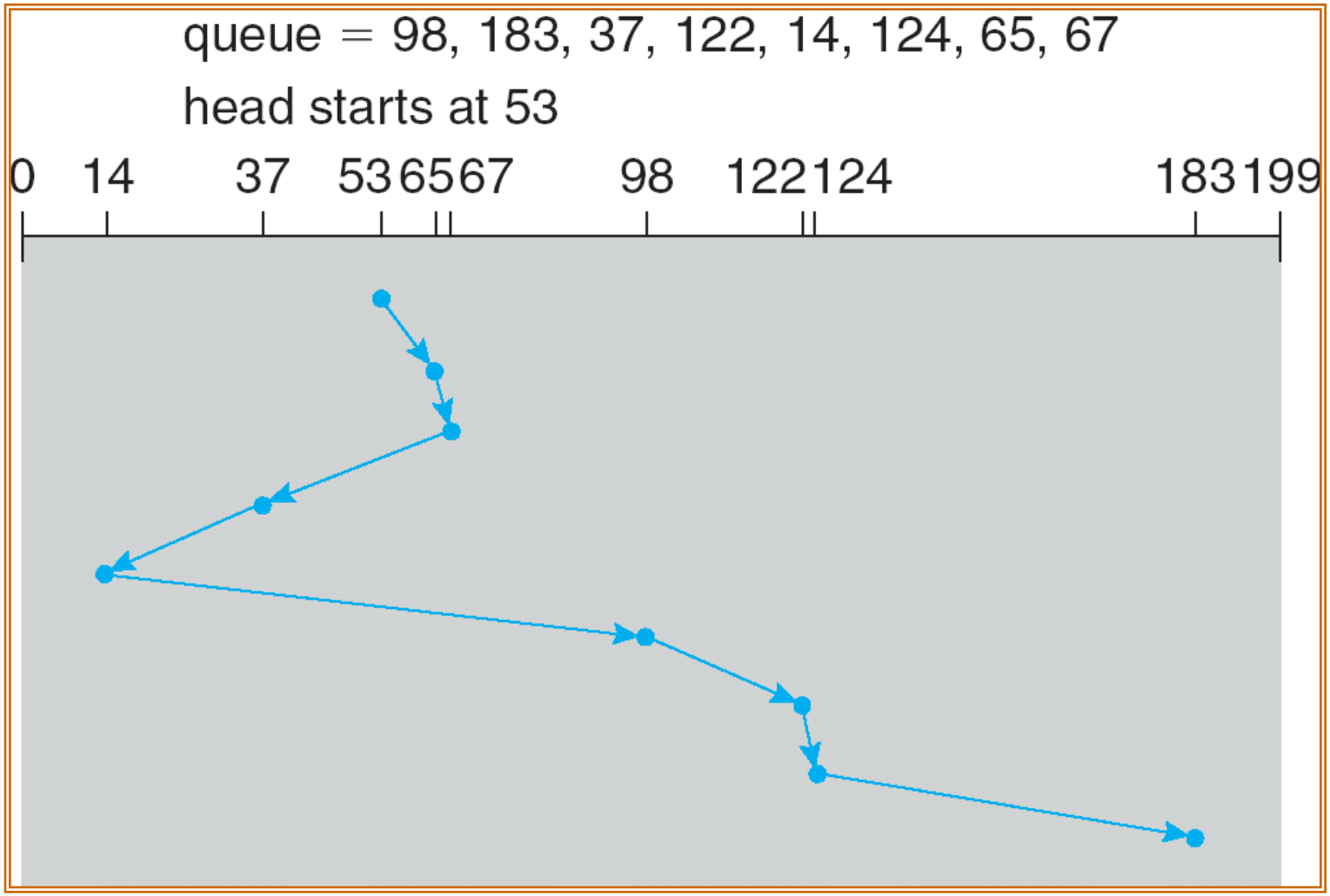
Illustration shows total head movement of 640 cylinders.



SSTF

- Selects the request with the minimum seek time from the current head position.
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests.
- Illustration shows total head movement of 236 cylinders.

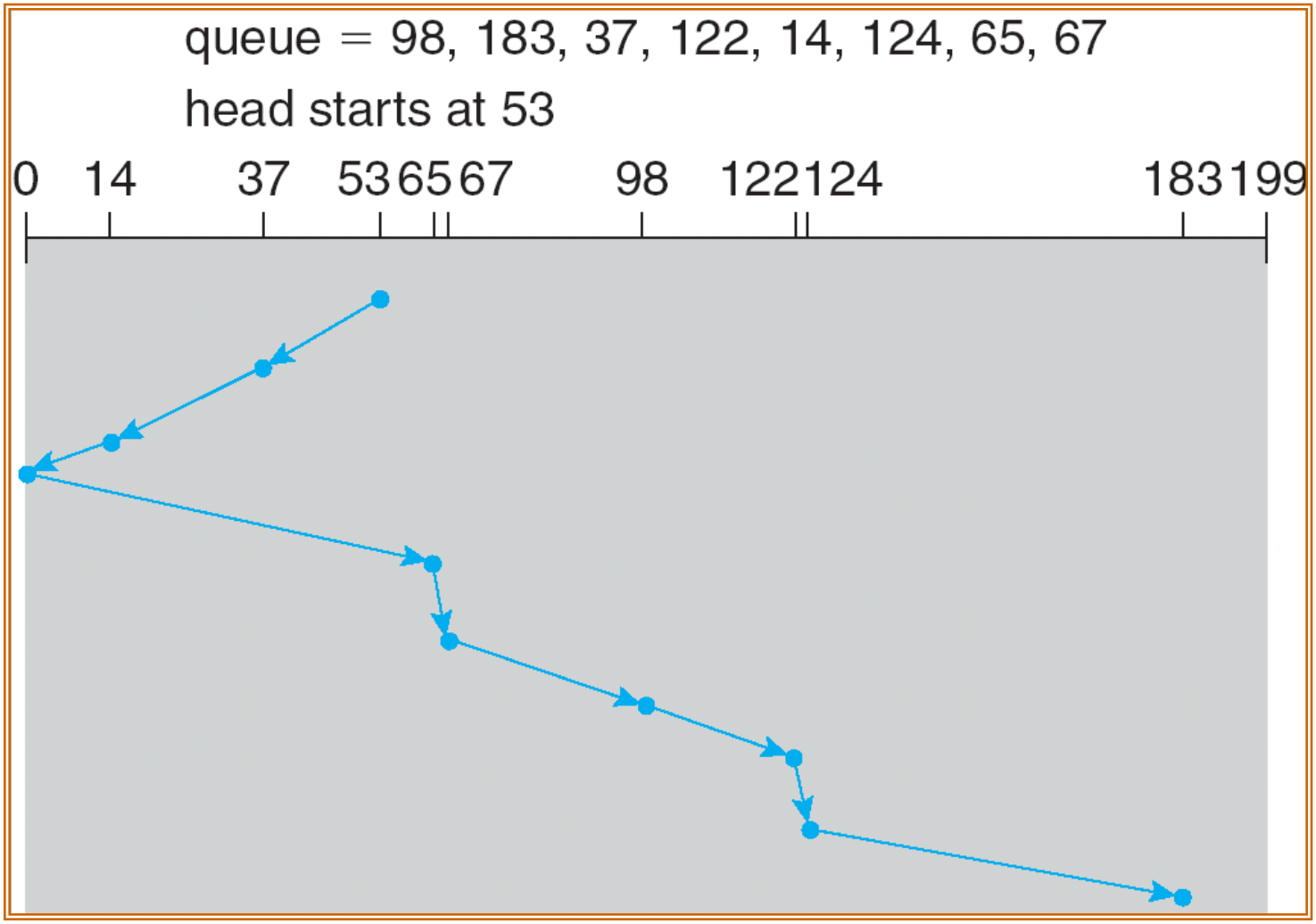
SSTF (Cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

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 of the disk, where the head movement is reversed and servicing continues.
- Sometimes called the *elevator algorithm*.
- Illustration shows total head movement of 208 cylinders.

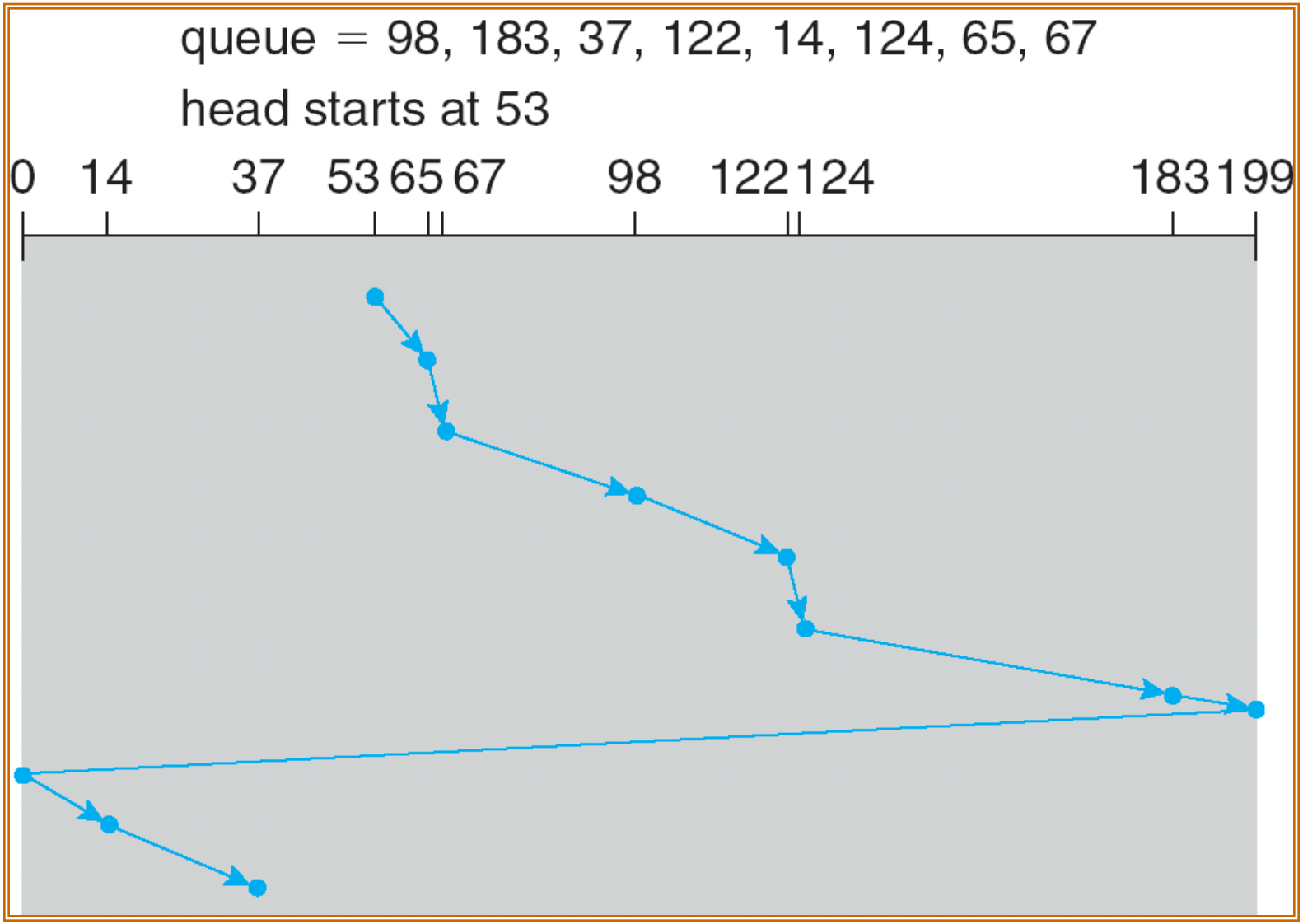
SCAN (Cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

C-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.
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one.

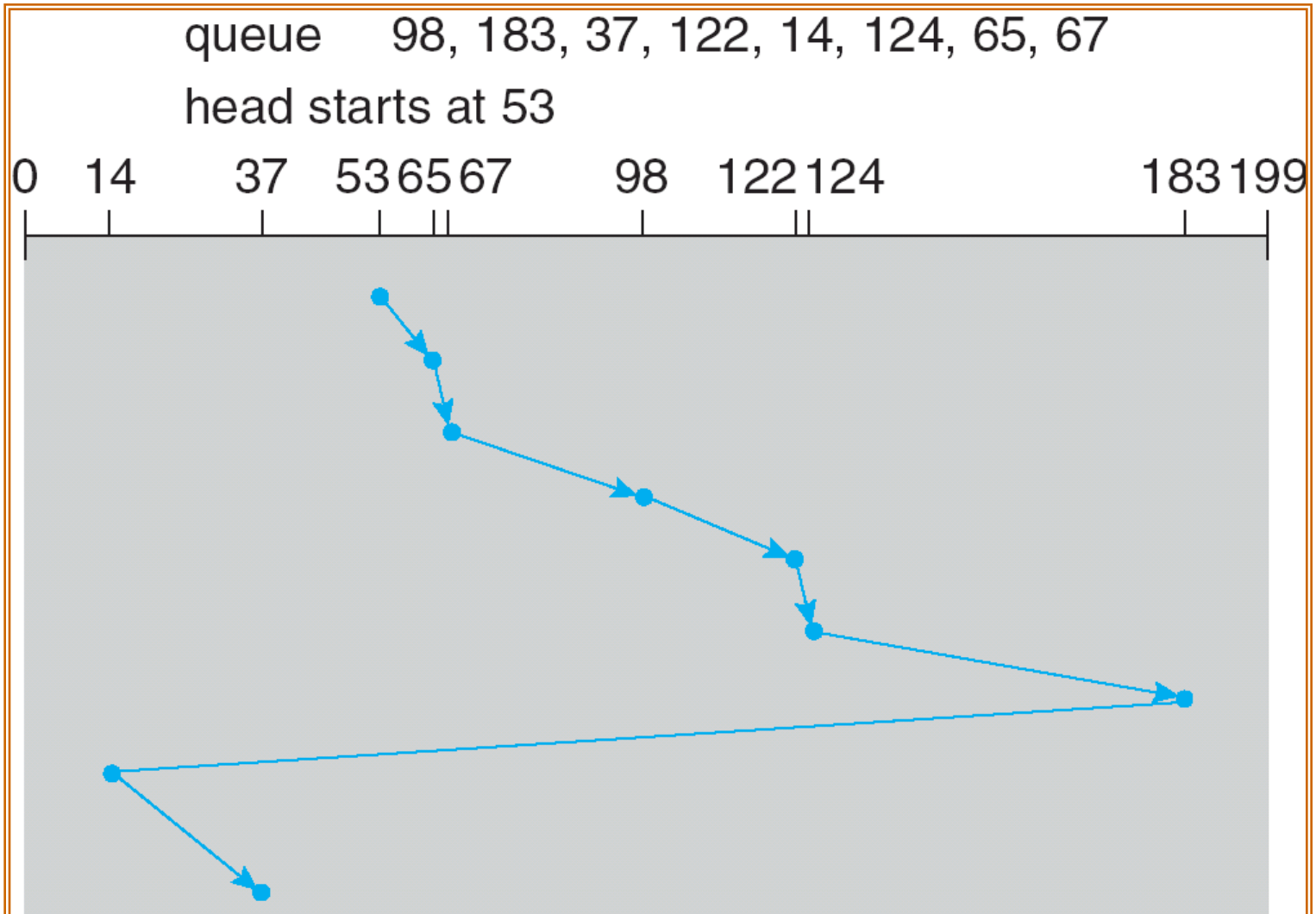
C-SCAN (Cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

C-LOOK

- Version 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.

C-LOOK (Cont.)


queue 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

Disk properties and implications on OS management

- Significantly slower than CPU/memory
  - Buffering will be crucial for speed mismatch
- Seek latency high -> large, contiguous data xfers preferable
  - Coalesce requests in buffers
  - Reorder them to improve “sequentiality”