# EECS 111:

# System Software

# Lecture: Mass-Storage Systems


## Prof. Mohammad Al Faruque

**The Henry Samueli School of Engineering**
**Electrical Engineering & Computer Science**
**University of California Irvine (UCI)**

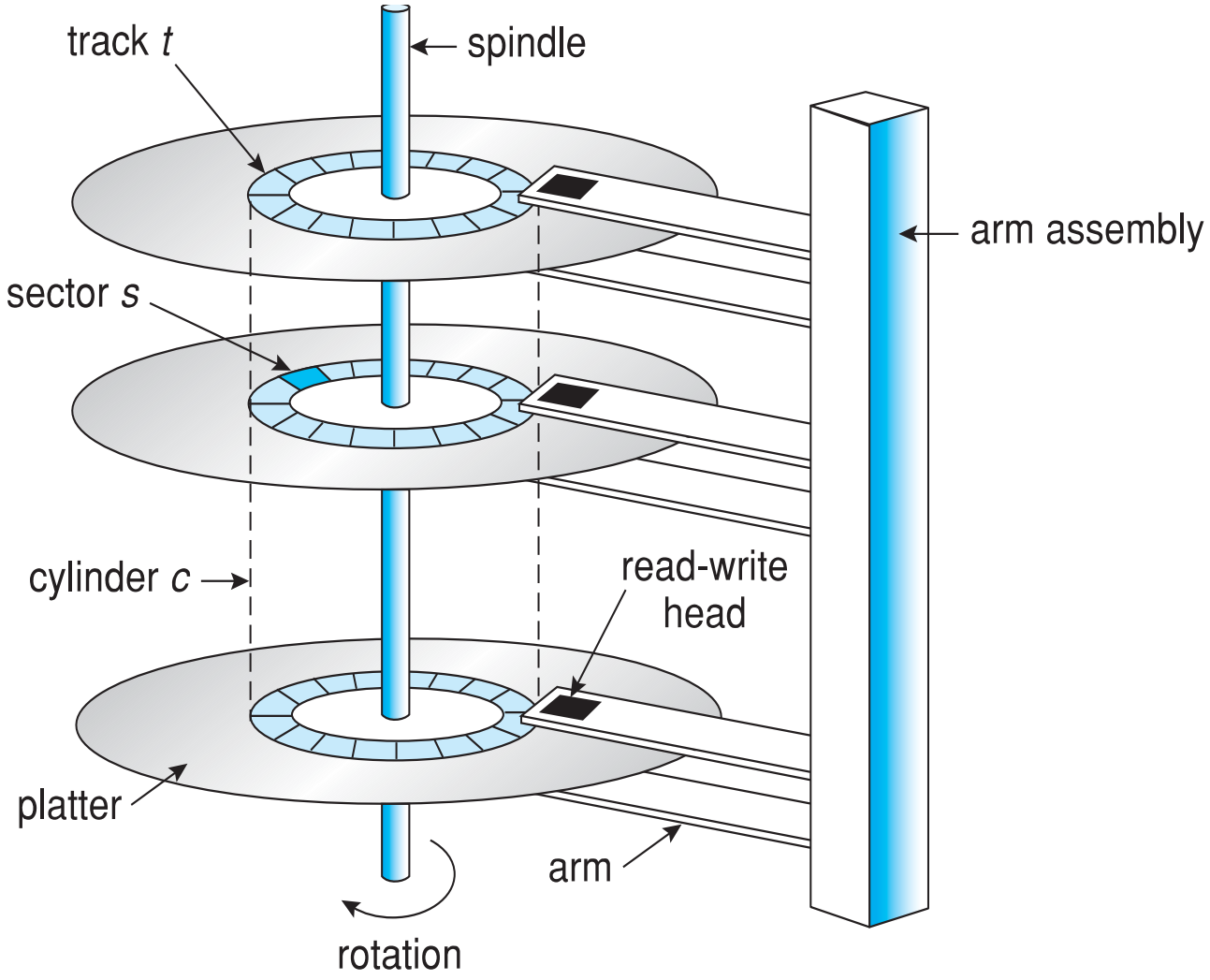
# Mass-Storage Systems

- Overview of Mass Storage Structure
- Disk Structure
- Disk Scheduling
- Disk Management
- Swap-Space Management

# Overview of Mass Storage Structure

- **Magnetic disks provide bulk of secondary storage of modern computers**
  - **Drives rotate at 60 to 250 times per second**
  - **Transfer rate → rate at which data flow between drive and computer**
  - **Positioning time (random-access time) is**
    - time to move disk arm to desired cylinder (**seek time**) **+**
      time for desired sector to rotate under the disk head (**rotational latency**)
  - **Head crash results from disk head making contact with disk surface**
    - That's bad

# Moving-head Disk Mechanism



# The First Commercial Disk Drive


- 1956
- IBM RAMDAC computer included the IBM Model 350 disk storage system

- 5M (6 bit) characters = 3.75 MB
- 50 x 24” platters
- 100 recording surfaces
- Each surface had 100 tracks
- Access time = < 1 second

(From Wikipedia)

# Disk Scheduling

- **The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth**

**Objectives of the OS**

- **Minimize seek time**
  - Seek time ≈ seek distance
- **Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer → maximize it**

# Disk Scheduling (Cont.)

- There are many sources of disk I/O request
  - OS
  - System processes
  - Users processes
- I/O request includes:
  - input or output mode,
  - disk address,
  - memory address,
  - number of sectors to transfer
- OS maintains queue of requests, per disk or device
- Idle disk can immediately work on I/O request, busy disk means work must queue
  - Optimization algorithms only make sense when a queue exists

# Disk Scheduling (Cont.)

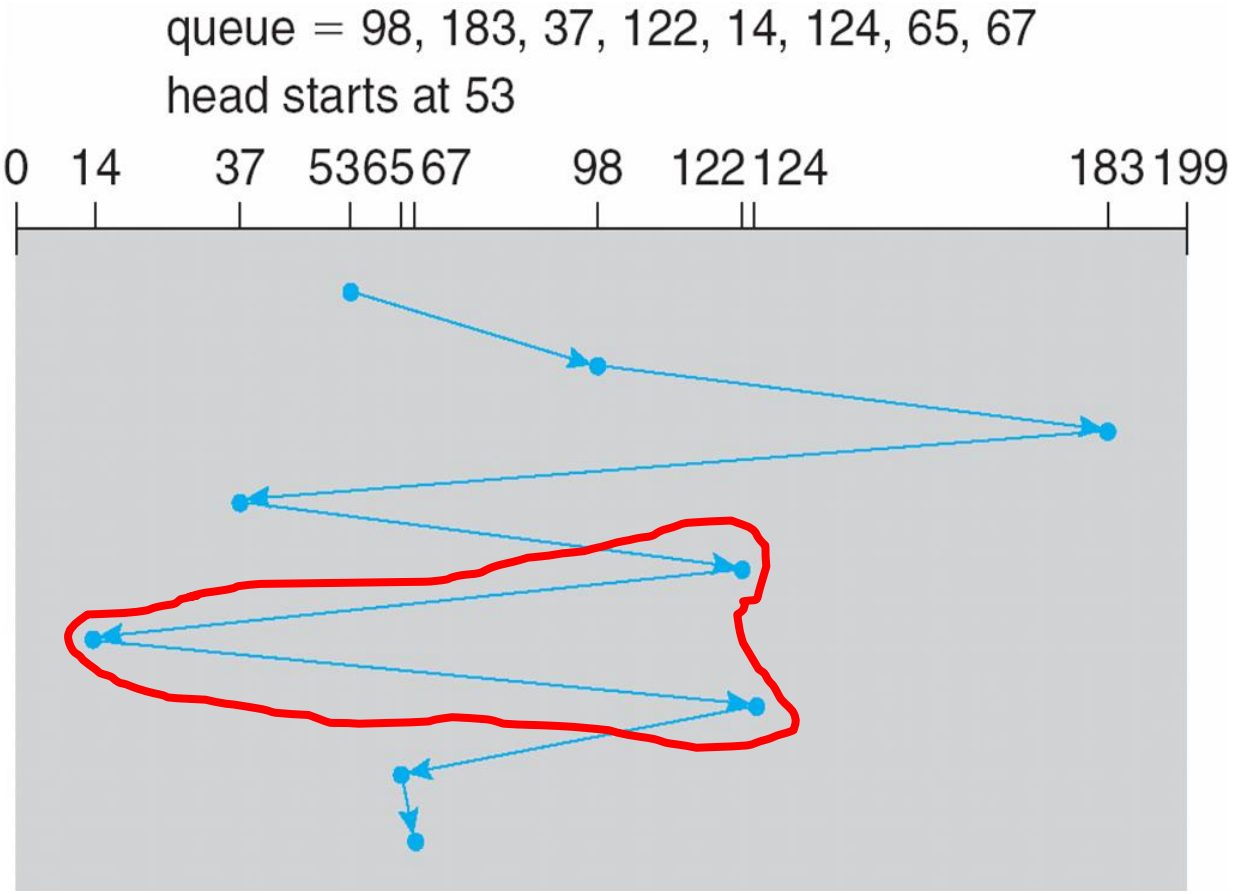
- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying “depth”)
- Several algorithms exist to schedule the servicing of disk I/O requests
- The analysis is true for one or many platters
- We illustrate scheduling algorithms with a request queue (0-199)

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

→ Head pointer 53

# FCFS

Illustration shows total head movement of 640 cylinders



# SSTF

- **Shortest Seek Time First (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**

# SSTF (Cont.)

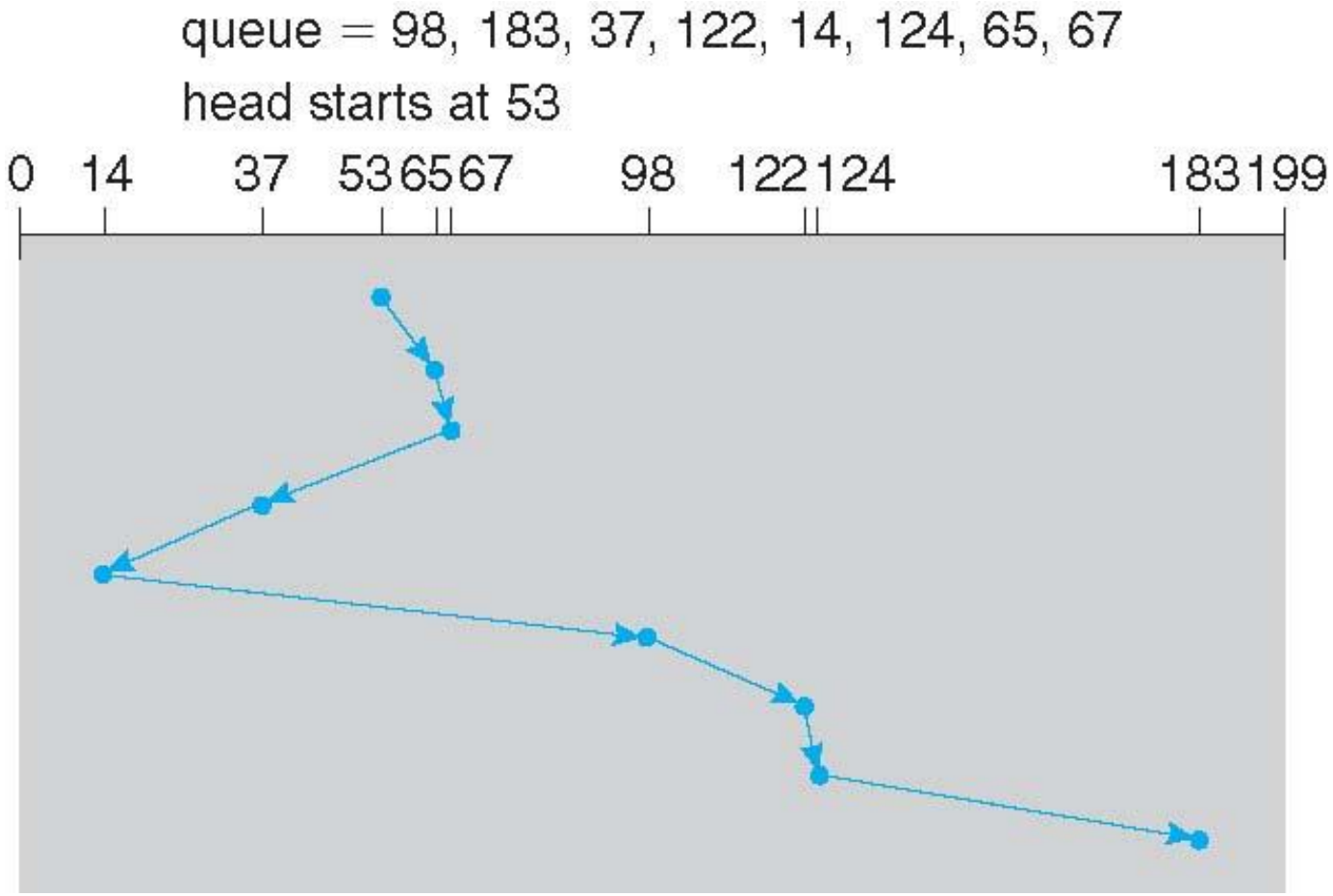



Illustration shows total head movement of 236 cylinders

SSTF not necessarily is optimal! How????? 53→ 37 → 14 → 65 → 67 →98 → 122 → 124 → 183 (208 cylinders)

# 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.

- SCAN algorithm Sometimes called the elevator algorithm

# SCAN (Cont.)

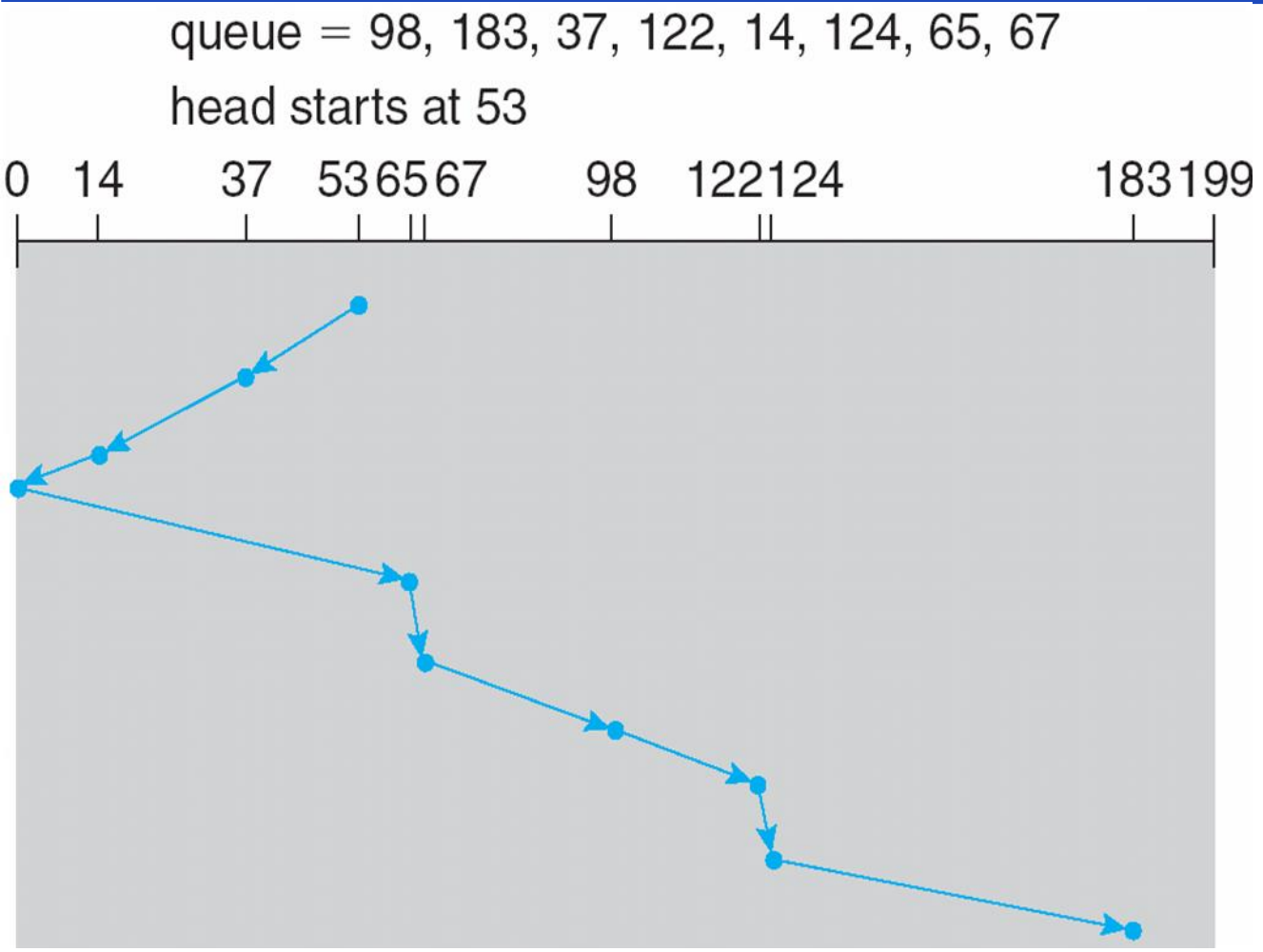



Illustration shows total head movement of 208 cylinders

- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest

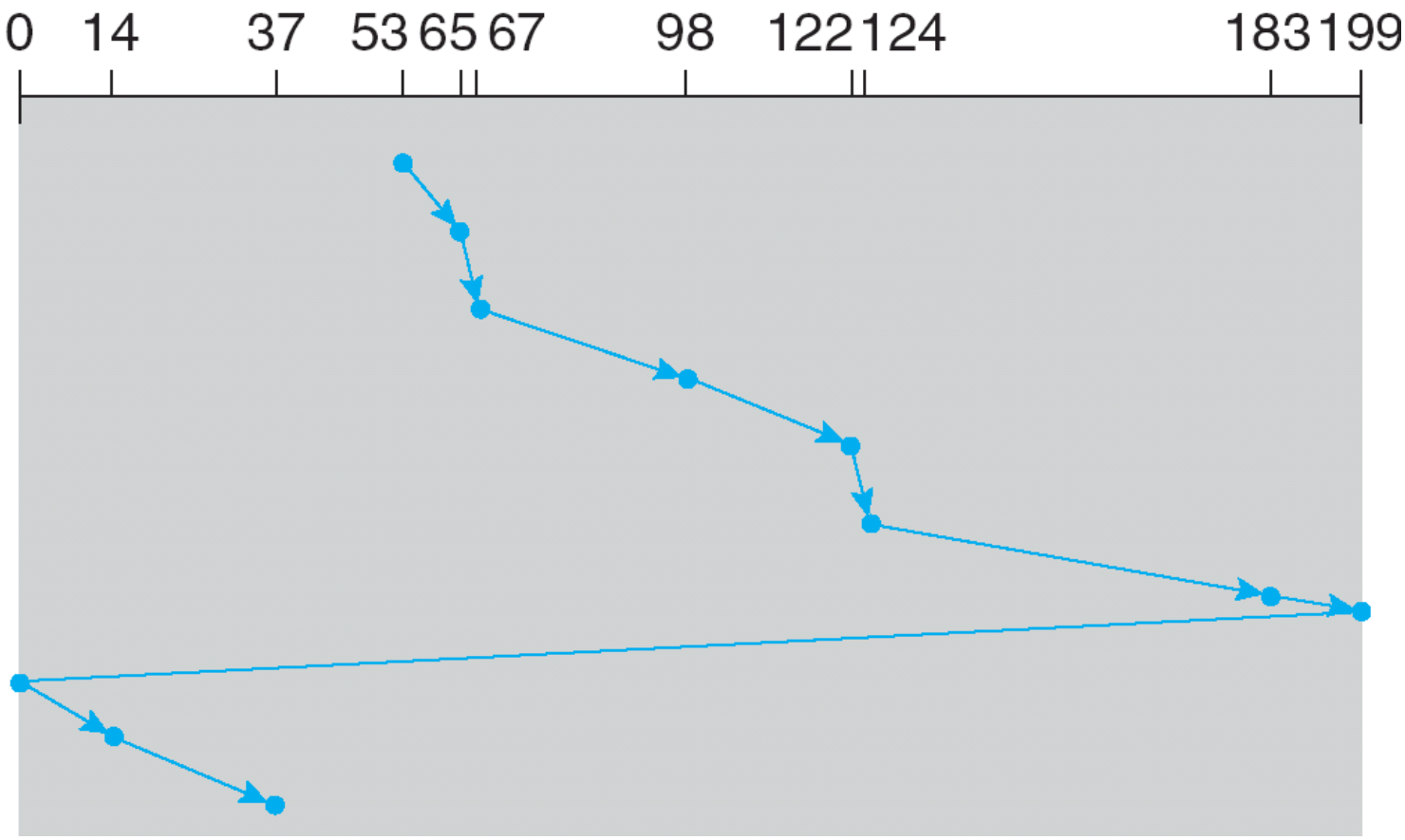
# C-SCAN

- 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

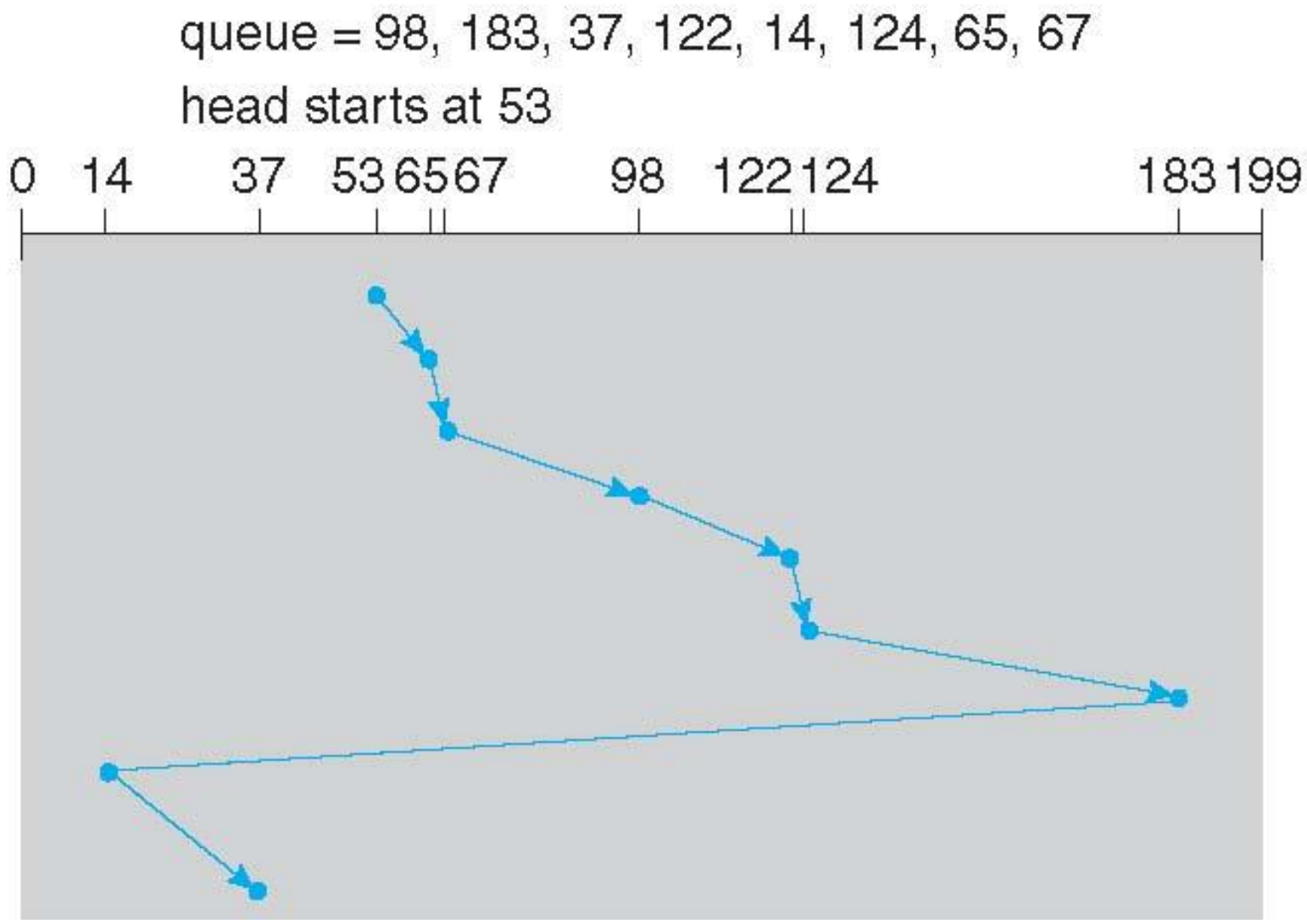
head starts at 53



# C-LOOK

- LOOK a version of SCAN, C-LOOK a 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

# Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
  - Less starvation
- Performance depends on the number and types of requests
- Requests for disk service can be influenced by the file-allocation method
  - And metadata layout

# Selecting a Disk-Scheduling Algorithm (continue)

- The disk-scheduling algorithm should be written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary
- Either SSTF or LOOK is a reasonable choice for the default algorithm
- What about rotational latency?
  - Difficult for OS to calculate
- How does disk-based queuing effect OS queue ordering efforts?

# Disk Management by OS

- **Low-level formatting, or physical formatting — Dividing a disk into sectors that the disk controller can read and write**
  - **Each sector can hold header information, plus data, plus error correction code (ECC)**
  - **Usually 512 bytes of data but can be selectable**

# Disk Management by OS (continue)

- To use a disk to hold files, the operating system still needs to record its own data structures on the disk
  - Partition the disk into one or more groups of cylinders, each treated as a logical disk
  - Logical formatting or "making a file system"
  - To increase efficiency most file systems group blocks into clusters
  - Disk I/O done in blocks
  - File I/O done in clusters
- Boot block initializes system
  - The bootstrap is stored in ROM
  - Bootstrap loader program stored in boot blocks of boot partition

# Swap-Space Management

- Swap-space — Virtual memory uses disk space as an extension of main memory
  - Less common now due to memory capacity increases
- Swap-space can be carved out of the normal file system, or, more commonly, it can be in a separate disk partition (raw)
- Swap-space management
  - Kernel uses swap maps to track swap-space use
- What if a system runs out of swap space?
- Some systems allow multiple swap spaces

# References

Part of the contents of this lecture has been adapted from the book Abraham Silberschatz, Peter B. Galvin, Greg Gagne: "Operating System Concept ", Publisher : Wiley; 9 edition (December 17, 2012), ISBN-13: 978-1118063330

Slides also contain lecture materials from John Kubiatowicz (Berkeley), John Ousterhout (Stanford), Nalini (UCI), Rainer (UCI), and others

Some slides adapted from http://www-inst.eecs.berkeley.edu/~cs162/ Copyright © 2010 UCB

# Thank you for your attention