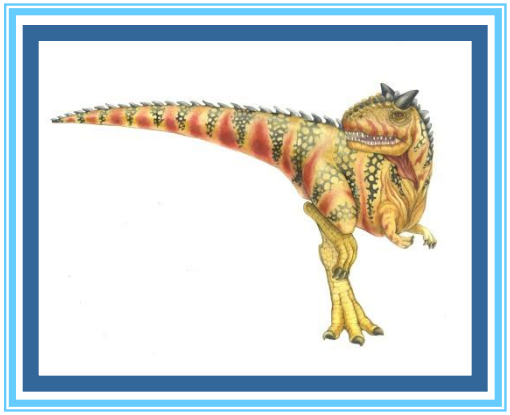
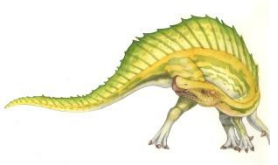
Chapter 10: Mass-Storage Systems



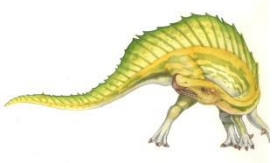
Chapter 10: Mass-Storage Systems

- Overview of Mass Storage Structure
- Disk Structure
- Disk Attachment
- Disk Scheduling

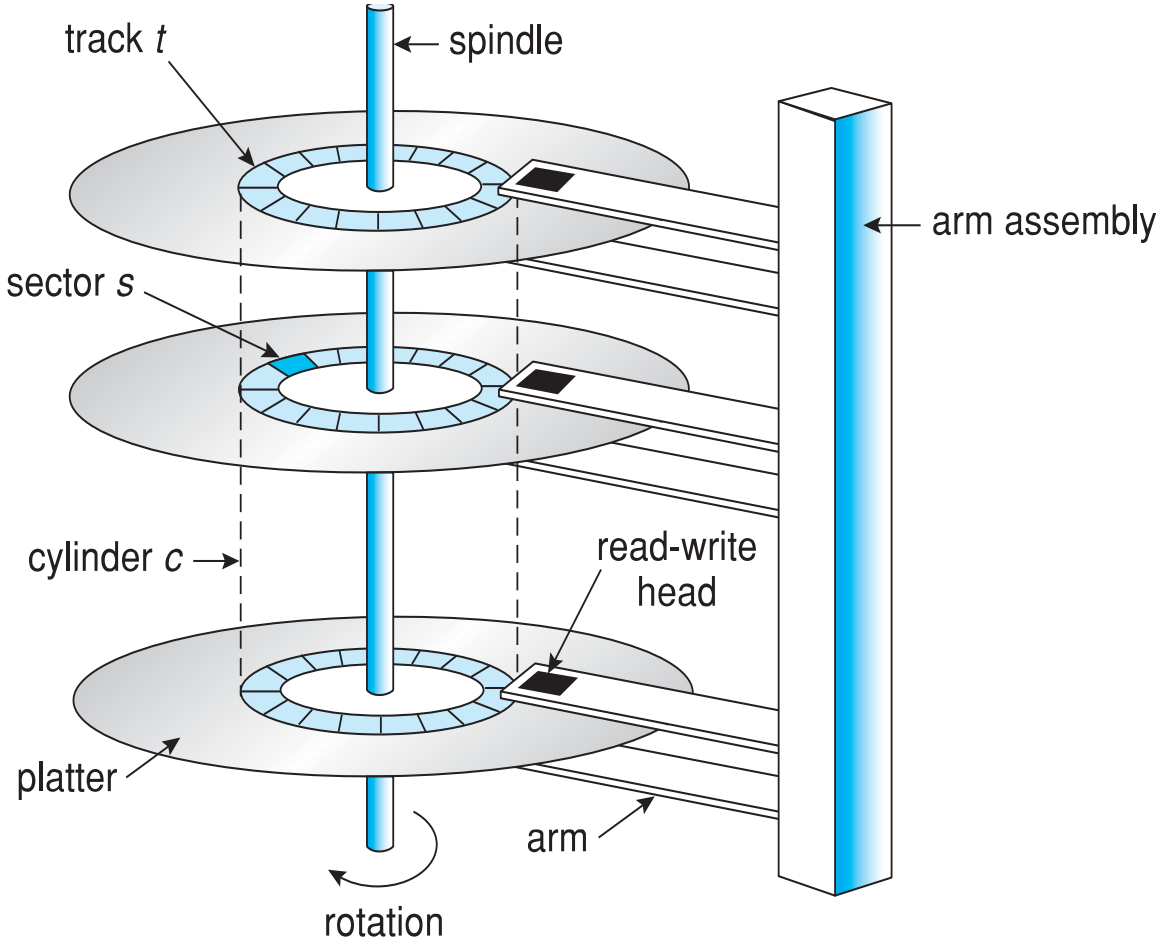


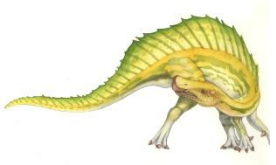
Objectives

- To describe the physical structure of secondary storage devices and its effects on the uses of the devices
- To explain the performance characteristics of mass-storage devices
- To evaluate disk scheduling algorithms



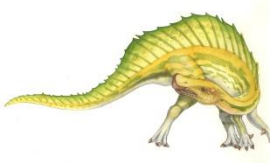
Moving-head Disk Mechanism





Hard Disk Performance

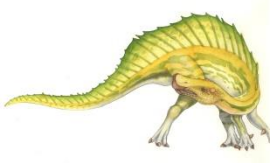
- **Access Latency** = **Average access time** = average seek time + average latency
  - For fastest disk 3ms + 2ms = 5ms
  - For slow disk 9ms + 5.56ms = 14.56ms
- Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead
- For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead =
  - 5ms + 4.17ms + 0.1ms + transfer time =
  - Transfer time = 4KB / 1Gb/s * 8Gb / GB * 1GB / 1024^2KB = 32 / (1024^2) = 0.031 ms
  - Average I/O time for 4KB block = 9.27ms + .031ms = 9.301ms



Disk Scheduling

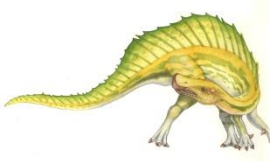
- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and large disk bandwidth
- Access time has two components:

  Seek time and Rotational latency
- Seek time: Time for the disk harm to move the heads to the cylinder containing the desired sector.
- Rotational latency : additional time for the disk to rotate the desired sector to the disk head.
- Minimize seek time
- 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



Disk Scheduling (Cont.)

- There are many sources of disk I/O request
  - System processes
  - OS
  - 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

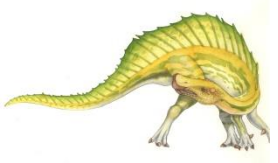


Disk Scheduling (Cont.)

- 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
- Suppose a disk is having 200 cylinders numbered from 0 to 199. The disk is currently servicing at cylinder 53 and previous request was at cylinder 60. The Queue of pending request in FIFO order is -

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

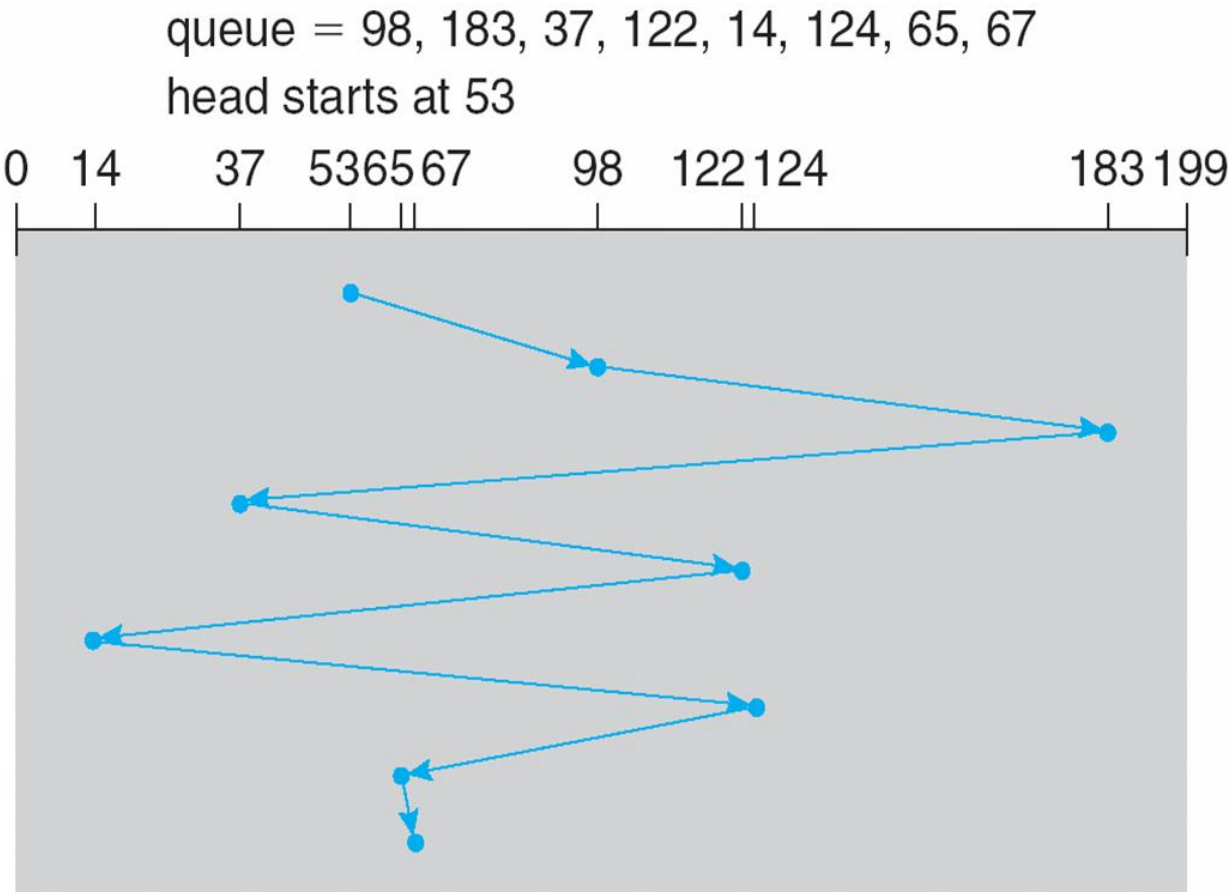
  Calculate the total distance the read/write head will travel.

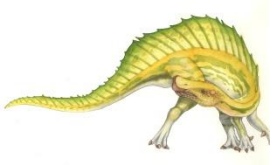


FCFS

Illustration shows total head movement of 640 cylinders :

640=(98-53)+(183-98)+(183-37)+(122-37)+(122-14)+(124-14)+(124-65)+(67-65)

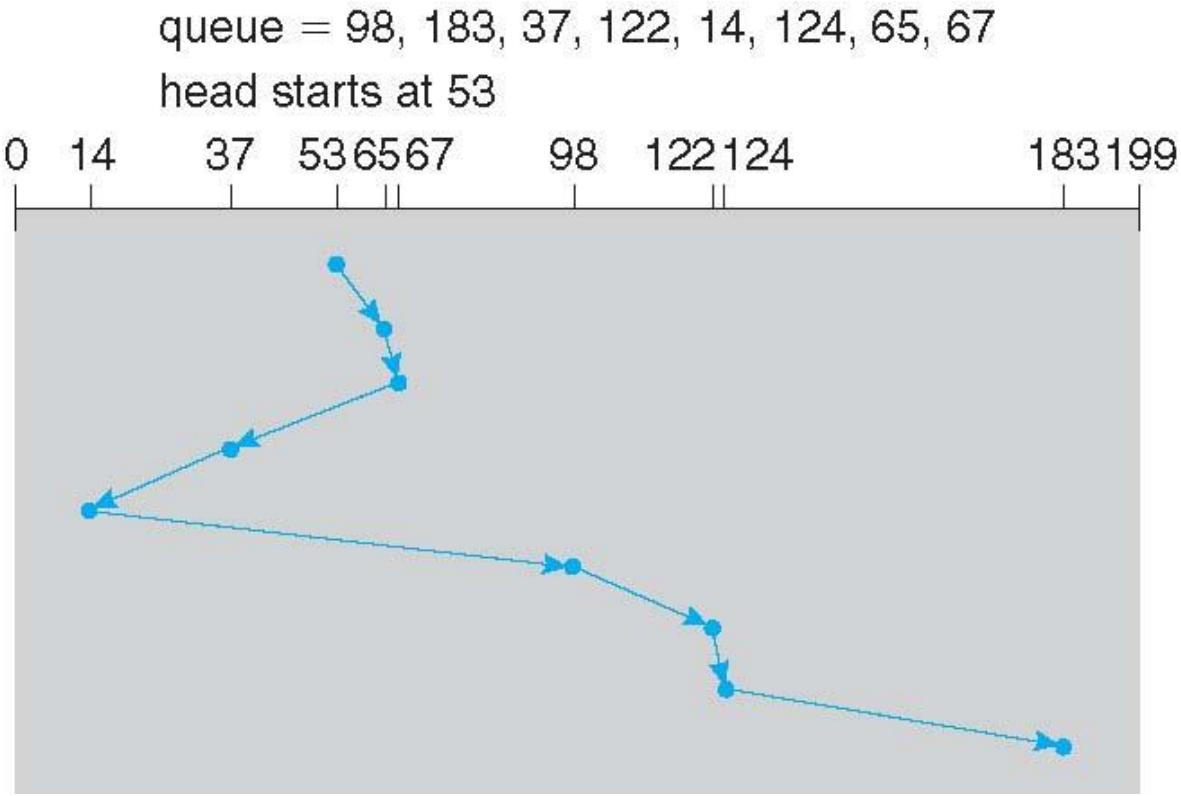


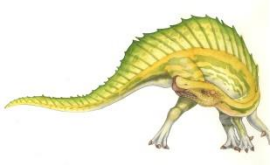


SSTF

- 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; may cause starvation of some requests
- Illustration shows total head movement of 236 cylinders

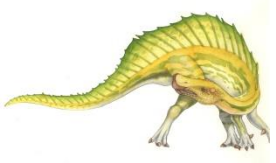
236=(65-53)+(67-65)+(67-37)+(37-14)+(98-14)+(122-98)+(124-122)+(183-124)





SCAN

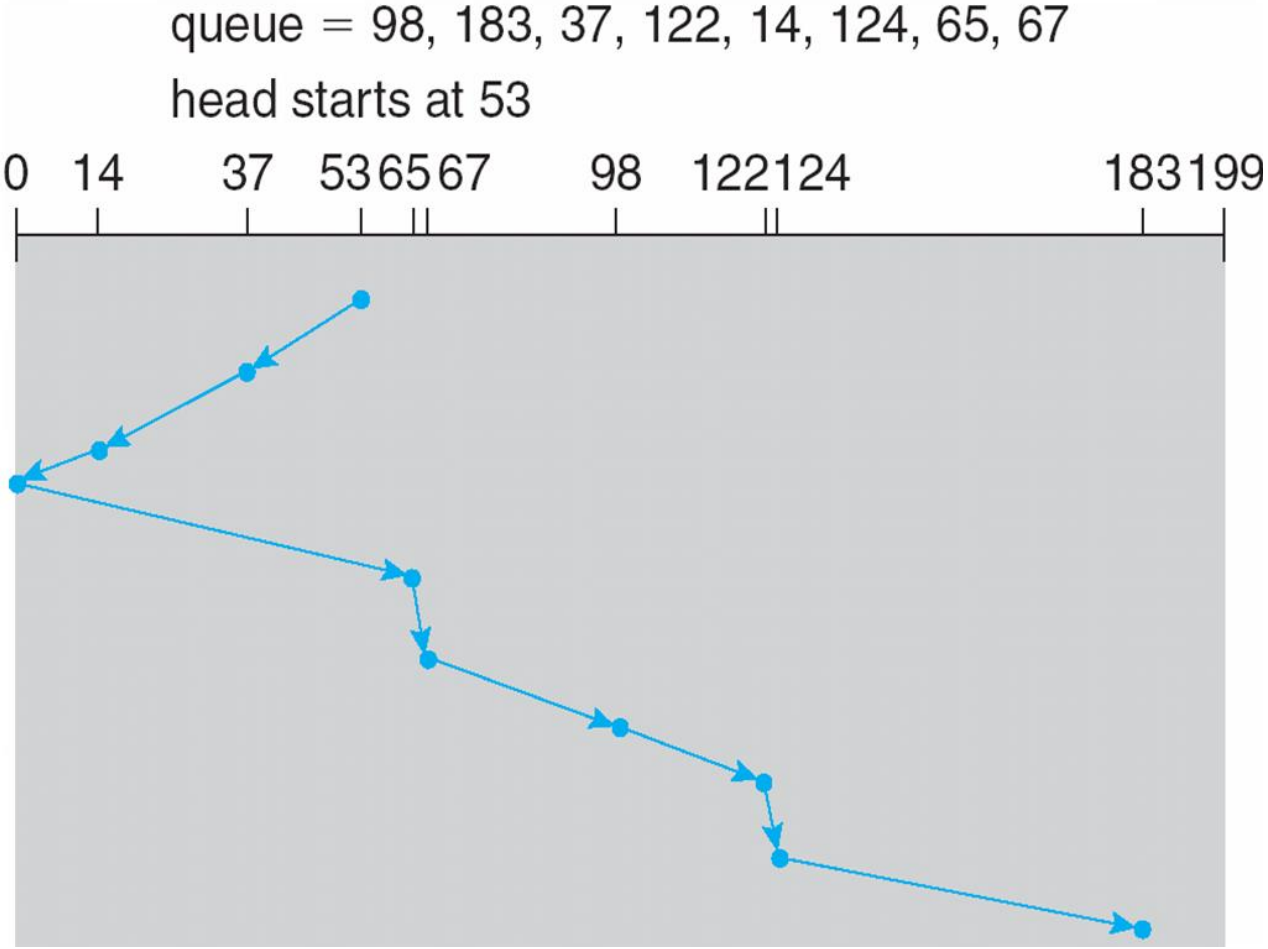
- The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.
- **SCAN algorithm** Sometimes called the **elevator algorithm**
- Illustration shows total head movement of 236 cylinders
- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest

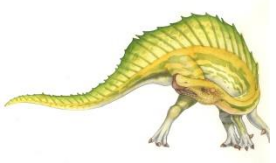


SCAN (Cont.)

236= (53-37)+(37-14)+(14-0)+(65-0)+(67-65)+(98-67)+(122-98)
+(124-122)+(183-124)

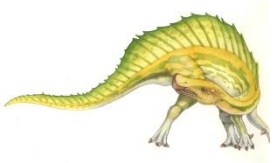
(OR) Total Head Movements=(53-0) + (183-0) = 236





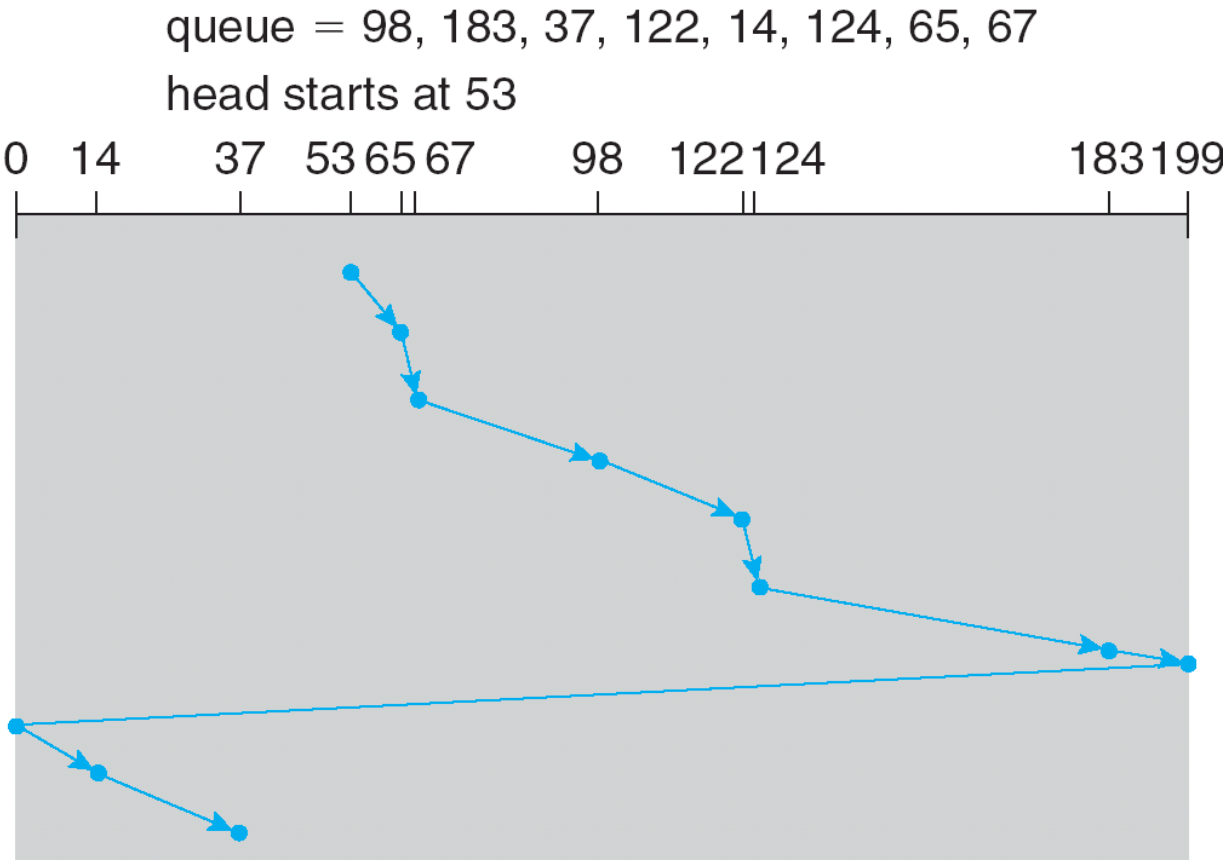
C-SCAN

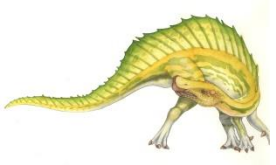
- Provides a more uniform wait time than SCAN
- The head moves from one end of the disk to the other servicing requests as it goes
  - When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one
- Total number of cylinders?



C-SCAN (Cont.)

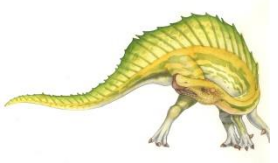
Total Head Movements=(199-53) + (199-0)+(37-0) = 382





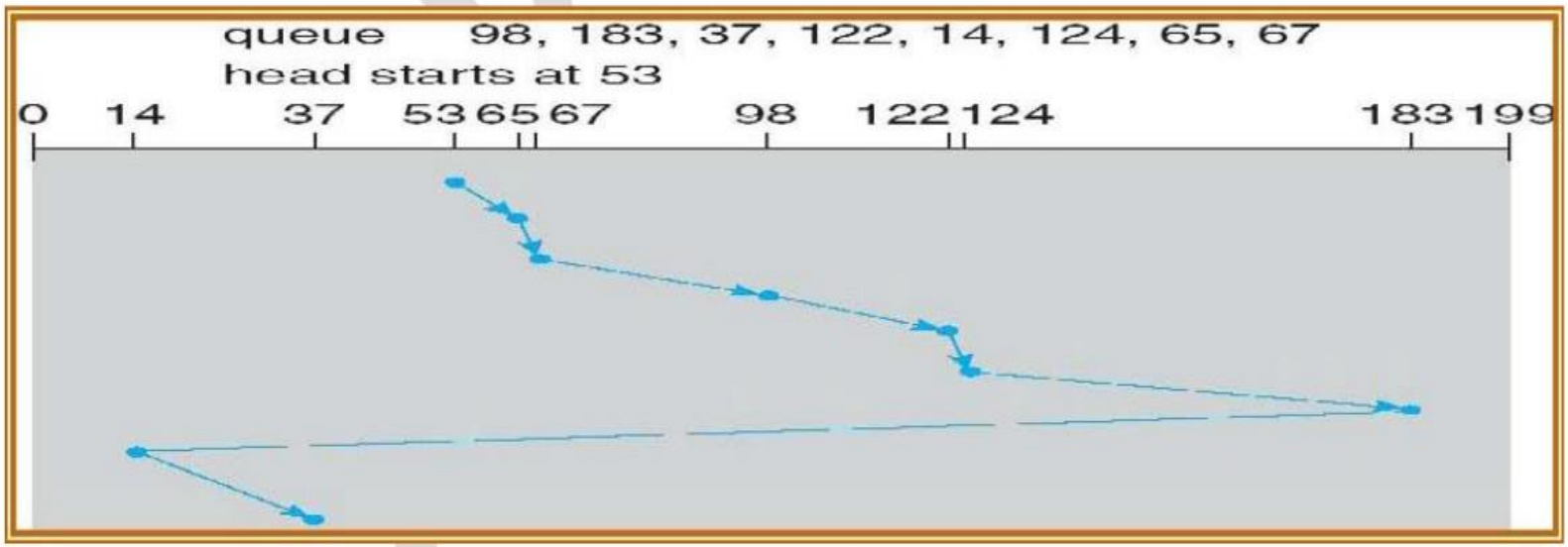
C-LOOK

- 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
- Total number of cylinders?



C-Look

vi) C-Look Scheduling algorithm:

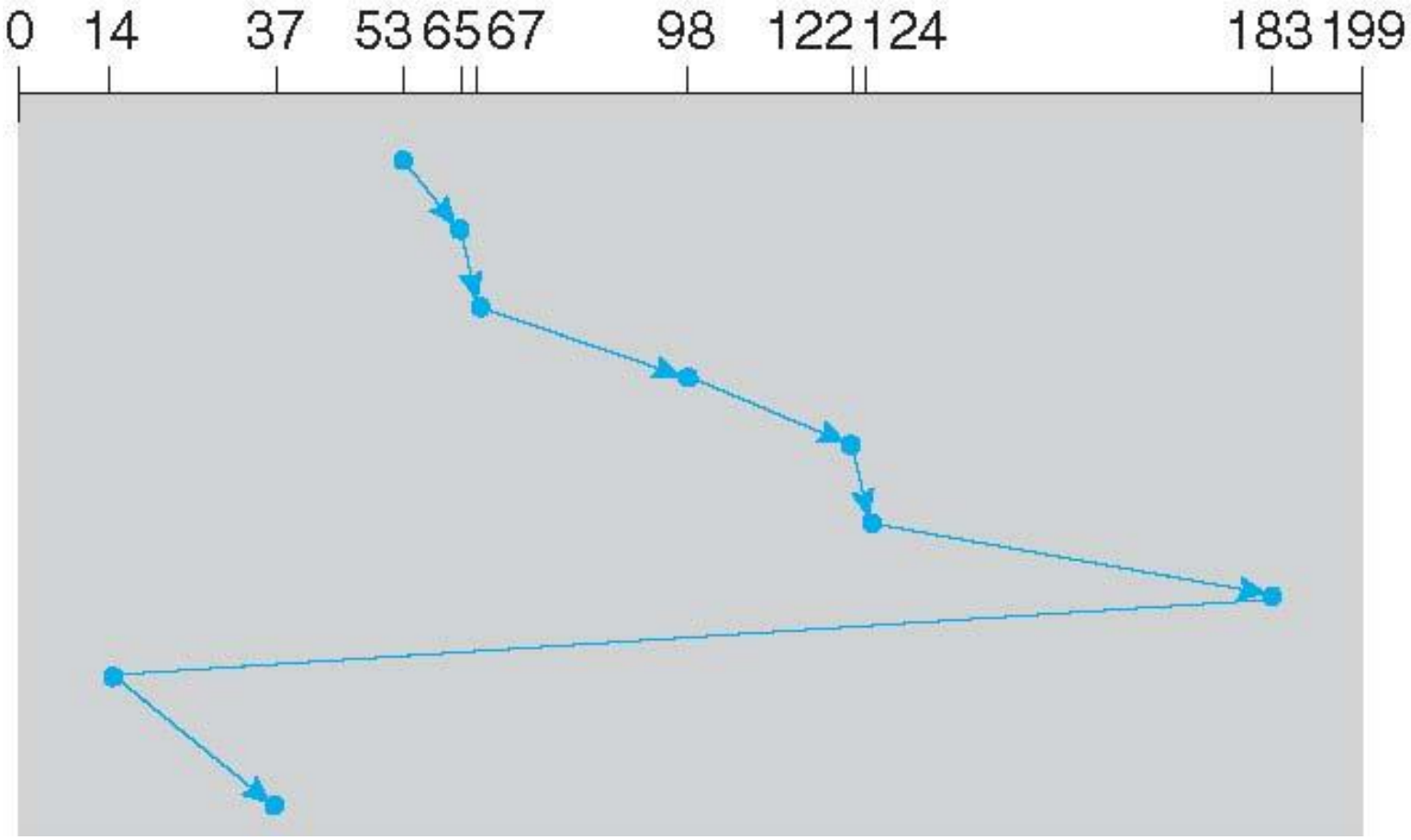


If the disk head is initially at 53 and if the head is moving towards the outer track, it services 65, 67, 98, 122, 124 and 183. At the last request, the arm will reverse and will move immediately towards the first request 14 and then serves 37.

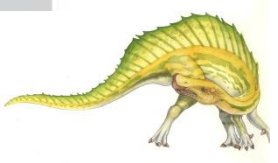
C-LOOK (Cont.)

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

head starts at 53



Total Head Movements=(183-53) + (183-14)+(37-14) = 322



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
- 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