Module 6

Storage Management and Security

Secondary-Storage Structure

The bulk of secondary storage for modern computers is provided by **hard disk drives** (**HDDs**) and **nonvolatile memory** (**NVM**) devices.

we describe the basic mechanisms of these devices and explain how operating systems translate their physical properties to logical storage via address mapping.

Hard Disk Drives

The following diagram shows the internal structure of a hard disk.

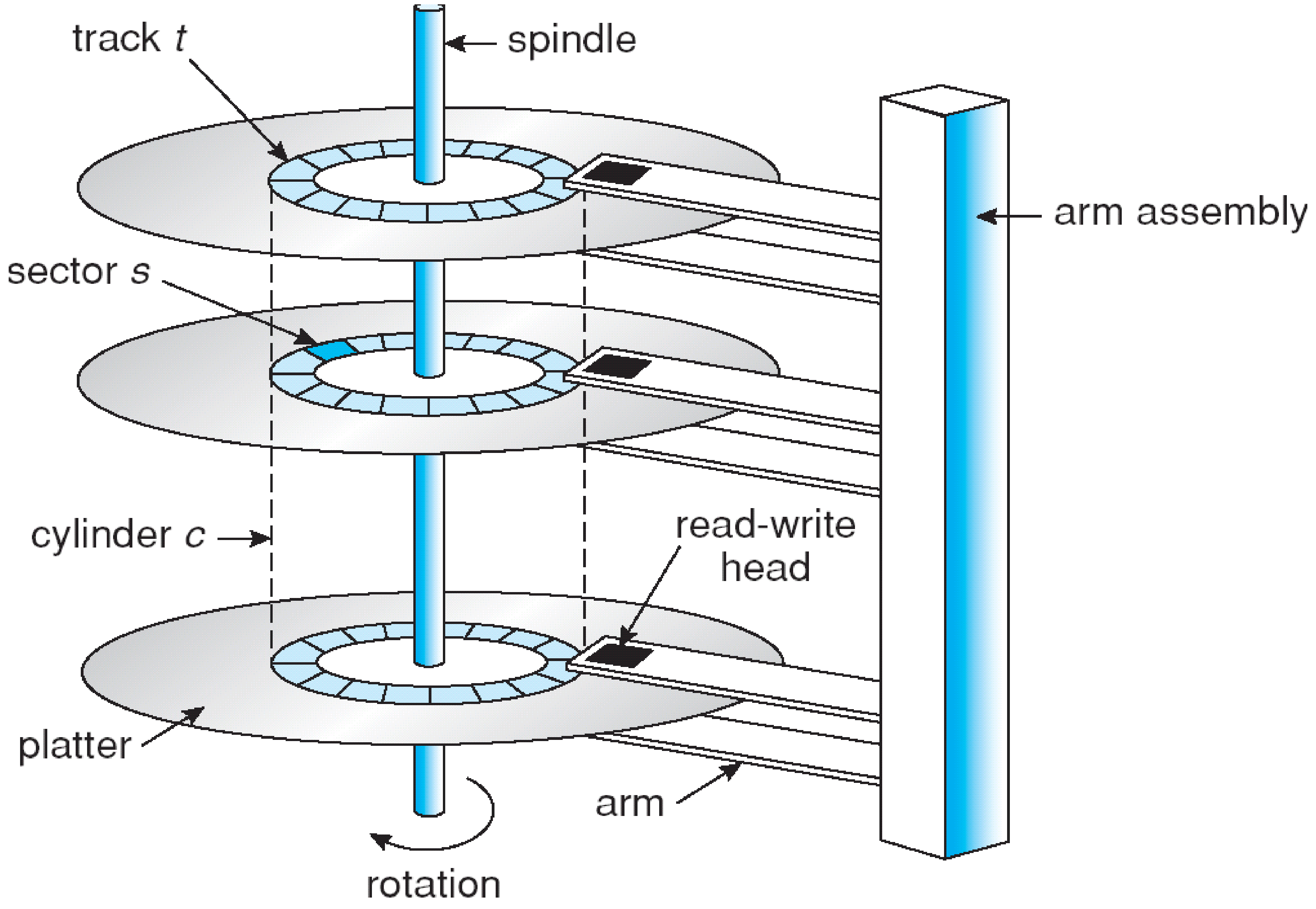
A hard disk consists of a number of platters.

Each disk platter has a flat circular shape, like a CD.

Diameter of platter ranges from 1.8 to 3.5 inches.

Information is stored on the two surfaces of platter.

A read-write head "flies" just above each surface of every platter.



track t
spindle
arm assembly
sector s
cylinder c
read-write
head
platter
arm
rotation

The heads are attached to a *disk arm* that moves all the heads as a unit.

The surface of a platter is divided into circular *tracks.*

Each track is subdivided into hundreds of sectors.

The set of tracks that are at one arm position makes up a *cylinder.*

There may be thousands of cylinders in a disk.

Each sector has a fixed size and is the smallest unit of transfer.

The sector size was commonly 512 bytes until around 2010.

The storage capacity of common disk drives is measured in gigabytes and terabytes.

A disk drive with the cover removed is shown in the following figure.

When the disk is in use, a drive motor spins it at high speed (60 to 250 times per second).

The arm assembly moves in horizontal direction (back and forth).

The time required to access data from the disk is called *access time*.

The *access time* consists of the time necessary to move the disk arm to the desired cylinder, called the *seek time* and the time necessary for the desired sector to rotate to the disk head, called the *rotational latency.*

Disks can transfer several megabytes of data per second, and they have *seek times* and *rotational latencies* of several milliseconds.

Mapping logical blocks to physical sectors

In the logical view of disk, the disk is collection of blocks. Block is the smallest unit of transfer.

In the physical view of disk, the disk is collection of sectors. Sector is the smallest unit of transfer.

The size of a sector or block is usually 512 bytes.

The logical blocks are mapped to the physical sectors.

Sector 0 is the first *sector* of the first *track* on the outermost *cylinder*.

The mapping proceeds in order through that *track*, then through the rest of the *tracks* in that *cylinder*, and then through the rest of the *cylinders* from outermost to innermost.

So, the block address is indicated as

<c, t, s>

'c' is the cylinder number, 't' is the track number and 's' is the sector number.

Ex: A hard disk has 63 sectors per track, 10 platters and 1000 cylinders.

1) Find the corresponding block number of the address <400, 16, 29>

400 x 20 x 63 + 16 x 63 + 29 = 504000 + 1008 + 29 = 505037

2) Find the address of block number 1039

Sector number = 1039 % 63 = 31
Track number = (1039 / 63) % 20 = 16 % 20 = 16
Cylinder number = 16 / 20 = 0

Address is <0, 16, 31>

3) Find the address of block number 3428

Sector number = 3428 % 63 = 26
Track number = (3428 / 63) % 20 = 54 % 20 = 14
Cylinder number = 54 / 20 = 2

Address is <2, 14, 26>

Formulae

1) Capacity of disk pack = Number of surfaces x Number of tracks per surface x Number of sectors per track x Storage capacity of one sector

2) Average disk access time = Average Seek time + Average Rotational delay or latency + Transfer time

3) Average seek time = $(k - 1) \times t / 2$

'k' is the number of tracks per surface
't' is the time taken by head to move from one track to adjacent track

4) Average rotational latency = 1 / 2 x Time taken for full rotation

5) Transfer time = Number of bytes to be transferred / Data transfer rate

6) Data transfer rate = Number of heads x Capacity of one track x Number of rotations in one second

7) Formatted disk space = Total disk capacity - Formatting overhead

8) Formatting overhead = Number of sectors x Overhead per sector

Ex1: Consider a disk pack with the following specifications - 16 surfaces, 128 tracks per surface, 256 sectors per track and 512 bytes per sector. The disk is rotating at 3600 RPM. Average seek time is 11.5 msec. Format overhead is 32 bytes per sector.

Answer the following questions-

1. What is the capacity of disk pack?
2. What is the data transfer rate?
3. What is the average access time?
4. What is the formatted disk space?

Capacity of disk pack = Number of surfaces x Number of tracks per surface x Number of sectors per track x Number of bytes per sector = 16 x 128 x 256 x 512 bytes = 2^{28} bytes = 256 MB

Data transfer rate = Number of heads x Capacity of one track x Number of rotations in one second = 16 x (256 x 512 bytes) x (3600 / 60) rotations/sec = 2^4 x 2^8 x 2^9 x 60 bytes/sec = 60 x 2^{21} bytes/sec = 120 MBps

Average rotational delay = 1/2 x Time taken for one full rotation = 1/2 x (1/3600) minutes = 1/2 x (60/3600) sec = 1/2 X 1/60 sec = 0.0083 sec = 0.0083 X 1000 msec = 8.3 msec

Average access time = Average seek time + Average rotational delay + Transfer time = 11.5 msec + 8.3 msec + 0 = 19.8 msec

Formatting overhead = Total number of sectors x overhead per sector = (16 x 128 x 256) x 32 bytes = 2^{19} x 2^5 bytes = 2^{24} bytes = 16 MB

Formatted disk space = Total disk space – Formatting overhead = 256 MB - 16 MB = 240 MB

Ex2: What is the average access time for transferring 512 bytes of data with the following specifications-

- Average seek time = 5 msec
- Disk rotation = 6000 RPM
- Data rate = 40 KB/sec

Average rotational delay = 1/2 x Time taken for one full rotation = 1/2 x 1/6000 minutes = 1/2 X 60/6000 sec = 1/2 X 1/100 sec = 1/200 sec = 0.005 sec = 0.005 X 1000 msec = 5 msec

Transfer time = Number of bytes to be transferred / Data transfer rate = 512 bytes / 40 KB per sec = 2^9 bytes / 40 X 2^{10} bytes per sec = 0.0125 sec = 0.0125 X 1000 = 12.5 msec

Average access time = Average seek time + Average rotational delay + Transfer time = 5 msec + 5 msec + 12.5 msec = 22.6 msec

Ex3: Consider a typical disk that rotates at 15000 RPM and has a transfer rate of 50 x 10^6 bytes/sec. If the average seek time of the disk is twice the average rotational delay. What is the average time (in milliseconds) to read or write a 512 byte sector of the disk?

Average rotational delay = 1/2 x Time taken for one full rotation = 1/2 x 1/15000 minutes = 1/2 X 60/15000 sec = 2 msec

Average seek time = 2 x Average rotational delay = 2 x 2 msec = 4 msec

Transfer time = Number of bytes to be transferred / Data transfer rate = 512 bytes / (50 x 10^6 bytes/sec) = 10.24 x 10^{-6} sec = 0.01024 msec

Average time to read or write 512 bytes = Average seek time + Average rotational delay + Transfer time = 4 msec + 2 msec + 0.01024 msec = 6.11 msec

Ex4: A hard disk system has the following parameters-
- Number of tracks = 500
- Number of sectors per track = 100
- Number of bytes per sector = 500
- Time taken by the head to move from one track to another adjacent track = 1 msec
- Rotation speed = 600 RPM

What is the average time taken for transferring 250 bytes from the disk?

Average seek time = (500 -1) x 1 msec / 2 = 249.5 msec

Average rotational delay = 1/2 x Time taken for one full rotation = 1/2 x 1/600 minutes = 1/2 x 60/600 sec = 1/200 sec = 0.005 sec = 0.005 X 1000 msec = 50 msec

Capacity of one track = Number of sectors per track x Number of bytes per sector = 100 x 500 bytes = 50000 bytes

Data transfer rate = Number of heads x Capacity of one track x Number of rotations in one second = 1 x 50000 bytes x (600 / 60) = = 50000 x 10 bytes/sec = 5 x 10^5 bytes/sec

Transfer time = (250 bytes / 5 x 10^5 bytes) sec = 50 x 10^{-5} sec = 0.5 msec

Average time taken to transfer 250 bytes = Average seek time + Average rotational delay + Transfer time = 249.5 msec + 50 msec + 0.5 msec = 300 msec

Ex5: A certain moving arm disk storage with one head has the following specifications-

- Number of tracks per surface = 200
- Disk rotation speed = 2400 RPM
- Track storage capacity = 62500 bits

Calculate Average latency and Data transfer rate.

Average latency = 1/2 x Time taken for one full rotation = 1/2 x 1/2400 minutes = 1/2 X 60 / 2400 sec = 1/2 X 1/40 sec = 1/80 sec = 0.0125 sec = 0.0125 X 1000 msec = 12.5 msec

Data transfer rate = Number of heads x Capacity of one track x Number of rotations in one second = 1 x 62500 bits x (2400 / 60) = 2500000 bits/sec = 2.5 x 10^6 bits/sec

Disk Scheduling

Operating system has to use the hardware efficiently.

To use the disk efficiently, the operating system has to minimize the access time and maximize the bandwidth.

The access time and bandwidth are improved by managing the order in which the disk I/O requests are serviced.

The disk I/O requests are placed in the disk queue.

To service these requests, several disk scheduling algorithms are used.

The different disk scheduling algorithms are:

1) First Come First Serve (FCFS) Scheduling
2) Shortest Seek Time First (SSTF) Scheduling
3) SCAN Scheduling
4) C- SCAN Scheduling
5) LOOK Scheduling
6) C-LOOK Scheduling

First Come First Serve (FCFS) Scheduling

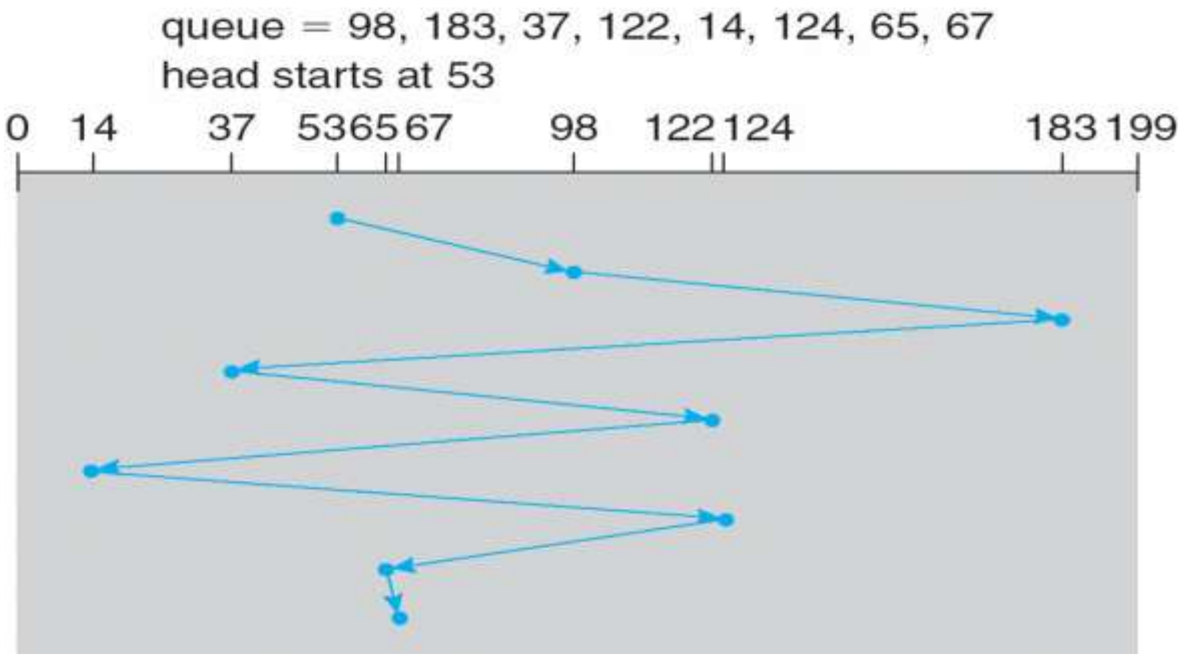
This algorithm is simple, but it does not provide the fastest service.

Consider a disk queue with requests for I/O to blocks on the cylinders 98, 183, 37, 122, 14, 124, 65, 67.

If the disk head is initially at cylinder 53, it will first move from 53 to 98, then to 183, 37, 122, 14, 124, 65, and finally to 67.

The total head movement is 640 cylinders.

This schedule is diagrammed in the following figure.



SSTF (Shortest Seek Time First) Scheduling

The SSTF algorithm selects the request with the least seek time from the current head position.

In other words, SSTF chooses the pending request closest to the current head position.

For the example request queue (98, 183, 37, 122, 14, 124, 65, 67), the closest request to the initial head position (53) is at cylinder 65.

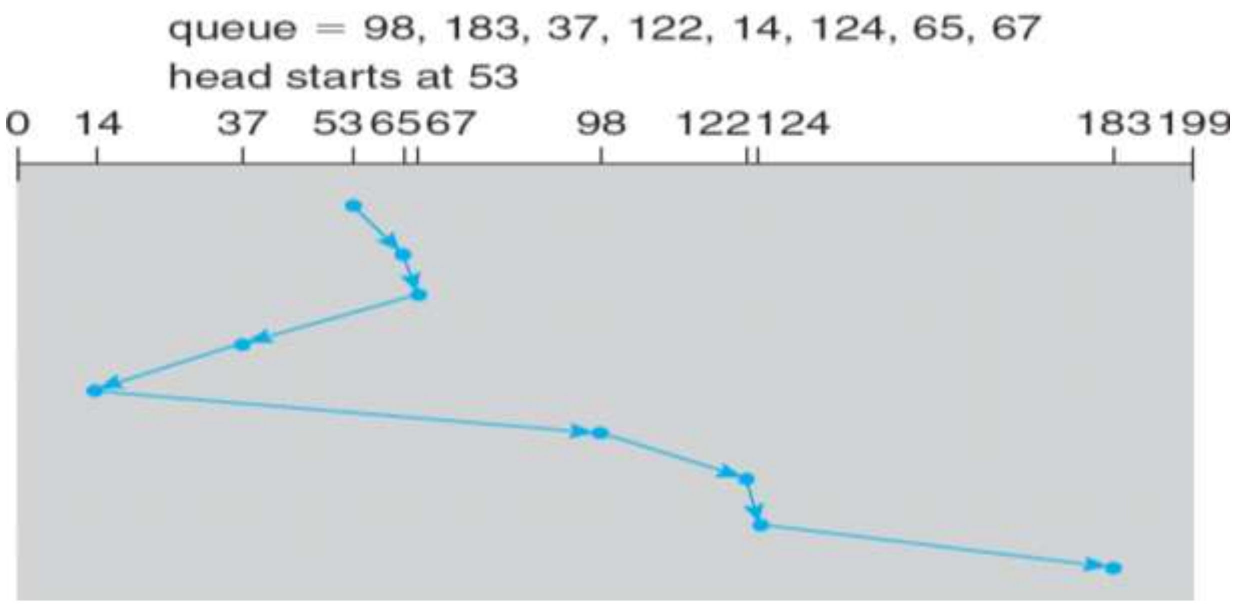
After processing the request at cylinder 65, the next closest request is at cylinder 67.

From there, the request at cylinder 37 is closer, so 37 is served next.

Continuing, we service the request at cylinder 14, then 98, 122, 124, and finally 183.

This scheduling method results in a total head movement of only 236 cylinders.

This schedule is diagrammed in the following figure.



SSTF scheduling may cause starvation of some requests.

Suppose that we have two requests in the queue, for cylinders 14 and 186, and while the request for 14 is being serviced, a new request near 14 arrives.

This new request will be serviced next, making the request at 186 wait.

While this request is being serviced, another request close to 14 could arrive.

A continual stream of requests near one another can cause the request for cylinder 186 to wait indefinitely.

SCAN Scheduling

In the SCAN algorithm, the disk arm starts at one end of the disk and moves towards the other end, servicing requests along the way, until it gets to the other end of the disk.

At the other end, the direction of head movement is reversed, and servicing continues.

The head continuously scans back and forth across the disk.

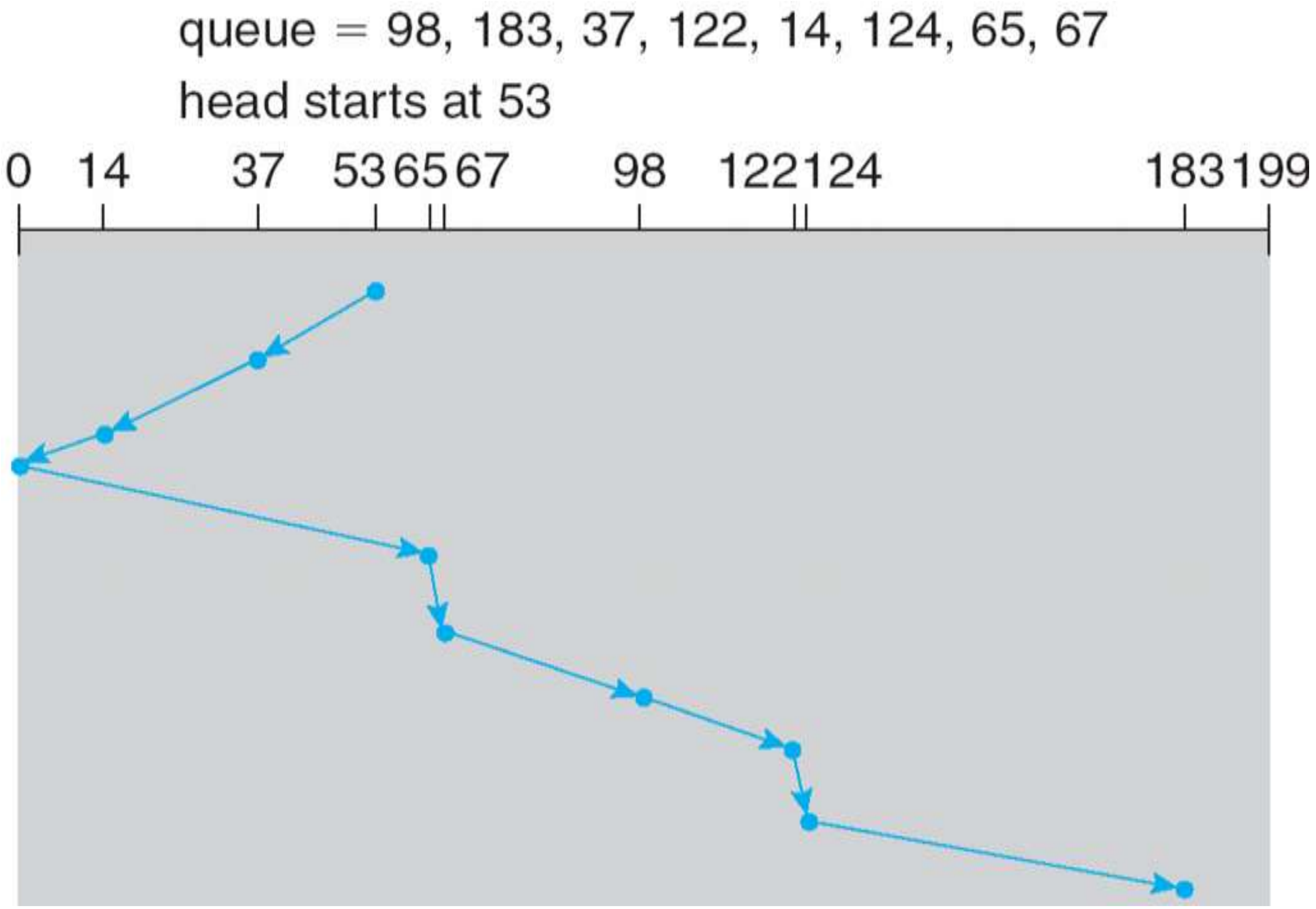
Before applying SCAN algorithm to schedule the requests, we need to know the direction of head movement in addition to the head's current position.

Assuming that the disk arm is moving towards 0 and that the initial head position is again 53, the head will next service 37 and then 14.

At cylinder 0, the arm will reverse and will move towards the other end of the disk, servicing the requests at 65, 67, 98, 122, 124, and 183.

The total head movement is 236 cylinders.

This schedule is diagrammed in the following figure.



When the head reaches one end and reverses direction, very few requests are in front of the head, since these cylinders have recently been serviced.

More requests will be at the other end of the disk and these requests have also waited the longest.

C-SCAN Scheduling

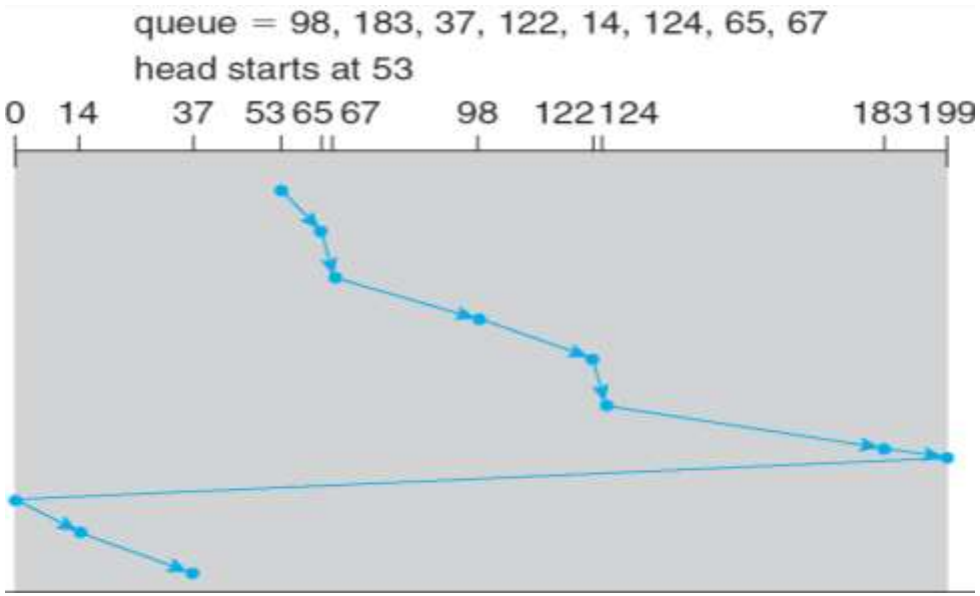
Circular SCAN (C-SCAN) is a variant of SCAN and provides a more uniform wait time.

Like SCAN, C-SCAN moves the head from one end of the disk to the other, servicing requests along the way.

When the head reaches the other end, it immediately returns to the beginning of the disk without servicing any requests on the return trip.

This schedule is diagrammed in the following figure.

The total head movement is 383 cylinders.



LOOK Scheduling

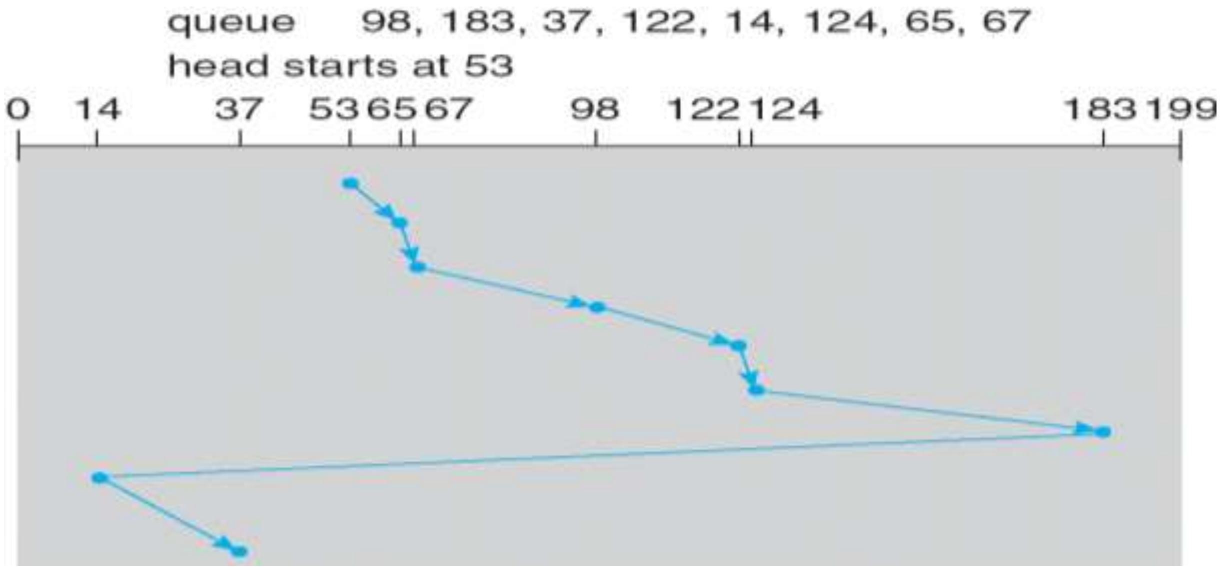
Both SCAN and C-SCAN move the disk arm across the full width of the disk.

In practice, the arm goes only as far as the final request in each direction.

Then, it reverses direction immediately, without going all the way to the end of the disk.

Versions of SCAN and C-SCAN that follow this pattern are called LOOK and C-LOOK scheduling, because they *look* for a request before continuing to move in a given direction.

The working of C-LOOK algorithm is scheduled in the following figure.



Ex: Consider a disk with 200 cylinders (0 to 199). Consider a disk queue with requests for I/O to blocks on the cylinders 82, 170, 43, 140, 24, 16, 190. The disk head is initially at cylinder 50. Head moves towards smallest cylinder number. Calculate total head movement under

1) FCFS
2) SSTF
3) SCAN
4) C-SCAN
5) LOOK
6) C-LOOK