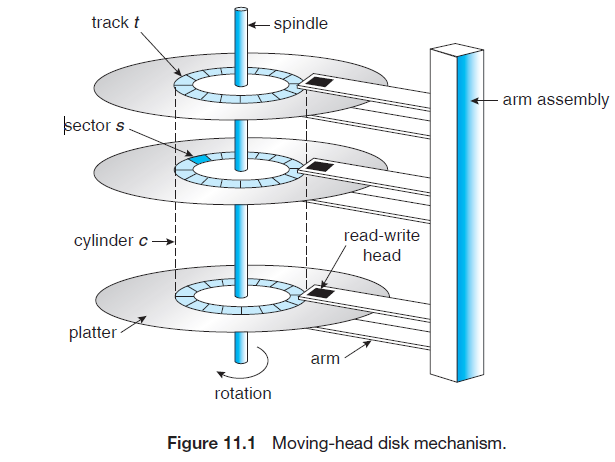
**Overview of Mass-Storage Structure:**

**Magnetic Disks:**

Magnetic disks provide bulk of secondary storage of modern computer systems. Disks are relatively simple shown in Figure 11.1. Each disk platter has a flat circular shape, like a CD. Common platter diameters range from1.8 to5.25 inches.The two surfaces of a platter are covered with a magnetic material. We store information by recording it magnetically on the platters.



A read–write head “flies” just above each surface of every platter. The heads are attached to a disk arm that moves all the heads as a unit. The surface of a platter is logically divided into circular tracks, which are subdivided into sectors. The set of tracks that are at one arm position makes up a cylinder. There may be thousands of concentric cylinders in a disk drive, and each track may contain hundreds of sectors. The storage capacity of common disk drives is measured in gigabytes.

When the disk is in use, a drive motor spins it at high speed. Most drives rotate 60 to 200 times per second. Disk speed has two parts. The transfer rate is the rate at which data flow between the drive and the computer. The positioning time, sometimes called the random-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.

The disk platters are coated with a thin protective layer, the head will sometimes damage the magnetic surface. This accident is called a head crash. A head crash normally cannot be repaired; the entire disk must be replaced.

A disk can be removable, allowing different disks to be mounted as needed. A disk drive is attached to a computer by a set of wires called an I/O bus. Several kinds of buses are available, including enhanced integrated drive electronics (EIDE), advanced technology attachment (ATA), serial ATA (SATA),

Universal serial bus (USB), fiber channel (FC), and small computer-systems interface (SCSI) buses. The data transfers on a bus are carried out by special electronic processors called controllers. The host controller is the controller at the computer end of the bus. A disk controller is built into each disk drive. To perform a disk I/O operation, the computer places a command into the host controller.

**Magnetic Tapes:**

Magnetic tape was used as an early mass-storage medium. Although it is relatively permanent and can hold large quantities of data, its access time is slow compared with that of main memory and magnetic disk. In addition, random access to magnetic tape is about a thousand times slower than random access to magnetic disk, so tapes are not very useful for mass storage. Tapes are used mainly for backup, for storage of infrequently used information, and as a medium for transferring information from one system to another.

A tape is kept in a spool and is wound or rewound past a read–write head. Moving to the correct spot on a tape can take minutes, but once positioned, tape drives can write data at speeds comparable to disk drives. Tape capacities vary greatly, depending on the particular kind of tape drive. Typically, they store from 20 GB to 200 GB. Some have built-in compression that can more than double the effective storage. Tapes and their drivers are usually categorized by width, including 4, 8, and 19 millimeters and 1/4 and 1/2 inch. Some are named according to technology, such as LTO-2 and SDLT.

**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 size of a logical block is usually 512 bytes, although some disks can be low-level formatted to have a different logical block size, such as 1,024 bytes. 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.

The number of sectors per track has been increasing as disk technology improves, and the outer zone of a disk usually has several hundred sectors per track. Similarly, the number of cylinders per disk has been increasing; large disks have tens of thousands of cylinders.

**Disk Scheduling:**

The responsibilities of the operating system is to use the hardware efficiently. For the disk drives, the responsibility is having fast access time and large disk bandwidth. The access time has two major components. The seek time is the time for the disk arm to move the heads to the cylinder containing the desired sector. The rotational latency is the additional time for the disk to rotate the desired sector to the disk head. The 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. We can improve both the access time and the bandwidth by managing the order in which disk I/O requests are serviced. Several algorithms exist to schedule the servicing of disk I/O requests.

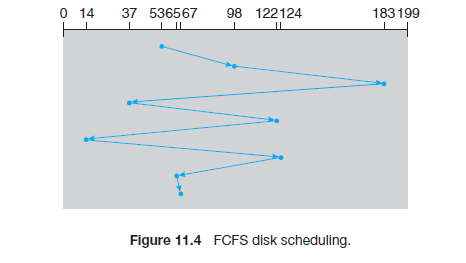
FCFS Scheduling:

The simplest form of disk scheduling is, of course, the first-come, first-served (FCFS) algorithm. This algorithm is intrinsically fair, but it generally does not provide the fastest service. Consider, for example, a disk queue with requests for I/O to blocks on cylinders

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

queue \_ 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



in that order. 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, for a total head movement of 640 cylinders. This schedule is diagrammed in Figure 11.4.

Advantages:

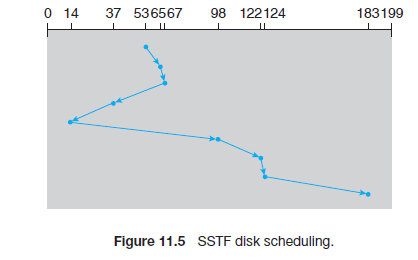
* Every request gets a fair chance
* No indefinite postponement

Disadvantages:

* Does not try to optimize seek time
* May not provide the best possible service

**SSTF (shortest-seek-time-first )Scheduling:**

The SSTF algorithm selects the request with the least seek time from the current head position. Since seek time increases with the number of cylinders traversed by the head, SSTF chooses the pending request closest to the current head position.



For our example request queue, the closest request to the initial head position (53) is at cylinder 65. Once we are at cylinder 65, the next closest request is at cylinder 67. From there, the request at cylinder 37 is closer than the one at 98, so 37 is served next. Continuing, we service the request at cylinder 14, then 98, 122, 124, and finally 183 shown in Figure 11.5. This scheduling method results in a total head movement of only 236 cylinder.

Advantage:

This algorithm gives a substantial improvement in performance.

Disadvantage:

It may cause starvation of some requests. Suppose that we have two requests in the queue, for cylinders 14 and 186, and while the request from 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. In theory, a continual stream of requests near one another could cause the request for cylinder 186 to wait indefinitely. This scenario becomes increasingly likely as the pending-request queue grows longer.

**SCAN Scheduling:**

The disk arm starts at one end of the disk, and moves toward the other end, servicing requests as it reaches each cylinder until it gets to the other end of the disk. At the other end, the direction of head movement is reversed, and servicing continues. SCAN algorithm sometimes called the elevator algorithm.

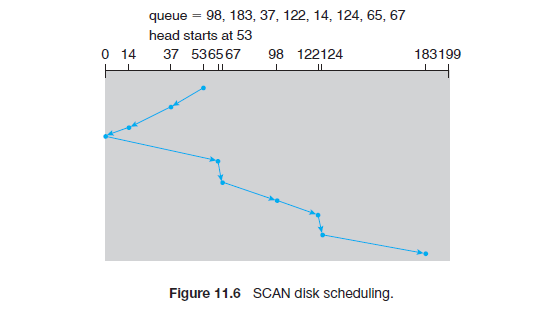
****

Illustration shows total head movement of 208 cylinders

Advantages:

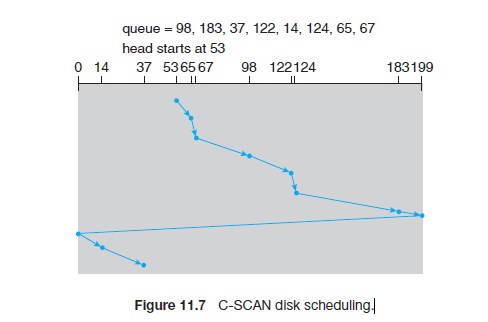
* High throughput
* Low variance of response time
* Average response time

Disadvantages:

* Long waiting time for requests for locations just visited by disk arm

**C-SCAN (Circular SCAN) Scheduling:**

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, however, it immediately returns to the beginning of the disk without servicing any requests on the return trip shown in Figure 11.7.

****

**Advantage:**

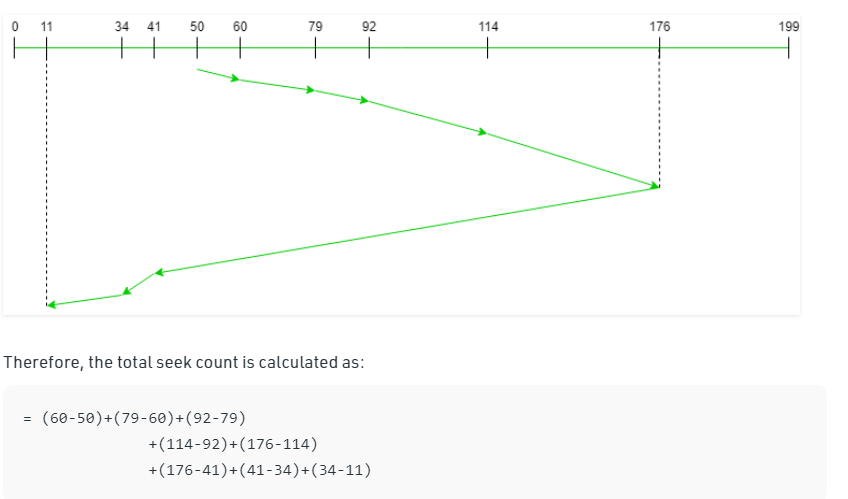
It provides a more uniform wait time.

**LOOK:** It is similar to the SCAN disk scheduling algorithm except for the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced in front of the head and then reverses its direction from there only. Thus it prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

Ex:

Request sequence = {176, 79, 34, 60, 92, 11, 41, 114}

Initial head position = 50



C-LOOK: 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.

Suppose the requests to be addressed are-82,170,43,140,24,16,190. And the Read/Write arm is at 50, and it is also given that the disk arm should move **“towards the larger value”** So, the seek time is calculated as:

= (190-50)+(190-16)+(43-16)  
= 341

