

## UNIT 4

### Device Management

- (A) Device Function.
- (B) Device Characteristic.
- (C) Disk space Management.
- (D) Allocation and Disk scheduling Methods.

#### **[4.1] Device Management Functions**

- The management of I/O devices such as printers, card readers, taps, disk and supporting devices such as control unit and control channel.
- The basic function of I/O devices are:
  1. Track status of all devices which are requires special mechanism.
  2. Deciding the policy on determine which process gets a device, for how long and where.
    - There are three basic techniques for implementing a device for policy.
      1. **Dedicated** : A technique where a device is assign to a single process.
      2. **Shared** : A technique where a device shared by many processes.
      3. **Virtual** : A technique where one physical device this simulated on another physical device. Its Combination of dedicated devices that have been transformed into shared devices.
  3. Allocation : Physically ascending a device to process with controlling unit and channel.
  4. De-allocation : De-allocate the devices at 2 levels.
    1. On job level : A device assign only for as long as the job exits in the system.
    2. On process level : A device assign only for as long as the process needs it.

#### **[4.2] Device characteristics**

Devices can be categorizes into two groups.

1. Input/Output Device.
2. Storage devices.

##### **1. Input/Output Devices :**

- An input devices takes data from the outside world.
- An output device which produce data to the outside world.
- Example : Card reader (Input device)

Printer (output Device)

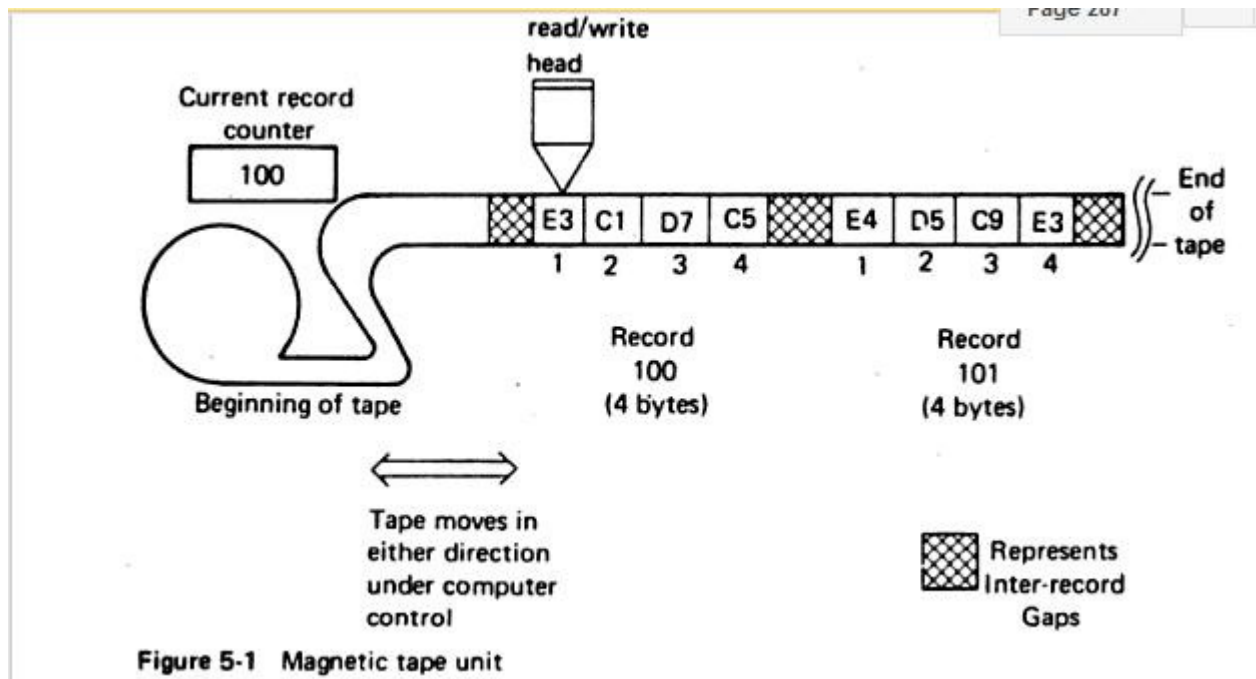
## 2. Storage Device :

- It is a mechanism by which the computer may store the information in such a way that information may be retrieved at a later time.
- There are following three types of storage devices.
  1. Serial Access Devices.
  2. Completely Direct Access.
  3. Direct Access Storage device (DASD).

1. **Serial Access Devices** : It can be characterized as one that depends on physical sequential positioning and accessing information.

- Example, Magnetic tape unit where information is stored as a group of bytes called records of any length.
- Each record can be identified by its physical position on the tape

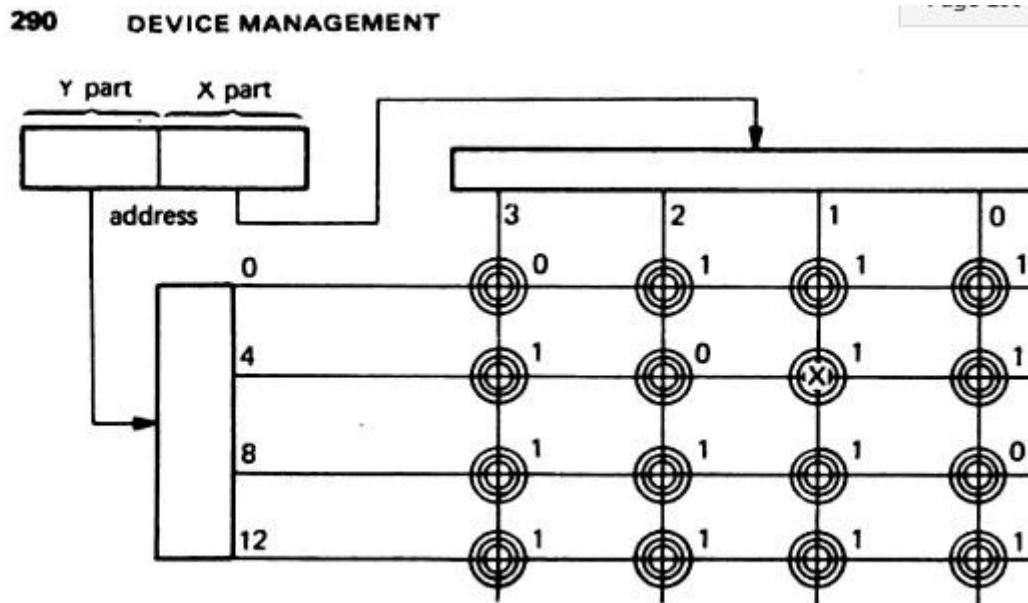
Example, the first record is one second record is second and so on.



## 2. Completely direct access devices :

- A completely direct access device is one in which the access time is constant.
- It requires less time and less cost.
- Example, Magnetic code memory,
- semiconductor memory

- And read only wired memory
- In figure the 16 circles represent magnetic cores. All cores are connected by wires, each core having two selection wires through it.
- When hardware receives a read request for some address, the address is decomposed into x and y position. The hardware then selects the appropriate core.



**Figure 5-3 Core memory**

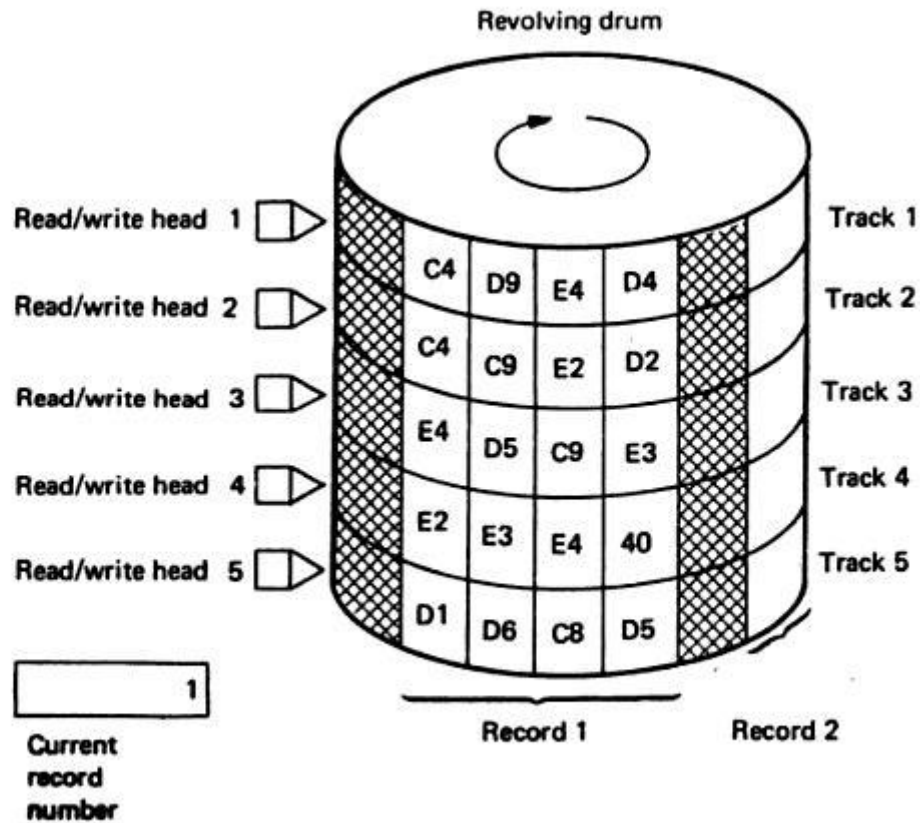
### 3.Direct Access storage device (DASD) :

- A direct access device is one that is characterized by small variances in the access time. These have been called direct access storage devices.
- Example :Magnetic drum and disk.
- In a magnetic drum can be viewed as several adjacent strips of magnetic tape wrapped around a drum so that the ends of each tape strip join. Each tape strip ,called track, and its separate read/write head.
- The drum continuously revolves at high speed so that record repeatedly pass under the read/write head. (e.g record1,record2.., then record1 again,record2...etc). Each individual record is identified by a track number and then a record number.

Fixed head drum

Example: record (2,1) is X'C4C9E2D2' and  
 record (5,1) is X'D1D6C5D5'.

Each record is identified by track no and record number.



**Figure 5-4 Fixed-head drum DASD**

#### 4.2.1 Three Factors Contribute To Time Required To Access a File

- **Seek time** -- time required to position the read/write head on the proper track.
- **Search time (rotational delay)** -- time it takes to rotate DASD until requested record is under read/write head.
- **Transfer time** -- when data is actually transferred from secondary storage to main memory.

#### [4.2.2] Four types of technology.

##### [1] CD-ROM Technology -- commonly used optical storage DASD.

- Stores very large databases, reference works, complex games, large software packages, system documentation, and user training material.

##### [2] CD-Recordable Technology (CD-R):

- CD-R drives record data on optical discs using a write-once technique.
- WORM (write once, read many).
- Only a finite amount of data can be recorded on each disc and, once data is written, it can't be erased or modified.

##### [3] CD-Rewritable Technology (CD-RW)

- CD-RW discs can be written and rewritten many times by focusing a low-energy laser beam on surface.
- Useful for storing large quantities of data and for sound, graphics, and multimedia applications.

##### [4] Digital Video Disc (DVD) Technology

- DVD uses infrared laser to read disc (holds equivalent of 13 CD-ROM discs).
- By using compression technologies, has more than enough space to hold a 2-hour of movie with enhanced audio.
  - Single layered DVDs can hold 4.7 GB
  - Double-layered disc can hold 8.5 GB on each side of the disc.DVDs are used to store music, movies, and multimedia applications.
- **DVD-RAM** is a writable technology that uses a red laser to read, modify, and write data to DVD discs.

#### [4.2.3] Direct Memory Access (DMA)

- It is an I/O technique that allows a control unit to access main memory directly.
- Once reading or writing begins, remainder of data can be transferred to and from memory without CPU intervention.
- To activate this process CPU sends enough info to control unit to initiate transfer of data
- Then CPU goes to another task while control unit completes transfer independently.

- Without DMA the CPU is responsible for moving or transferring the data which is time consuming process.

#### [4.2.4] Device Driver

- It is a software module that controls an I/O device and handle its interrupts.
- Each device has its own device driver.

#### [4.2.5] BIOS :

- BIOS is the position of a device manager that handles the device driver software and is stored in both ROM and RAM.
- We can also write on your own device driver, if you have know assembly languages and information about hardware.

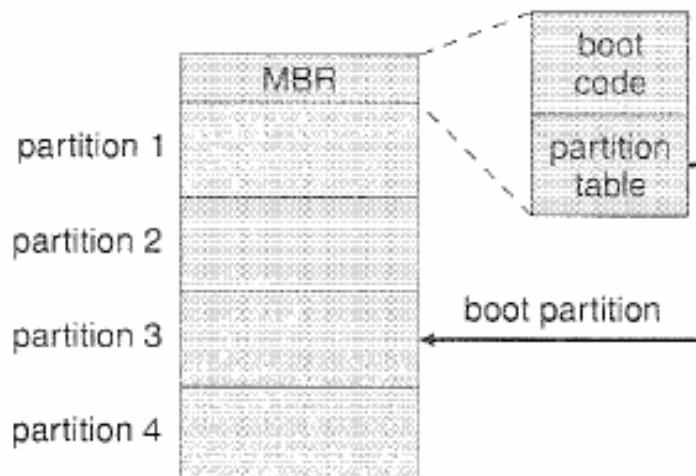
### [4.3] Disk space Management.

#### [4.3.1] Boot block:

➤ To reboot or to start the computer, it must have bootstrap program. Which initialize the system from CPU registers to device controller and the contained of main memory and started a operating system.

- Boot strap is stored in read only memory (RAM).
- It stored in the boot block at the fixed location on the disk.
- A disk that has boot partition is call boot disk.
- Hard disk is divided into one or more partitions and a partition

Identified as a boot partition that contains operating system and device Drivers.



➤ Operating system stored boot card in the first sector of hard disk Which is called master boot records (MBR) and this sector is known As boot sector.

#### [4.3.2] Bad block

- If one or more sector of hard disk become defective that known
- as bad block.
- This block are handle in different ways.
  1. Bad block are handle manually.
  2. Sector sparing.
  3. Sector slipping.

1. **Bad block are handle manually** : in this process disk is scanned to find the bad blocks. After that a special value is returned in to FAT( File allocation table) to allocation routine not to that block.

#### 2. Sector sparing :

- the device controller maintain a list of bad block on the disk.
- The list is initialized during low level formatting. Low level Formatting also set spare sectors, which replaces bad sector logically. This process is known as sector sparing or forwarding.

#### 3. Sector slipping :

- Some controller can be instructed to replace a bad block by sector slipping.

E.G suppose logical block no 17 becomes defective and first available spare sector is 202. then sector slipping technique ramps all the sector form 17 to 202. the sector 202 is copied into another spare sectors. Then 201 to 202 and so on. Until the sector 18 is copied the sector 19. In this way this technique frees Up the space of sector 18 so, sector 17 can we mapped into it.

#### [4.3.3] Buffering :

- A buffer is a memory area that stores data being transferred between two devices or between a device and application.

#### [4.3.4] Spooling :

- Spooling is a buffer that holds output for a device, such as printer. Each application output is spooled to a separate disk file. When the application finished and printing the spooling system queues the corresponding spool file for output to printer.

- The spooling system copied the queues spool files to the printer on at a time.
- Device Manager divides task into 3 parts, with each handled by specific software component of I/O subsystem.

#### [4.3.5] Management of I/O Requests

- **I/O traffic controller** watches status of all devices, control units, and channels.
- **I/O scheduler** implements policies that use to allocate devices, control units, and channels. Access I/O request from jobs.
- **I/O device handler** processes the I/O interrupts, handles error conditions, and provides detailed scheduling algorithms, which are extremely device dependent.

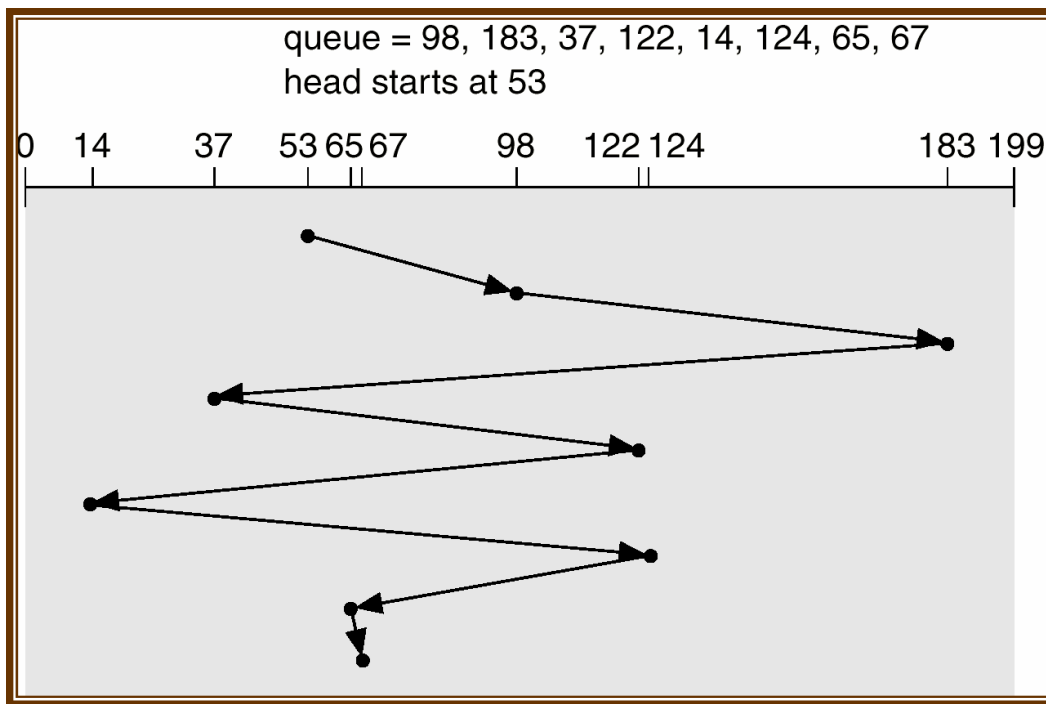
#### [4.4] Disk Scheduling.

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth.
- Access time has two major components
  - *Seek time* is the time for the disk are to move the heads to the cylinder containing the desired sector.
  - *Rotational latency* is the additional time waiting for the disk to rotate the desired sector to the disk head.
- Minimize seek time
- Seek time  $\approx$  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.



#### [4.4.1] FCFS : FIRST COME FIRST SERVES.

- First-come first served scheduling follow first in first out method.
- It serves the request in the same order as they are received
- The average waiting time for FCFS is often quite long. It is non-preemptive.
- In the below example the disk head is initially at cylinder 53, it will first move 53 to 98 then to 183,37,122,14,124,65 and finally to 67 for a total head movement of 640 cylinders.



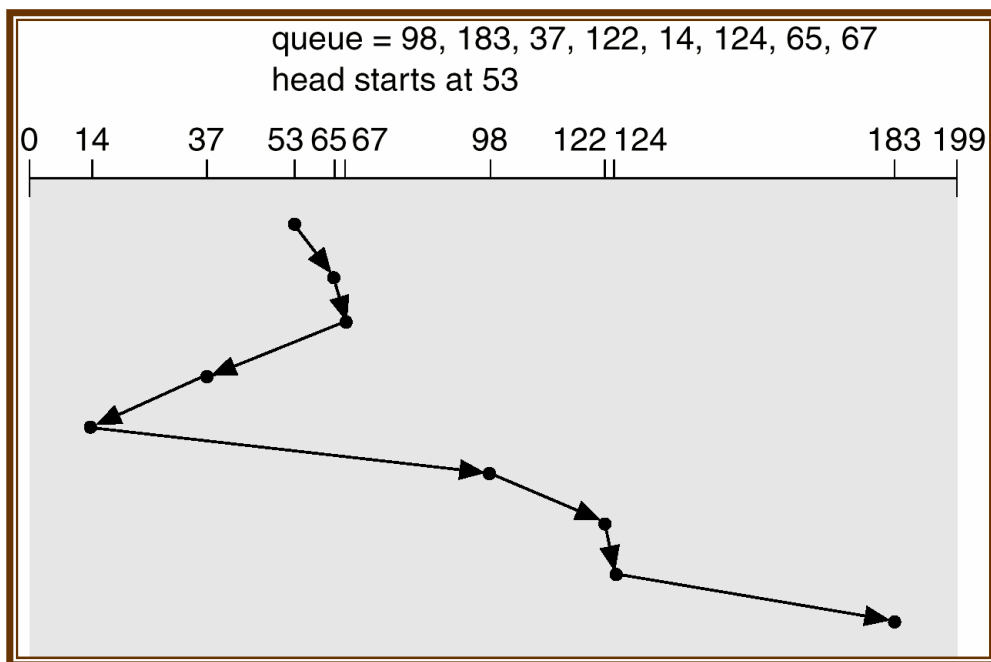
$$\text{ex} - 98 - 53 = 45$$

$$\begin{aligned} \text{Total head movement} &= 45 + 85 + 146 + 85 + 108 + 110 + 59 + 2 \\ &= 640 \end{aligned}$$

- **Advantages :**
  - Simple ,fair to all request.
  - No starvation.
- **Disadvantages :**
  - Not efficient because the average seek time is high.

## SSTF : SHORTEST SEEKTIME FIRST SCHEDULING

- 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.

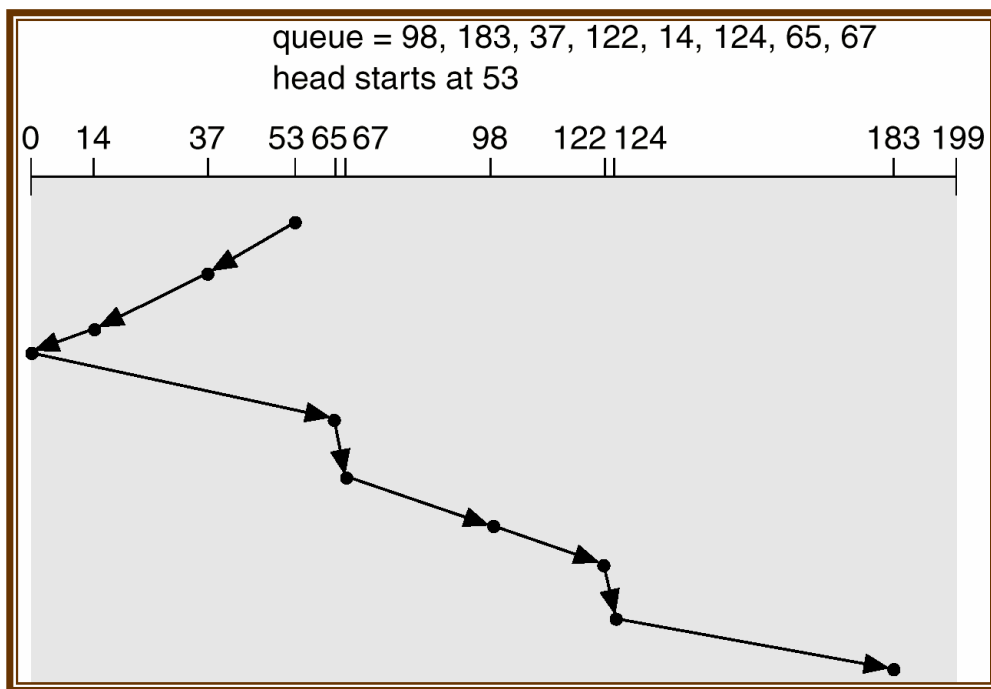


$$\begin{aligned} \text{Total head movement} &= 12+2+30+23+84+24+2+59 \\ &= 236 \end{aligned}$$

- Advantages :
  - More efficient than FCFS.
- Disadvantages :
  - Starvation is possible for request involving longer seek-time.

## SCAN SCHEDULING

- 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*.
- The head starts at the one end of the disk and moves toward on the another end. it serves all the request coming in the way.
- After reaching another end the direction of head movement is reverse.
- Illustration shows total head movement of 236 cylinders.

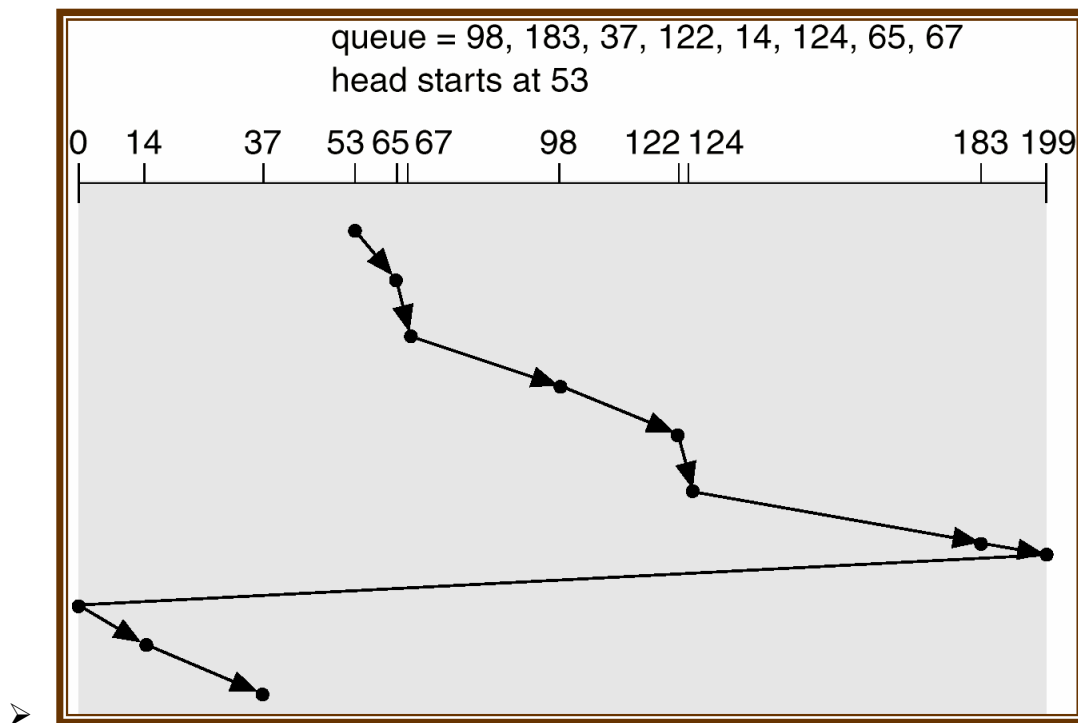


$$\begin{aligned}\text{Total head movement} &= 16+23+14+65+2+31+24+2+59 \\ &= 236\end{aligned}$$

- **Advantages :**
  - More efficient than FCFS.
  - No starvation for any request.
- **Disadvantages :**Not so fair, because cylinder which are just behind the head will wait longer.
  - Requires extra head movement between two extreme points.
  - For example, after serving 5<sup>th</sup> cylinder, there is no need to visit 0<sup>th</sup> cylinder.

## C-SCAN [Circular SCAN] SCHEDULING

- The circular SCAN algorithm improves upon SCAN by treating all the request in a circular queue fashion.- once the head reaches the end of the disk, it returns to the other end without processing any request, and then starts again from the beginning of the disk.
- 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.
- Illustration shows **total head movement of 382 cylinders**, plus return time.
- Provides a more uniform wait time than SCAN.



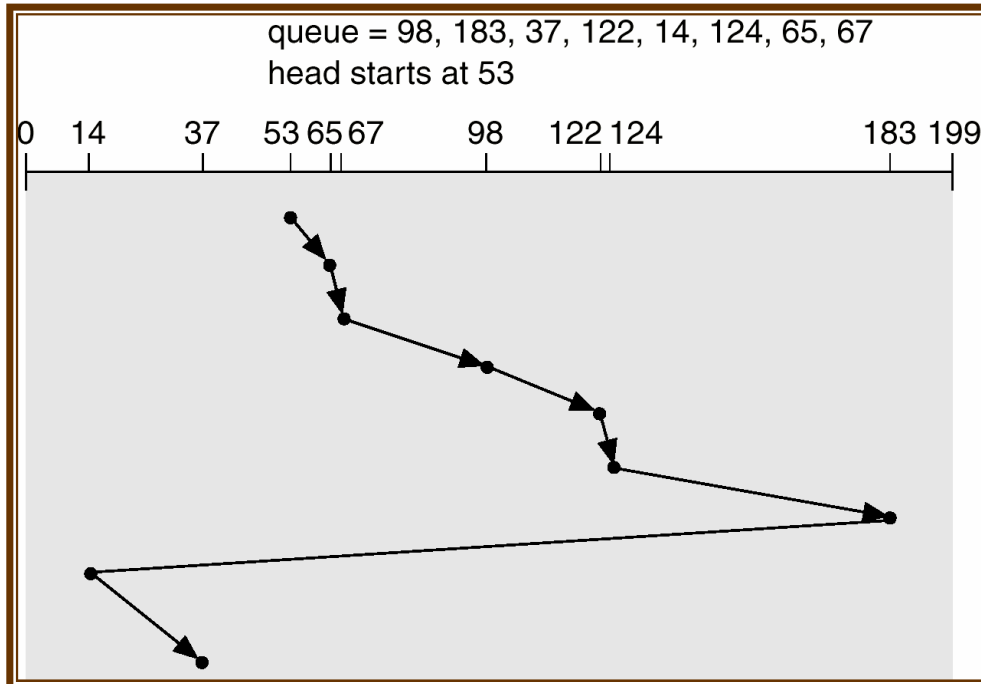
$$\begin{aligned} \text{Total head movement} &= 12+2+31+24+2+59+16+199+14+23 \\ &= 382 \end{aligned}$$

## LOOK

- it is same like SCAN scheduling but the difference that end points are not visited unnecessary.
- 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.
- **Advantages :**
- More efficient than FCFS.
- No starvation for any request.
- No extra head movement.
  
- **Disadvantages :**
- Not so fair, because cylinder behind the head with longer.

## C-LOOK [ circular – 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.



## 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.
- 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.

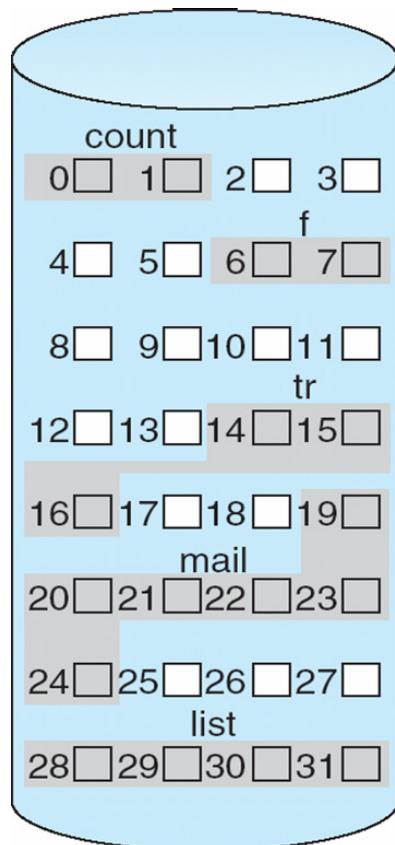
- Newer disks have
- Lower seek time relative to latency
- Built in scheduling - send requests in batches
- OS level scheduling is important at a coarser grain level, for important disk activity, e.g., VM

#### **[4.5] Allocation Methods**

- An allocation method refers to how disk blocks are allocated for files:
- **Contiguous allocation**
- **Linked allocation**
- **Indexed allocation**

#### **1. Contiguous allocation :**

- Each file occupies a set of contiguous blocks on disk.
- When file is created, a disk is searched to find free memory having enough size to store a file, once the free memory is found then memory is allocated.
- Each directory entry contains.
  - File name
  - Starting address of the first block.
  - Length in block.



directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

### Example of Contiguous Allocation Methods

➤ **Advantages :**

- It supports fast sequential and direct access.
- It provides a good performance.
- The number of disk seek required is minimal.
- Simple to implement, there are two things required.
  1. starting block number
  2. length of the file.

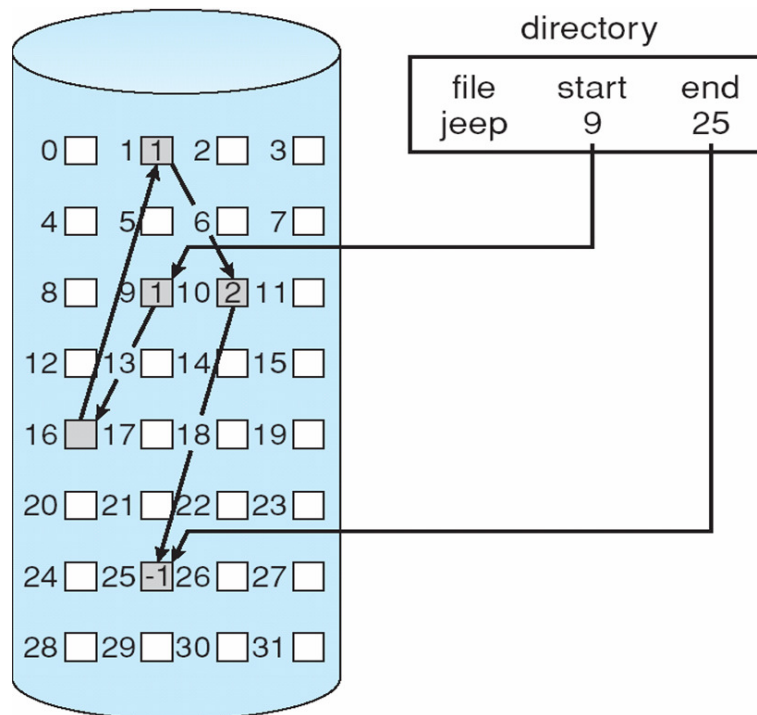
➤ **Disadvantages :**

- Find free spaces for new file, its time consuming.
- If size of the existing file increase that it may not be possible to accommodate such a data.
- External fragmentation is possible.



## 2. Linked Allocation Methods

- In this method, each file is linked list of disk blocks , each block contain pointer to next block.
- Each directory entry contains.
  - File name
  - Starting block no.
  - Ending block no.



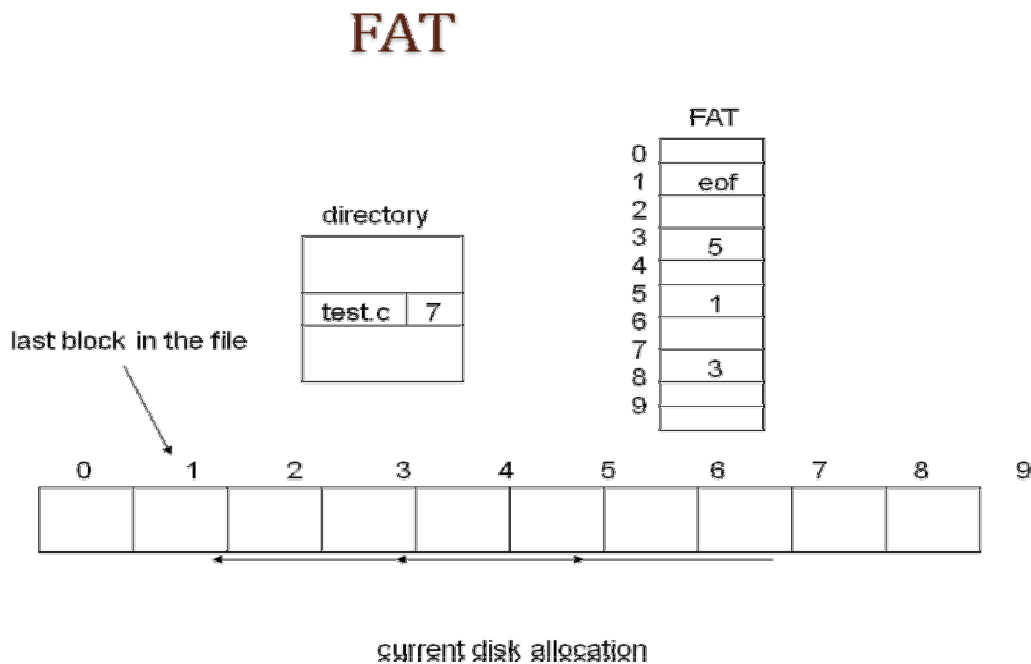
- When new file is created a new directory entry is created an initially it contain null Value.
- A write operation to the required free blocks to be added to the file at the end of link list.
- To read a file all blocks are read by the pointer from 1 block to another.
- A view of the linked list.
- --→9 →16 →1 →10 →25
- **Advantages :**
  - It does not suffer from external fragmentation.

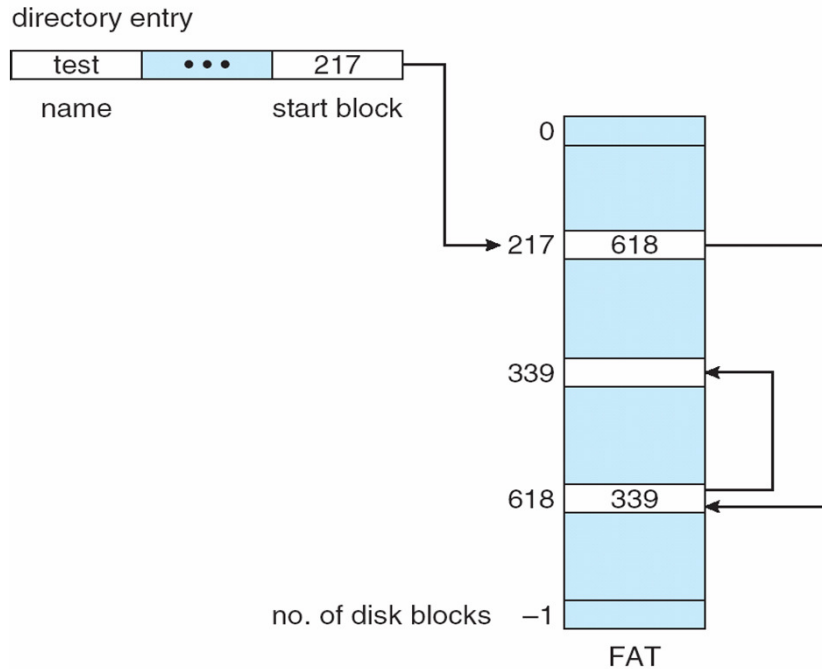
➤ **Disadvantages :**

- File access is time consuming because it required to search link list to reach at same block.
- Direct access is not supported.
- Memory space required for the pointers.
- Reliability is compromised if the pointers are lost or damaged.

**File Allocation Table (FAT)**

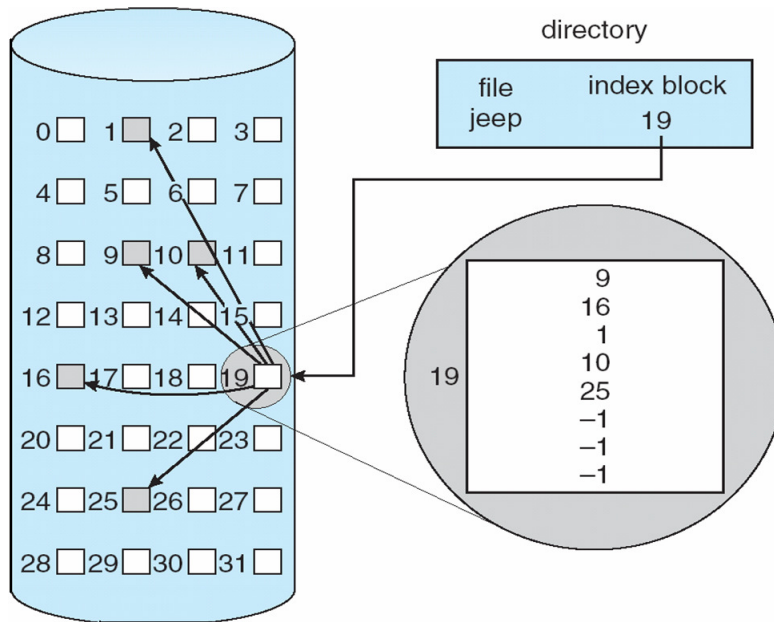
- This is an extension of the linked allocation
- Instead of putting the pointers in the file, keep a table of the pointers around
- This table can be quickly searched to find any random block in the file.
- All the blocks on disk must be included in the table
- A nice side effect of FAT is for the free list,
  - whether a block is free or not can be recorded in the table.
- This method works well for both sequential and random access
- This is the method used by Windows.





### 3.Index allocation :

- link allocation method can not support direct access to solved this problem index allocation is used.
- It store all pointer together in the index table.
- It bring all the pointer together in to one location called the index block.
- Each file contain its own index block, the entry in the index block points to block no of the file
- Directory entry contains.
  - File name
  - Index block
- When the first block is return a free block is allocated and its address shown in the entry in the index block.



➤ **Advantages :**

- No external fragmentation.
- Support Direct access.
- Support sequential access
- Number of disk seeks is minimal.

➤ **Disadvantages :**

- It suffer from wastage spaces.
- Maximum allowable file size depends on size of an index block.