Lecture 13

Mass-Storage Structure

Lecture Information

Ceng328 *Operating Systems* at May 18, 2010

Dr. Cem Özdoğan
Computer Engineering Department
Çankaya University

Contents

Mass-Storage Structure
Overview of Mass-Storage Structure
Magnetic Disks
Disk Structure
Disk Attachment
Host-Attached Storage
Network-Attached Storage
Storage-Area Network
Disk Scheduling
FCFS Scheduling
SSTF Scheduling
SCAN Scheduling
C-SCAN Scheduling
LOOK Scheduling
Disk Management
Disk Formatting
RAID Structure
Improvement of Reliability via Redundancy
Improvement in Performance via Parallelism
RAID Levels

Mass-Storage Structure

- Huge speed gap between memory and disk.

Mass-Storage Structure

- Huge speed gap between memory and disk.
- Disk throughput is extremely sensitive to

Mass-Storage Structure

- Huge speed gap between memory and disk.
- Disk throughput is extremely sensitive to
  - Request order $\Longrightarrow$ **Disk Scheduling**.

Dr. Cem Özdoğan

Mass-Storage Structure

- Huge speed gap between memory and disk.
- Disk throughput is extremely sensitive to
  - Request order $\Longrightarrow$ **Disk Scheduling**.
  - Placement of data on the disk $\Longrightarrow$ **File System Design**.

Mass-Storage Structure

- Huge speed gap between memory and disk.
- Disk throughput is extremely sensitive to
  - Request order $\Longrightarrow$ **Disk Scheduling**.
  - Placement of data on the disk $\Longrightarrow$ **File System Design**.
- Formatting

Mass-Storage Structure

- Huge speed gap between memory and disk.
- Disk throughput is extremely sensitive to
  - Request order $\Longrightarrow$ **Disk Scheduling**.
  - Placement of data on the disk $\Longrightarrow$ **File System Design**.
- Formatting
  - *Physical*: divide the blank slate into sectors identified by headers containing such information as sector number.

Dr. Cem Özdoğan

Mass-Storage Structure

- Huge speed gap between memory and disk.
- Disk throughput is extremely sensitive to
  - Request order $\Longrightarrow$ **Disk Scheduling**.
  - Placement of data on the disk $\Longrightarrow$ **File System Design**.
- Formatting
  - *Physical*: divide the blank slate into sectors identified by headers containing such information as sector number.
  - *Logical*: marking bad blocks; partitioning and writing a blank directory on disk; installing file allocation tables, and other relevant information (file system initialization).

Mass-Storage Structure

- Huge speed gap between memory and disk.
- Disk throughput is extremely sensitive to
  - Request order $\Longrightarrow$ **Disk Scheduling**.
  - Placement of data on the disk $\Longrightarrow$ **File System Design**.
- Formatting
  - *Physical*: divide the blank slate into sectors identified by headers containing such information as sector number.
  - *Logical*: marking bad blocks; partitioning and writing a blank directory on disk; installing file allocation tables, and other relevant information (file system initialization).
- Reliability

Mass-Storage Structure

- Huge speed gap between memory and disk.
- Disk throughput is extremely sensitive to
  - Request order $\Longrightarrow$ **Disk Scheduling**.
  - Placement of data on the disk $\Longrightarrow$ **File System Design**.
- Formatting
  - *Physical*: divide the blank slate into sectors identified by headers containing such information as sector number.
  - *Logical*: marking bad blocks; partitioning and writing a blank directory on disk; installing file allocation tables, and other relevant information (file system initialization).
- Reliability
  - RAIDs (Redundant Array of Inexpensive Disks): various levels, level 0 is disk striping).

Dr. Cem Özdoğan

Magnetic Disks I

Magnetic disks provide the bulk of secondary storage for modern computer systems. Conceptually, disks are relatively simple (see Fig. 1).

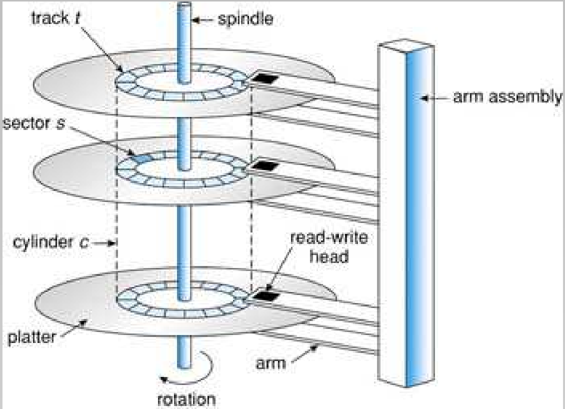


Figure: Moving-head disk mechanism.

Magnetic Disks II

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

Dr. Cem Özdoğan

Magnetic Disks II

- Each disk **platter** has a flat circular shape, like a CD.
  - Common platter diameters range from 1.8 to 5.25 inches.

Magnetic Disks II

- Each disk **platter** has a flat circular shape, like a CD.
  - Common platter diameters range from 1.8 to 5.25 inches.
  - The two surfaces of a platter are covered with a magnetic material.

Magnetic Disks II

- Each disk **platter** has a flat circular shape, like a CD.
  - Common platter diameters range from 1.8 to 5.25 inches.
  - The two surfaces of a platter are covered with a magnetic material.
  - The heads are attached to a disk arm that moves all the heads as a unit.

Magnetic Disks II

- Each disk **platter** has a flat circular shape, like a CD.
  - Common platter diameters range from 1.8 to 5.25 inches.
  - The two surfaces of a platter are covered with a magnetic material.
  - 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**.

Magnetic Disks II

- Each disk **platter** has a flat circular shape, like a CD.
  - Common platter diameters range from 1.8 to 5.25 inches.
  - The two surfaces of a platter are covered with a magnetic material.
  - 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**.

Magnetic Disks II

- Each disk **platter** has a flat circular shape, like a CD.
  - Common platter diameters range from 1.8 to 5.25 inches.
  - The two surfaces of a platter are covered with a magnetic material.
  - 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.

Magnetic Disks II

- Each disk **platter** has a flat circular shape, like a CD.
  - Common platter diameters range from 1.8 to 5.25 inches.
  - The two surfaces of a platter are covered with a magnetic material.
  - 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.
- When the disk is in use, a drive motor spins it at high speed.

Magnetic Disks II

- Each disk **platter** has a flat circular shape, like a CD.
    - Common platter diameters range from 1.8 to 5.25 inches.
    - The two surfaces of a platter are covered with a magnetic material.
    - 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.
- When the disk is in use, a drive motor spins it at high speed.
- Most drives rotate 60 to 200 times per second (3600 to 12000 rpm).

Magnetic Disks III

- Execution of a disk operation involves (see Fig. 2)

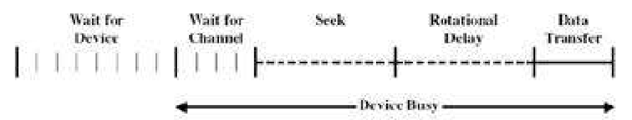


Figure: Disk Performance.

Magnetic Disks III

- Execution of a disk operation involves (see Fig. 2)

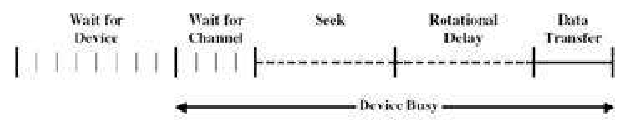


Figure: Disk Performance.

1. **Wait time**: the process waits to be granted device access;

Magnetic Disks III

- Execution of a disk operation involves (see Fig. 2)

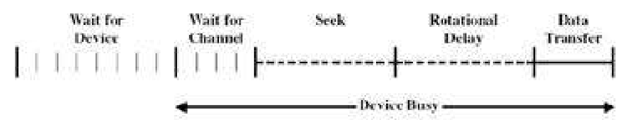


Figure: Disk Performance.

1. **Wait time**: the process waits to be granted device access;
   - *Wait for device*

Magnetic Disks III

- Execution of a disk operation involves (see Fig. 2)

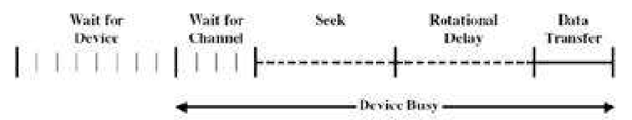


Figure: Disk Performance.

1. **Wait time**: the process waits to be granted device access;
   - *Wait for device*
   - *Wait for channel*

Magnetic Disks III

- Execution of a disk operation involves (see Fig. 2)

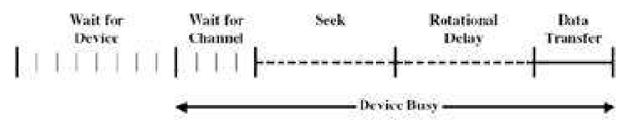


Figure: Disk Performance.

1. **Wait time**: the process waits to be granted device access;
    - *Wait for device*
    - *Wait for channel*
2. **Positioning time**: time hardware need to position the head.

Magnetic Disks III

- Execution of a disk operation involves (see Fig. 2)

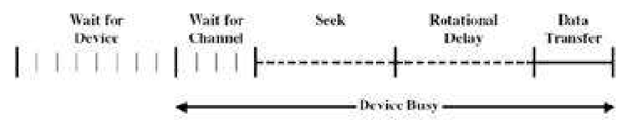


Figure: Disk Performance.

1. **Wait time**: the process waits to be granted device access;
   - *Wait for device*
   - *Wait for channel*
2. **Positioning time**: time hardware need to position the head.
   - consists of the time to move the disk arm to the desired cylinder, called the **seek time** T_S (in milliseconds),

Magnetic Disks III

- Execution of a disk operation involves (see Fig. 2)

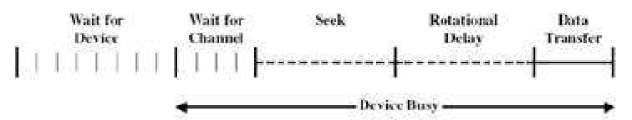


Figure: Disk Performance.

1. **Wait time**: the process waits to be granted device access;
   - *Wait for device*
   - *Wait for channel*
2. **Positioning time**: time hardware need to position the head.
   - consists of the time to move the disk arm to the desired cylinder, called the **seek time** T_s (in milliseconds),
   - the time for the desired sector to rotate to the disk head, called the **rotational latency** (in milliseconds). Rotational speed, r, of 6000 to 12000 rpm.

Magnetic Disks III

- Execution of a disk operation involves (see Fig. 2)

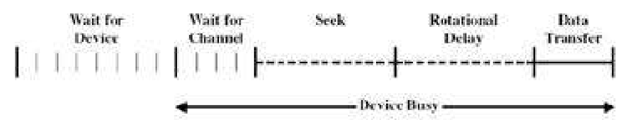


Figure: Disk Performance.

1. **Wait time**: the process waits to be granted device access;
   - *Wait for device*
   - *Wait for channel*
2. **Positioning time**: time hardware need to position the head.
   - consists of the time to move the disk arm to the desired cylinder, called the **seek time** T_S (in milliseconds),
   - the time for the desired sector to rotate to the disk head, called the **rotational latency** (in milliseconds). Rotational speed, r, of 6000 to 12000 rpm.
3. **Transfer rate** is the rate at which data flow between the drive and the computer (megabytes of data per second). **Transfer time**: to transfer b bytes, with N bytes per track;

$$T = \frac{b}{rN}$$

Magnetic Disks IV

- **Total average access time;**

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

Magnetic Disks IV

- **Total average access time**;

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

- A timing comparison for $T_s = 2$ ms, $r = 10000$rpm, 512B sector size, 320 sectors per track, 1.3 MB file size.

Magnetic Disks IV

- **Total average access time**;

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

- A timing comparison for $T_s = 2$ ms, $r = 10000$rpm, 512B sector size, 320 sectors per track, 1.3 MB file size.
  - At 10000 rpm, one revolution per 6ms$\Rightarrow$ average delay 3ms (=(60 second/10000) & $\frac{1}{r} = 6$).

Magnetic Disks IV

- **Total average access time**;

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

- A timing comparison for $T_s = 2$ ms, $r = 10000$rpm, 512B sector size, 320 sectors per track, 1.3 MB file size.
  - At 10000 rpm, one revolution per 6ms$\Rightarrow$ average delay 3ms (=(60 second/10000) & $\frac{1}{r} = 6$).
  - Read a file with 2560 sectors (=(1.3MB/512))

Magnetic Disks IV

- **Total average access time**;

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

- A timing comparison for $T_s = 2$ ms, $r = 10000$rpm, 512B sector size, 320 sectors per track, 1.3 MB file size.
  - At 10000 rpm, one revolution per 6ms$\Rightarrow$ average delay 3ms (=(60 second/10000) & $\frac{1}{r} = 6$).
  - Read a file with 2560 sectors (=(1.3MB/512))

1. File stored compactly (8 adjacent tracks (=(2560/320))). Read first track;

Average seek	2ms
Rot. Delay	3ms
Read 320 sectors	6ms ($\frac{1}{r} = 6$, b=512*320 & N=512*320)
Total	11ms
All sectors	11+7*9=74ms

Magnetic Disks IV

- **Total average access time**;

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

- A timing comparison for $T_s = 2$ ms, $r = 10000$rpm, 512B sector size, 320 sectors per track, 1.3 MB file size.
  - At 10000 rpm, one revolution per 6ms$\Rightarrow$ average delay 3ms (=(60 second/10000) & $\frac{1}{r} = 6$).
  - Read a file with 2560 sectors (=(1.3MB/512))

1. File stored compactly (8 adjacent tracks (=(2560/320))). Read first track;

Average seek	2ms
Rot. Delay	3ms
Read 320 sectors	6ms ($\frac{1}{r} = 6$, b=512*320 & N=512*320)
Total	11ms
All sectors	11+7*9=74ms

2. Sectors distributed randomly over the disk: Read any sector

Average seek	2ms
Rot. Delay	3ms
Read 1 sectors	0.01875ms (=(6/320); $\frac{1}{r} = 6$, b=512, N=512*320)
Total	5.01875ms
All	2560*5.01875=12848ms

Magnetic Disks V

- Disk Performance is entirely dominated by *Seek* and *Rotational Delays*.

Magnetic Disks V

- Disk Performance is entirely dominated by *Seek* and *Rotational Delays*.
  - It will only get worse as capacity increases much faster than increase in seek time and rotation speed.

Magnetic Disks V

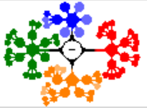
- Disk Performance is entirely dominated by *Seek* and *Rotational Delays*.
    - It will only get worse as capacity increases much faster than increase in seek time and rotation speed.
    - It has been easier to spin the disk faster than improve seek time.

Magnetic Disks V

- Disk Performance is entirely dominated by *Seek* and *Rotational Delays*.
  - It will only get worse as capacity increases much faster than increase in seek time and rotation speed.
  - It has been easier to spin the disk faster than improve seek time.
- A disk drive is attached to a computer by a set of wires called an **I/O bus**.

Magnetic Disks V

- Disk Performance is entirely dominated by *Seek* and *Rotational Delays*.
  - It will only get worse as capacity increases much faster than increase in seek time and rotation speed.
  - It has been easier to spin the disk faster than improve seek time.
- 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



Magnetic Disks V

- Disk Performance is entirely dominated by *Seek* and *Rotational Delays*.
  - It will only get worse as capacity increases much faster than increase in seek time and rotation speed.
  - It has been easier to spin the disk faster than improve seek time.
- 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),

Magnetic Disks V

- Disk Performance is entirely dominated by *Seek* and *Rotational Delays*.
  - It will only get worse as capacity increases much faster than increase in seek time and rotation speed.
  - It has been easier to spin the disk faster than improve seek time.
- 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),

Magnetic Disks V

- Disk Performance is entirely dominated by *Seek* and *Rotational Delays*.
  - It will only get worse as capacity increases much faster than increase in seek time and rotation speed.
  - It has been easier to spin the disk faster than improve seek time.
- 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),

Magnetic Disks V

- Disk Performance is entirely dominated by *Seek* and *Rotational Delays*.
    - It will only get worse as capacity increases much faster than increase in seek time and rotation speed.
    - It has been easier to spin the disk faster than improve seek time.
- 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),

Magnetic Disks V

- Disk Performance is entirely dominated by *Seek* and *Rotational Delays*.
  - It will only get worse as capacity increases much faster than increase in seek time and rotation speed.
  - It has been easier to spin the disk faster than improve seek time.
- 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),

Magnetic Disks V

- Disk Performance is entirely dominated by *Seek* and *Rotational Delays*.
  - It will only get worse as capacity increases much faster than increase in seek time and rotation speed.
  - It has been easier to spin the disk faster than improve seek time.
- 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),
  - SCSI buses

Magnetic Disks VI

- The data transfers on a bus are carried out by special electronic processors called **controllers**.

Magnetic Disks VI

- The data transfers on a bus are carried out by special electronic processors called **controllers**.

1. The **host controller** is the controller at the computer end of the bus.

Magnetic Disks VI

- The data transfers on a bus are carried out by special electronic processors called **controllers**.

1. The **host controller** is the controller at the computer end of the bus.
   - To perform a disk I/O operation, the computer places a command into the host controller (memory-mapped I/O ports).

Magnetic Disks VI

- The data transfers on a bus are carried out by special electronic processors called **controllers**.

1. The **host controller** is the controller at the computer end of the bus.
    - To perform a disk I/O operation, the computer places a command into the host controller (memory-mapped I/O ports).
    - The host controller then sends the command via messages to the disk controller.

Magnetic Disks VI

- The data transfers on a bus are carried out by special electronic processors called **controllers**.

1. The **host controller** is the controller at the computer end of the bus.
    - To perform a disk I/O operation, the computer places a command into the host controller (memory-mapped I/O ports).
    - The host controller then sends the command via messages to the disk controller.
    - The disk controller operates the disk-drive hardware to carry out the command.

Magnetic Disks VI

- The data transfers on a bus are carried out by special electronic processors called **controllers**.

1. The **host controller** is the controller at the computer end of the bus.
   - To perform a disk I/O operation, the computer places a command into the host controller (memory-mapped I/O ports).
   - The host controller then sends the command via messages to the disk controller.
   - The disk controller operates the disk-drive hardware to carry out the command.
2. A **disk controller** is built into each disk drive.

Magnetic Disks VI

- The data transfers on a bus are carried out by special electronic processors called **controllers**.

1. The **host controller** is the controller at the computer end of the bus.
    - To perform a disk I/O operation, the computer places a command into the host controller (memory-mapped I/O ports).
    - The host controller then sends the command via messages to the disk controller.
    - The disk controller operates the disk-drive hardware to carry out the command.
2. A **disk controller** is built into each disk drive.
    - Disk controllers usually have a built-in cache.

Magnetic Disks VI

- The data transfers on a bus are carried out by special electronic processors called **controllers**.

1. The **host controller** is the controller at the computer end of the bus.
    - To perform a disk I/O operation, the computer places a command into the host controller (memory-mapped I/O ports).
    - The host controller then sends the command via messages to the disk controller.
    - The disk controller operates the disk-drive hardware to carry out the command.
2. A **disk controller** is built into each disk drive.
    - Disk controllers usually have a built-in cache.
    - Data transfer at the disk drive happens

Magnetic Disks VI

- The data transfers on a bus are carried out by special electronic processors called **controllers**.

1. The **host controller** is the controller at the computer end of the bus.
   - To perform a disk I/O operation, the computer places a command into the host controller (memory-mapped I/O ports).
   - The host controller then sends the command via messages to the disk controller.
   - The disk controller operates the disk-drive hardware to carry out the command.
2. A **disk controller** is built into each disk drive.
   - Disk controllers usually have a built-in cache.
   - Data transfer at the disk drive happens
     - between the cache and the disk surface,

Magnetic Disks VI

- The data transfers on a bus are carried out by special electronic processors called **controllers**.

1. The **host controller** is the controller at the computer end of the bus.
    - To perform a disk I/O operation, the computer places a command into the host controller (memory-mapped I/O ports).
    - The host controller then sends the command via messages to the disk controller.
    - The disk controller operates the disk-drive hardware to carry out the command.
2. A **disk controller** is built into each disk drive.
    - Disk controllers usually have a built-in cache.
    - Data transfer at the disk drive happens
        - between the cache and the disk surface,
        - between the cache and the host controller.

Magnetic Disks VII

```
root@ozdogan:~# fdisk -l

Disk /dev/sda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 14593 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x23df34d3

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1         637     5116671   12  Compaq diagnostics
/dev/sda2   *         638        3070    19543072+   c  W95 FAT32 (LBA)
/dev/sda3            3071       14593    92558497+   5  Extended
/dev/sda5            3071       14255    89843481   83  Linux
/dev/sda6           14256       14593     2714953+  82  Linux swap / Solaris
```

Figure: System's partition table.

Disk Structure I

- Modern disk drives are addressed as large one-dimensional arrays of *logical blocks*,

Disk Structure I

- Modern disk drives are addressed as large one-dimensional arrays of *logical blocks*,
- where the logical block is the smallest unit of transfer.

Disk Structure I

- Modern disk drives are addressed as large one-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.

Disk Structure I

- Modern disk drives are addressed as large one-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.
- The one-dimensional array of logical blocks is mapped onto the sectors of the disk sequentially.

Disk Structure I

- Modern disk drives are addressed as large one-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.
- The one-dimensional array of logical blocks is mapped onto the sectors of the disk sequentially.
  - Sector 0 is the first sector of the first track on the outermost cylinder.

Disk Structure I

- Modern disk drives are addressed as large one-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.
- The one-dimensional array of logical blocks is mapped onto the sectors of the disk sequentially.
  - 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.

Disk Structure I

- Modern disk drives are addressed as large one-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.
- The one-dimensional array of logical blocks is mapped onto the sectors of the disk sequentially.
  - 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.
- By using this mapping, we can convert a logical block number into an old-style disk address that consists of a cylinder number, a track number within that cylinder, and a sector number within that track.

Disk Structure I

- Modern disk drives are addressed as large one-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.
- The one-dimensional array of logical blocks is mapped onto the sectors of the disk sequentially.
  - 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.
- By using this mapping, we can convert a logical block number into an old-style disk address that consists of a cylinder number, a track number within that cylinder, and a sector number within that track.
- In practice, it is difficult to perform this translation, for two reasons.

Disk Structure I

- Modern disk drives are addressed as large one-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.
- The one-dimensional array of logical blocks is mapped onto the sectors of the disk sequentially.
  - 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.
- By using this mapping, we can convert a logical block number into an old-style disk address that consists of a cylinder number, a track number within that cylinder, and a sector number within that track.
- In practice, it is difficult to perform this translation, for two reasons.
  - First, most disks have some defective sectors.

Disk Structure I

- Modern disk drives are addressed as large one-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.
- The one-dimensional array of logical blocks is mapped onto the sectors of the disk sequentially.
    - 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.
- By using this mapping, we can convert a logical block number into an old-style disk address that consists of a cylinder number, a track number within that cylinder, and a sector number within that track.
- In practice, it is difficult to perform this translation, for two reasons.
    - First, most disks have some defective sectors.
    - Second, the number of sectors per track is not a constant on some drives.

Disk Structure II

- Let's look more closely at the second reason.

Disk Structure II

- Let’s look more closely at the second reason.
- On media that use **constant linear velocity** (CLV), the density of bits per track is uniform.

Disk Structure II

- Let's look more closely at the second reason.
- On media that use **constant linear velocity** (CLV), the density of bits per track is uniform.
    - The farther a track is from the center of the disk, the greater its length, so the more sectors it can hold.

Disk Structure II

- Let's look more closely at the second reason.
- On media that use **constant linear velocity** (CLV), the density of bits per track is uniform.
  - The farther a track is from the center of the disk, the greater its length, so the more sectors it can hold.
  - As we move from outer zones to inner zones, the number of sectors per track decreases (see Fig. 4).

Disk Structure II

- Let's look more closely at the second reason.
- On media that use **constant linear velocity** (CLV), the density of bits per track is uniform.
  - The farther a track is from the center of the disk, the greater its length, so the more sectors it can hold.
  - As we move from outer zones to inner zones, the number of sectors per track decreases (see Fig. 4).
- Alternatively, the disk rotation speed can stay constant, and the density of bits decreases from inner tracks to outer tracks to keep the data rate constant.

Disk Structure II

- Let's look more closely at the second reason.
- On media that use **constant linear velocity** (CLV), the density of bits per track is uniform.
  - The farther a track is from the center of the disk, the greater its length, so the more sectors it can hold.
  - As we move from outer zones to inner zones, the number of sectors per track decreases (see Fig. 4).
- Alternatively, the disk rotation speed can stay constant, and the density of bits decreases from inner tracks to outer tracks to keep the data rate constant.
- This method is used in hard disks and is known as **constant angular velocity (CAV)**.

Disk Structure III

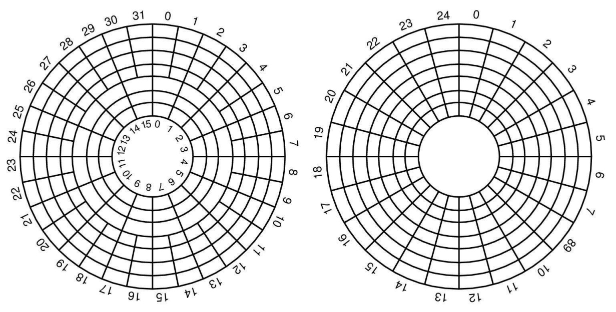


Figure: Physical geometry of a disk with two zones and a possible virtual geometry for this disk.

Disk Structure IV

- Evolution of Disk Hardware (see Fig. 5)

Parameter	IBM 360-KB floppy disk	WD 18300 hard disk
Number of cylinders	40	10601
Tracks per cylinder	2	12
Sectors per track	9	281 (avg)
Sectors per disk	720	35742000
Bytes per sector	512	512
Disk capacity	360 KB	18.3 GB
Seek time (adjacent cylinders)	6 msec	0.8 msec
Seek time (average case)	77 msec	6.9 msec
Rotation time	200 msec	8.33 msec
Motor stop/start time	250 msec	20 sec
Time to transfer 1 sector	22 msec	17 μsec

Figure: Disk parameters for the original IBM PC floppy disk and a Western Digital WD 18300 hard disk.

Disk Structure IV

- Evolution of Disk Hardware (see Fig. 5)

Parameter	IBM 360-KB floppy disk	WD 18300 hard disk
Number of cylinders	40	10601
Tracks per cylinder	2	12
Sectors per track	9	281 (avg)
Sectors per disk	720	35742000
Bytes per sector	512	512
Disk capacity	360 KB	18.3 GB
Seek time (adjacent cylinders)	6 msec	0.8 msec
Seek time (average case)	77 msec	6.9 msec
Rotation time	200 msec	8.33 msec
Motor stop/start time	250 msec	20 sec
Time to transfer 1 sector	22 msec	17 μsec

Figure: Disk parameters for the original IBM PC floppy disk and a Western Digital WD 18300 hard disk.

- Average seek time is approx 12 times better.

Disk Structure IV

- Evolution of Disk Hardware (see Fig. 5)

Parameter	IBM 360-KB floppy disk	WD 18300 hard disk
Number of cylinders	40	10601
Tracks per cylinder	2	12
Sectors per track	9	281 (avg)
Sectors per disk	720	35742000
Bytes per sector	512	512
Disk capacity	360 KB	18.3 GB
Seek time (adjacent cylinders)	6 msec	0.8 msec
Seek time (average case)	77 msec	6.9 msec
Rotation time	200 msec	8.33 msec
Motor stop/start time	250 msec	20 sec
Time to transfer 1 sector	22 msec	17 μsec

Figure: Disk parameters for the original IBM PC floppy disk and a Western Digital WD 18300 hard disk.

- Average seek time is approx 12 times better.
- Rotation time is 24 times faster.

Disk Structure IV

- Evolution of Disk Hardware (see Fig. 5)

Parameter	IBM 360-KB floppy disk	WD 18300 hard disk
Number of cylinders	40	10601
Tracks per cylinder	2	12
Sectors per track	9	281 (avg)
Sectors per disk	720	35742000
Bytes per sector	512	512
Disk capacity	360 KB	18.3 GB
Seek time (adjacent cylinders)	6 msec	0.8 msec
Seek time (average case)	77 msec	6.9 msec
Rotation time	200 msec	8.33 msec
Motor stop/start time	250 msec	20 sec
Time to transfer 1 sector	22 msec	17 μsec

Figure: Disk parameters for the original IBM PC floppy disk and a Western Digital WD 18300 hard disk.

- Average seek time is approx 12 times better.
- Rotation time is 24 times faster.
- Transfer time is 1300 times faster.

Disk Structure IV

- Evolution of Disk Hardware (see Fig. 5)

Parameter	IBM 360-KB floppy disk	WD 18300 hard disk
Number of cylinders	40	10601
Tracks per cylinder	2	12
Sectors per track	9	281 (avg)
Sectors per disk	720	35742000
Bytes per sector	512	512
Disk capacity	360 KB	18.3 GB
Seek time (adjacent cylinders)	6 msec	0.8 msec
Seek time (average case)	77 msec	6.9 msec
Rotation time	200 msec	8.33 msec
Motor stop/start time	250 msec	20 sec
Time to transfer 1 sector	22 msec	17 μsec

Figure: Disk parameters for the original IBM PC floppy disk and a Western Digital WD 18300 hard disk.

- Average seek time is approx 12 times better.
- Rotation time is 24 times faster.
- Transfer time is 1300 times faster.
- Most of this gain is due to increase in density.

Disk Attachment

Computers access disk storage in two ways.

1. One way is via I/O ports (or host-attached storage); this is common on small systems.

Disk Attachment

Computers access disk storage in two ways.

1. One way is via I/O ports (or host-attached storage); this is common on small systems.
2. The other way is via a remote host in a distributed file system; this is referred to as network-attached storage.

Host-Attached Storage I

- Host-attached storage is storage accessed through local I/O ports.

Host-Attached Storage I

- Host-attached storage is storage accessed through local I/O ports.
- These ports use several technologies.

Host-Attached Storage I

- Host-attached storage is storage accessed through local I/O ports.
- These ports use several technologies.
    - The typical desktop PC uses an I/O bus architecture called IDE or ATA.

Host-Attached Storage I

- Host-attached storage is storage accessed through local I/O ports.
- These ports use several technologies.
  - The typical desktop PC uses an I/O bus architecture called IDE or ATA.
  - This architecture supports a maximum of two drives per I/O bus.

Host-Attached Storage I

- Host-attached storage is storage accessed through local I/O ports.
- These ports use several technologies.
  - The typical desktop PC uses an I/O bus architecture called IDE or ATA.
  - This architecture supports a maximum of two drives per I/O bus.
  - A newer, similar protocol that has simplified cabling is SATA.

Host-Attached Storage I

- Host-attached storage is storage accessed through local I/O ports.
- These ports use several technologies.
  - The typical desktop PC uses an I/O bus architecture called IDE or ATA.
  - This architecture supports a maximum of two drives per I/O bus.
  - A newer, similar protocol that has simplified cabling is SATA.
  - High-end workstations and servers generally use more sophisticated I/O architectures, such as SCSI and fiber channel (FC).

Host-Attached Storage II

- **SCSI** is a bus architecture;

Host-Attached Storage II

- **SCSI** is a bus architecture;
    - The SCSI protocol supports a maximum of 16 devices on the bus. Generally, the devices include one controller card in the host (the SCSI initiator) and up to 15 storage devices (the SCSI targets).

Host-Attached Storage II

- **SCSI** is a bus architecture;
  - The SCSI protocol supports a maximum of 16 devices on the bus. Generally, the devices include one controller card in the host (the SCSI initiator) and up to 15 storage devices (the SCSI targets).
  - A SCSI disk is a common SCSI target, but the protocol provides the ability to address up to 8 logical units in each SCSI target.

Host-Attached Storage II

- **SCSI** is a bus architecture;
  - The SCSI protocol supports a maximum of 16 devices on the bus. Generally, the devices include one controller card in the host (the SCSI initiator) and up to 15 storage devices (the SCSI targets).
  - A SCSI disk is a common SCSI target, but the protocol provides the ability to address up to 8 logical units in each SCSI target.
  - A typical use of logical unit addressing is to direct commands to components of a RAID array.

Host-Attached Storage II

- **SCSI** is a bus architecture;
  - The SCSI protocol supports a maximum of 16 devices on the bus. Generally, the devices include one controller card in the host (the SCSI initiator) and up to 15 storage devices (the SCSI targets).
  - A SCSI disk is a common SCSI target, but the protocol provides the ability to address up to 8 logical units in each SCSI target.
  - A typical use of logical unit addressing is to direct commands to components of a RAID array.
- **FC** is a high-speed serial architecture that can operate over optical fiber or over a four-conductor copper cable.

Host-Attached Storage II

- **SCSI** is a bus architecture;
  - The SCSI protocol supports a maximum of 16 devices on the bus. Generally, the devices include one controller card in the host (the SCSI initiator) and up to 15 storage devices (the SCSI targets).
  - A SCSI disk is a common SCSI target, but the protocol provides the ability to address up to 8 logical units in each SCSI target.
  - A typical use of logical unit addressing is to direct commands to components of a RAID array.
- **FC** is a high-speed serial architecture that can operate over optical fiber or over a four-conductor copper cable.
- A wide variety of storage devices are suitable for use as host-attached storage.

Host-Attached Storage II

- **SCSI** is a bus architecture;
  - The SCSI protocol supports a maximum of 16 devices on the bus. Generally, the devices include one controller card in the host (the SCSI initiator) and up to 15 storage devices (the SCSI targets).
  - A SCSI disk is a common SCSI target, but the protocol provides the ability to address up to 8 logical units in each SCSI target.
  - A typical use of logical unit addressing is to direct commands to components of a RAID array.
- **FC** is a high-speed serial architecture that can operate over optical fiber or over a four-conductor copper cable.
- A wide variety of storage devices are suitable for use as host-attached storage.
- The I/O commands that initiate data transfers to a host-attached storage device are reads and writes of logical data blocks directed to specifically identified storage units (such as bus ID, SCSI ID, and target logical unit).

Network-Attached Storage I

- A network-attached storage (NAS) device is a special-purpose storage system that is accessed remotely over a data network (see Fig. 6).

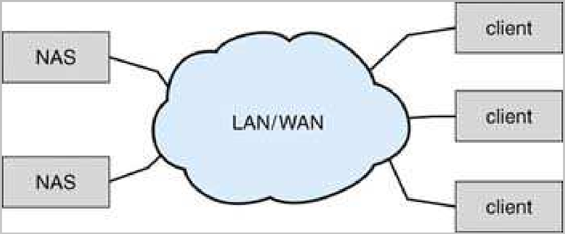


Figure: Network-attached storage.

Network-Attached Storage I

- A network-attached storage (NAS) device is a special-purpose storage system that is accessed remotely over a data network (see Fig. 6).

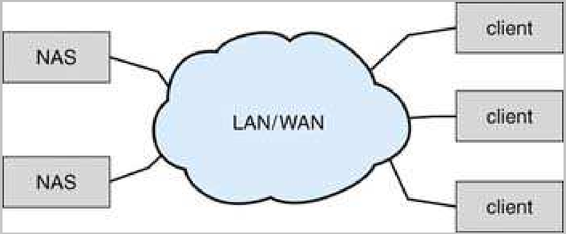


Figure: Network-attached storage.

- Clients access network-attached storage via a remote-procedure-call interface such as NFS for UNIX systems or CIFS for Windows machines.

Network-Attached Storage I

- A network-attached storage (NAS) device is a special-purpose storage system that is accessed remotely over a data network (see Fig. 6).

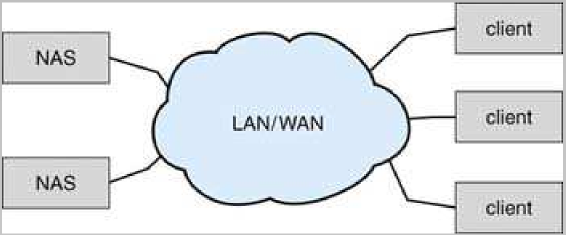


Figure: Network-attached storage.

- Clients access network-attached storage via a remote-procedure-call interface such as NFS for UNIX systems or CIFS for Windows machines.
- The remote procedure calls (RPCs) are carried via TCP or UDP over an IP network -usually the same local-area network (LAN) that carries all data traffic to the clients.

Dr. Cem Özdoğan

Network-Attached Storage II

- Network-attached storage provides a convenient way for all the computers on a LAN to share a pool of storage with the same ease of naming and access enjoyed with local host-attached storage.

Network-Attached Storage II

- Network-attached storage provides a convenient way for all the computers on a LAN to share a pool of storage with the same ease of naming and access enjoyed with local host-attached storage.
- However, it tends to be less efficient and have lower performance than some direct-attached storage options.

Network-Attached Storage II

- Network-attached storage provides a convenient way for all the computers on a LAN to share a pool of storage with the same ease of naming and access enjoyed with local host-attached storage.
- However, it tends to be less efficient and have lower performance than some direct-attached storage options.
- ISCSI is the latest network-attached storage protocol.

Network-Attached Storage II

- Network-attached storage provides a convenient way for all the computers on a LAN to share a pool of storage with the same ease of naming and access enjoyed with local host-attached storage.
- However, it tends to be less efficient and have lower performance than some direct-attached storage options.
- ISCSI is the latest network-attached storage protocol.
- In essence, it uses the IP network protocol to carry the SCSI protocol.

Network-Attached Storage II

- Network-attached storage provides a convenient way for all the computers on a LAN to share a pool of storage with the same ease of naming and access enjoyed with local host-attached storage.
- However, it tends to be less efficient and have lower performance than some direct-attached storage options.
- ISCSI is the latest network-attached storage protocol.
- In essence, it uses the IP network protocol to carry the SCSI protocol.
- Thus, networks rather than SCSI cables can be used as the interconnects between hosts and their storage.

Storage-Area Network I

- One drawback of network-attached storage systems is that the storage I/O operations consume bandwidth on the data network, thereby increasing the latency of network communication.

Storage-Area Network I

- One drawback of network-attached storage systems is that the storage I/O operations consume bandwidth on the data network, thereby increasing the latency of network communication.
- A storage-area network (SAN) is a private network (using storage protocols rather than networking protocols) connecting servers and storage units, as shown in Fig. 7.

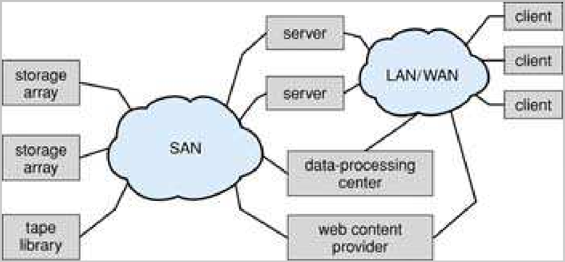


Figure: Storage-area network.

Storage-Area Network II

- The power of a SAN lies in its flexibility.

Storage-Area Network II

- The power of a SAN lies in its flexibility.
- Multiple hosts and multiple storage arrays can attach to the same SAN, and storage can be dynamically allocated to hosts.

Storage-Area Network II

- The power of a SAN lies in its flexibility.
- Multiple hosts and multiple storage arrays can attach to the same SAN, and storage can be dynamically allocated to hosts.
- SANs typically have more ports, and less expensive ports, than storage arrays.

Storage-Area Network II

- The power of a SAN lies in its flexibility.
- Multiple hosts and multiple storage arrays can attach to the same SAN, and storage can be dynamically allocated to hosts.
- SANs typically have more ports, and less expensive ports, than storage arrays.
- FC is the most common SAN interconnect.

Storage-Area Network II

- The power of a SAN lies in its flexibility.
- Multiple hosts and multiple storage arrays can attach to the same SAN, and storage can be dynamically allocated to hosts.
- SANs typically have more ports, and less expensive ports, than storage arrays.
- FC is the most common SAN interconnect.
- An emerging alternative is a special-purpose bus architecture named InfiniBand,

Storage-Area Network II

- The power of a SAN lies in its flexibility.
- Multiple hosts and multiple storage arrays can attach to the same SAN, and storage can be dynamically allocated to hosts.
- SANs typically have more ports, and less expensive ports, than storage arrays.
- FC is the most common SAN interconnect.
- An emerging alternative is a special-purpose bus architecture named InfiniBand,
- which provides hardware and software support for high-speed interconnection networks for servers and storage units.

Disk Scheduling

- For the disk drives: having fast access time and large disk bandwidth.

Disk Scheduling

- For the disk drives: having fast access time and large disk bandwidth.
- The access time has two major components.

Disk Scheduling

- For the disk drives: having fast access time and large disk bandwidth.
- The access time has two major components.
  - The **seek time**.

Disk Scheduling

- For the disk drives: having fast access time and large disk bandwidth.
- The access time has two major components.
  - The **seek time**.
  - The **rotational latency**.

Disk Scheduling

- For the disk drives: having fast access time and large disk bandwidth.
- The access time has two major components.
  - The **seek time**.
  - The **rotational latency**.
- 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.

Disk Scheduling

- For the disk drives: having fast access time and large disk bandwidth.
- The access time has two major components.
  - The **seek time**.
  - The **rotational latency**.
- 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 scheduling the servicing of disk I/O requests in a **good order**.

Disk Scheduling

- For the disk drives: having fast access time and large disk bandwidth.
- The access time has two major components.
  - The **seek time**.
  - The **rotational latency**.
- 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 scheduling the servicing of disk I/O requests in a **good order**.
- For a multiprogramming system with many processes, the disk queue may often have several pending requests.

Disk Scheduling

- For the disk drives: having fast access time and large disk bandwidth.
- The access time has two major components.
  - The **seek time**.
  - The **rotational latency**.
- 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 scheduling the servicing of disk I/O requests in a **good order**.
- For a multiprogramming system with many processes, the disk queue may often have several pending requests.
  - Thus, when one request is completed, the OS chooses which pending request to service next.

Disk Scheduling

- For the disk drives: having fast access time and large disk bandwidth.
- The access time has two major components.
  - The **seek time**.
  - The **rotational latency**.
- 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 scheduling the servicing of disk I/O requests in a **good order**.
- For a multiprogramming system with many processes, the disk queue may often have several pending requests.
  - Thus, when one request is completed, the OS chooses which pending request to service next.
  - How does the OS make this choice? **Disk-scheduling algorithms**.

FCFS Scheduling I

- The simplest form of disk scheduling is the first-come, first-served (FCFS) algorithm.

FCFS Scheduling II

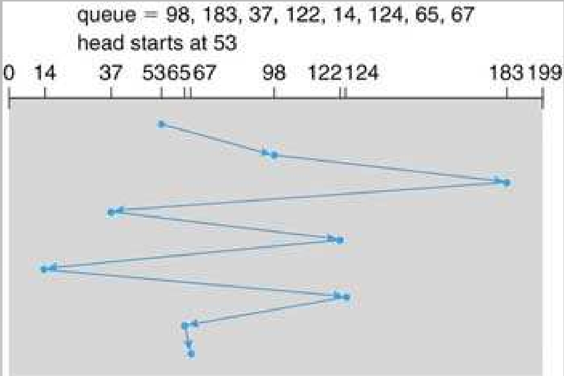


Figure: FCFS disk scheduling.

- The wild swing from 122 to 14 and then back to 124 illustrates the problem with this schedule.

FCFS Scheduling II

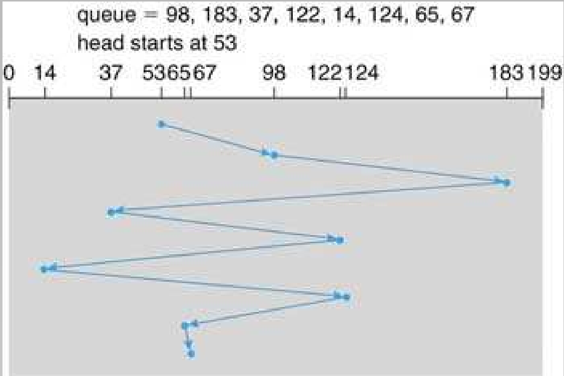


Figure: FCFS disk scheduling.

- The wild swing from 122 to 14 and then back to 124 illustrates the problem with this schedule.
- If the requests for cylinders 37 and 14 could be serviced together, before or after the requests at 122 and 124, the total head movement could be decreased substantially.

SSTF Scheduling I

- It seems reasonable to service all the requests close to the current head position before moving the head far away to service other requests.

SSTF Scheduling II

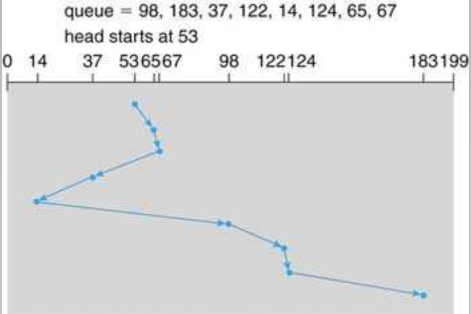


Figure: SSTF disk scheduling.

- This scheduling method results in a total head movement of only 236 cylinders-little more than one-third of the distance needed for FCFS scheduling of this request queue.

SSTF Scheduling II

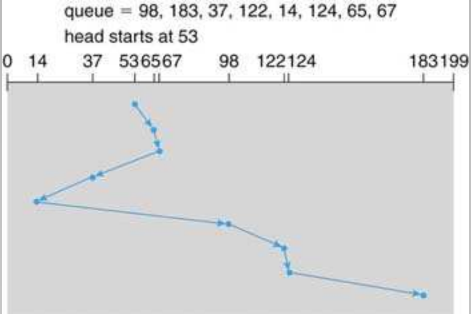


Figure: SSTF disk scheduling.

- This scheduling method results in a total head movement of only 236 cylinders-little more than one-third of the distance needed for FCFS scheduling of this request queue.
- This algorithm gives a substantial improvement in performance.

SSTF Scheduling III

- SSTF scheduling is essentially a form of shortest-job-first (SJF) scheduling;

SCAN Scheduling I

- In the SCAN algorithm, 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.

SCAN Scheduling II

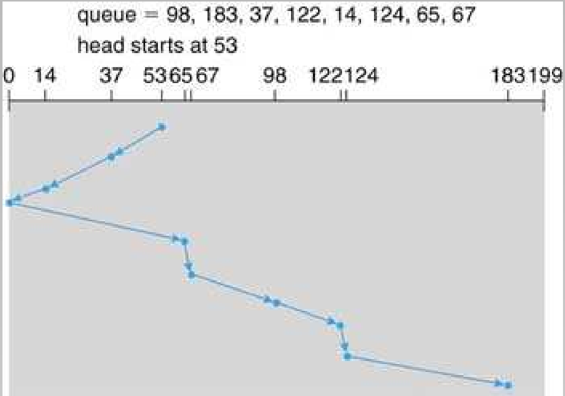


Figure: SCAN disk scheduling.

- If a request arrives in the queue just in front of the head, it will be serviced almost immediately

SCAN Scheduling II

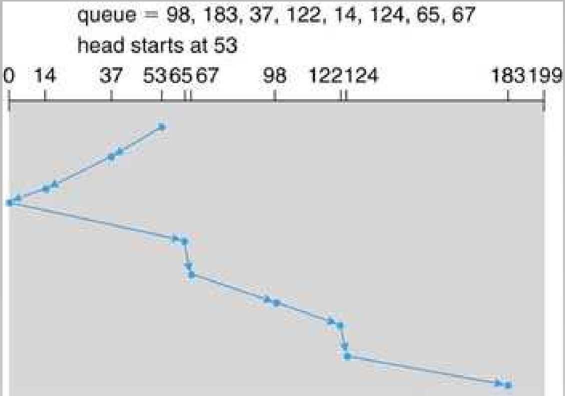


Figure: SCAN disk scheduling.

- If a request arrives in the queue just in front of the head, it will be serviced almost immediately
- If a request arriving just behind the head will have to wait until the arm moves to the end of the disk, reverses direction, and comes back.

SCAN Scheduling III

- Assuming a uniform distribution of requests for cylinders, consider the density of requests when the head reaches one end and reverses direction.

SCAN Scheduling III

- Assuming a uniform distribution of requests for cylinders, consider the density of requests when the head reaches one end and reverses direction.
  - At this point, relatively few requests are immediately in front of the head, since these cylinders have recently been serviced.

SCAN Scheduling III

- Assuming a uniform distribution of requests for cylinders, consider the density of requests when the head reaches one end and reverses direction.
  - At this point, relatively few requests are immediately in front of the head, since these cylinders have recently been serviced.
  - The heaviest density of requests is at the other end of the disk.

SCAN Scheduling III

- Assuming a uniform distribution of requests for cylinders, consider the density of requests when the head reaches one end and reverses direction.
    - At this point, relatively few requests are immediately in front of the head, since these cylinders have recently been serviced.
    - The heaviest density of requests is at the other end of the disk.
    - These requests have also waited the longest, so why not go there first?

SCAN Scheduling III

- Assuming a uniform distribution of requests for cylinders, consider the density of requests when the head reaches one end and reverses direction.
  - At this point, relatively few requests are immediately in front of the head, since these cylinders have recently been serviced.
  - The heaviest density of requests is at the other end of the disk.
  - These requests have also waited the longest, so why not go there first?
  - That is the idea of the next algorithm.

C-SCAN Scheduling

- Circular SCAN (C-SCAN) scheduling is a variant of SCAN.

C-SCAN Scheduling

- Circular SCAN (C-SCAN) scheduling is a variant of SCAN.
- Like SCAN, CSCAN moves the head from one end of the disk to the other, servicing requests along the way.

C-SCAN Scheduling

- Circular SCAN (C-SCAN) scheduling is a variant of SCAN.
- Like SCAN, CSCAN 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 (see Fig. 11).

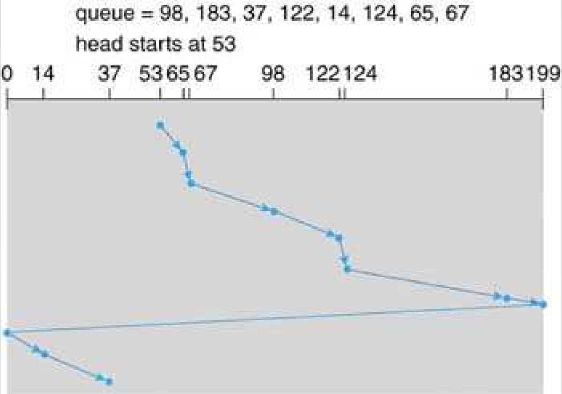


Figure: C-SCAN disk scheduling.

LOOK Scheduling

- The arm goes only as far as the final request in each direction.

LOOK Scheduling

- 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.

Dr. Cem Özdoğan

LOOK Scheduling

- 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 (see Fig. 12).

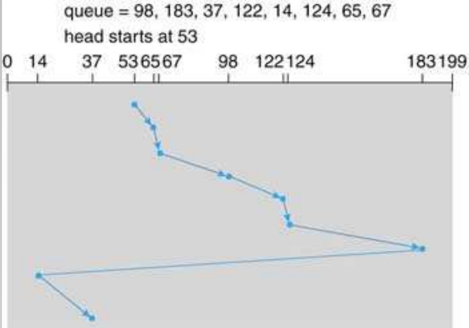


Figure: C-LOOK disk scheduling.

Disk Formatting I

- A new magnetic disk is a blank slate: It is just a platter of a magnetic recording material.

Disk Formatting I

- A new magnetic disk is a blank slate: It is just a platter of a magnetic recording material.
- Before a disk can store data, it must be divided into sectors that the disk controller can read and write.

Disk Formatting I

- A new magnetic disk is a blank slate: It is just a platter of a magnetic recording material.
- Before a disk can store data, it must be divided into sectors that the disk controller can read and write.
- This process is called **low-level formatting**, or **physical formatting**.

Disk Formatting I

- A new magnetic disk is a blank slate: It is just a platter of a magnetic recording material.
- Before a disk can store data, it must be divided into sectors that the disk controller can read and write.
- This process is called **low-level formatting**, or **physical formatting**.
- Low-level formatting fills the disk with a special data structure for each sector.

Disk Formatting I

- A new magnetic disk is a blank slate: It is just a platter of a magnetic recording material.
- Before a disk can store data, it must be divided into sectors that the disk controller can read and write.
- This process is called **low-level formatting**, or **physical formatting**.
- Low-level formatting fills the disk with a special data structure for each sector.
  - The data structure for a sector typically consists of a header, a data area (usually 512 bytes in size), and a trailer.

Disk Formatting I

- A new magnetic disk is a blank slate: It is just a platter of a magnetic recording material.
- Before a disk can store data, it must be divided into sectors that the disk controller can read and write.
- This process is called **low-level formatting**, or **physical formatting**.
- Low-level formatting fills the disk with a special data structure for each sector.
  - The data structure for a sector typically consists of a header, a data area (usually 512 bytes in size), and a trailer.
  - The header and trailer contain information used by the disk controller, such as a sector number and an **error-correcting code** (ECC).

Disk Formatting I

- A new magnetic disk is a blank slate: It is just a platter of a magnetic recording material.
- Before a disk can store data, it must be divided into sectors that the disk controller can read and write.
- This process is called **low-level formatting**, or **physical formatting**.
- Low-level formatting fills the disk with a special data structure for each sector.
  - The data structure for a sector typically consists of a header, a data area (usually 512 bytes in size), and a trailer.
  - The header and trailer contain information used by the disk controller, such as a sector number and an **error-correcting code** (ECC).
- When the controller **writes** a sector of data during normal I/O, the ECC is updated with a value calculated from all the bytes in the data area.

Disk Formatting I

- A new magnetic disk is a blank slate: It is just a platter of a magnetic recording material.
- Before a disk can store data, it must be divided into sectors that the disk controller can read and write.
- This process is called **low-level formatting**, or **physical formatting**.
- Low-level formatting fills the disk with a special data structure for each sector.
  - The data structure for a sector typically consists of a header, a data area (usually 512 bytes in size), and a trailer.
  - The header and trailer contain information used by the disk controller, such as a sector number and an **error-correcting code** (ECC).
- When the controller **writes** a sector of data during normal I/O, the ECC is updated with a value calculated from all the bytes in the data area.
- When the sector is **read**, the ECC is recalculated and is compared with the stored value.

Disk Formatting I

- A new magnetic disk is a blank slate: It is just a platter of a magnetic recording material.
- Before a disk can store data, it must be divided into sectors that the disk controller can read and write.
- This process is called **low-level formatting**, or **physical formatting**.
- Low-level formatting fills the disk with a special data structure for each sector.
  - The data structure for a sector typically consists of a header, a data area (usually 512 bytes in size), and a trailer.
  - The header and trailer contain information used by the disk controller, such as a sector number and an **error-correcting code** (ECC).
- When the controller **writes** a sector of data during normal I/O, the ECC is updated with a value calculated from all the bytes in the data area.
- When the sector is **read**, the ECC is recalculated and is compared with the stored value.
- The controller automatically does the ECC processing whenever a sector is read or written.

Disk Formatting II

- To use a disk to hold files, the OS still needs to record its own data structures on the disk. It does so in two steps.

Disk Formatting II

- To use a disk to hold files, the OS still needs to record its own data structures on the disk. It does so in two steps.

1. The first step is to partition the disk into one or more groups of cylinders.

Disk Formatting II

- To use a disk to hold files, the OS still needs to record its own data structures on the disk. It does so in two steps.

1. The first step is to partition the disk into one or more groups of cylinders.
    - The OS can treat each partition as though it were a separate disk.

Disk Formatting II

- To use a disk to hold files, the OS still needs to record its own data structures on the disk. It does so in two steps.

1. The first step is to partition the disk into one or more groups of cylinders.
    - The OS can treat each partition as though it were a separate disk.
    - For instance, one partition can hold a copy of the OS's executable code, while another holds user files.

Disk Formatting II

- To use a disk to hold files, the OS still needs to record its <u>own data</u> <u>structures</u> on the disk. It does so in two steps.

1. The first step is to partition the disk into one or more groups of cylinders.
   - The OS can treat each partition as though it were a separate disk.
   - For instance, one partition can hold a copy of the OS's executable code, while another holds user files.
2. After partitioning, the second step is **logical formatting** (or creation of a file system).

Disk Formatting II

- To use a disk to hold files, the OS still needs to record its <u>own data</u> <u>structures</u> on the disk. It does so in two steps.

1. The first step is to partition the disk into one or more groups of cylinders.
   - The OS can treat each partition as though it were a separate disk.
   - For instance, one partition can hold a copy of the OS's executable code, while another holds user files.
2. After partitioning, the second step is **logical formatting** (or creation of a file system).
   - In this step, the OS stores the initial file-system data structures onto the disk.

Disk Formatting II

- To use a disk to hold files, the OS still needs to record its own data structures on the disk. It does so in two steps.

1. The first step is to partition the disk into one or more groups of cylinders.
   - The OS can treat each partition as though it were a separate disk.
   - For instance, one partition can hold a copy of the OS's executable code, while another holds user files.
2. After partitioning, the second step is **logical formatting** (or creation of a file system).
   - In this step, the OS stores the initial file-system data structures onto the disk.
   - These data structures may include maps of free and allocated space (a FAT or inodes) and an initial empty directory.

Disk Formatting III

- When reading sequential blocks, the seek time can result in missing block 0 in the next **track**.

Disk Formatting III

- When reading sequential blocks, the seek time can result in missing block 0 in the next **track**.
- Disk can be formatted using a cylinder *skew* to avoid this (see Fig. 13).

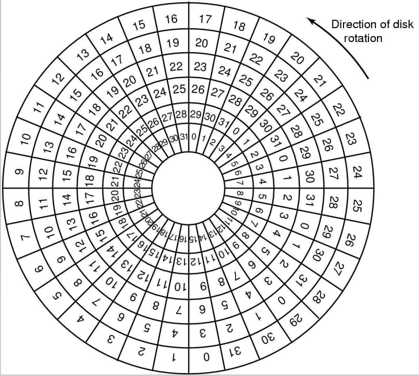


Figure: An illustration of cylinder skew.

RAID Structure

- A variety of disk-organization techniques, collectively called **redundant arrays of inexpensive disks** (**RAIDs**) are commonly used to address the performance and reliability issues.

RAID Structure

- A variety of disk-organization techniques, collectively called **redundant arrays of inexpensive disks (RAIDs)** are commonly used to address the performance and reliability issues.
- In the past, RAIDs composed of small, cheap disks were viewed as a cost-effective alternative to large, expensive disks.

RAID Structure

- A variety of disk-organization techniques, collectively called **redundant arrays of inexpensive disks (RAIDs)** are commonly used to address the performance and reliability issues.
- In the past, RAIDs composed of small, cheap disks were viewed as a cost-effective alternative to large, expensive disks.
- Today, RAIDs are used for their higher reliability and higher data-transfer rate, rather than for economic reasons.

RAID Structure

- A variety of disk-organization techniques, collectively called **redundant arrays of inexpensive disks** (**RAIDs**) are commonly used to address the performance and reliability issues.
- In the past, RAIDs composed of small, cheap disks were viewed as a cost-effective alternative to large, expensive disks.
- Today, RAIDs are used for their higher reliability and higher data-transfer rate, rather than for economic reasons.
- Hence, the *I* in RAID now stands for "independent" instead of "inexpensive".

Improvement of Reliability via Redundancy

- If we store only one copy of the data, then each disk failure will result in loss of a significant amount of data-and such a high rate of data loss is unacceptable.

Improvement of Reliability via Redundancy

- If we store only one copy of the data, then each disk failure will result in loss of a significant amount of data-and such a high rate of data loss is unacceptable.
- The solution to the problem of reliability is to introduce redundancy;

Improvement of Reliability via Redundancy

- If we store only one copy of the data, then each disk failure will result in loss of a significant amount of data-and such a high rate of data loss is unacceptable.
- The solution to the problem of reliability is to introduce redundancy;
- We store extra information that is not normally needed but that can be used in the event of failure of a disk to rebuild the lost information.

Improvement of Reliability via Redundancy

- If we store only one copy of the data, then each disk failure will result in loss of a significant amount of data-and such a high rate of data loss is unacceptable.
- The solution to the problem of reliability is to introduce redundancy;
- We store extra information that is not normally needed but that can be used in the event of failure of a disk to rebuild the lost information.
- Thus, even if a disk fails, data are not lost.

Improvement of Reliability via Redundancy

- If we store only one copy of the data, then each disk failure will result in loss of a significant amount of data-and such a high rate of data loss is unacceptable.
- The solution to the problem of reliability is to introduce redundancy;
- We store extra information that is not normally needed but that can be used in the event of failure of a disk to rebuild the lost information.
- Thus, even if a disk fails, data are not lost.
- The simplest (but most expensive) approach to introducing redundancy is to duplicate every disk.

Improvement of Reliability via Redundancy

- If we store only one copy of the data, then each disk failure will result in loss of a significant amount of data-and such a high rate of data loss is unacceptable.
- The solution to the problem of reliability is to introduce redundancy;
- We store extra information that is not normally needed but that can be used in the event of failure of a disk to rebuild the lost information.
- Thus, even if a disk fails, data are not lost.
- The simplest (but most expensive) approach to introducing redundancy is to duplicate every disk.
- This technique is called **mirroring**.

Improvement in Performance via Parallelism

- With multiple disks, we can improve the transfer rate as well (or instead) by striping data across the disks.

Improvement in Performance via Parallelism

- With multiple disks, we can improve the transfer rate as well (or instead) by striping data across the disks.
- In its simplest form, data striping consists of splitting the bits of each byte across multiple disks;

Improvement in Performance via Parallelism

- With multiple disks, we can improve the transfer rate as well (or instead) by striping data across the disks.
- In its simplest form, data striping consists of splitting the bits of each byte across multiple disks;
- Such striping is called bit-level striping.

Improvement in Performance via Parallelism

- With multiple disks, we can improve the transfer rate as well (or instead) by striping data across the disks.
- In its simplest form, data striping consists of splitting the bits of each byte across multiple disks;
- Such striping is called bit-level striping.
  - For example, if we have an array of eight disks, we write bit *i* of each byte to disk *i*.

Improvement in Performance via Parallelism

- With multiple disks, we can improve the transfer rate as well (or instead) by striping data across the disks.
- In its simplest form, data striping consists of splitting the bits of each byte across multiple disks;
- Such striping is called bit-level striping.
  - For example, if we have an array of eight disks, we write bit *i* of each byte to disk *i*.
- The array of eight disks can be treated as a single disk with sectors that are eight times the normal size,

Improvement in Performance via Parallelism

- With multiple disks, we can improve the transfer rate as well (or instead) by striping data across the disks.
- In its simplest form, data striping consists of splitting the bits of each byte across multiple disks;
- Such striping is called bit-level striping.
  - For example, if we have an array of eight disks, we write bit *i* of each byte to disk *i*.
- The array of eight disks can be treated as a single disk with sectors that are eight times the normal size,
- More important, that have eight times the access rate.

Improvement in Performance via Parallelism

- With multiple disks, we can improve the transfer rate as well (or instead) by striping data across the disks.
- In its simplest form, data striping consists of splitting the bits of each byte across multiple disks;
- Such striping is called bit-level striping.
  - For example, if we have an array of eight disks, we write bit *i* of each byte to disk *i*.
- The array of eight disks can be treated as a single disk with sectors that are eight times the normal size,
- More important, that have eight times the access rate.
- Parallelism in a disk system, as achieved through striping, has two main goals:

Improvement in Performance via Parallelism

- With multiple disks, we can improve the transfer rate as well (or instead) by striping data across the disks.
- In its simplest form, data striping consists of splitting the bits of each byte across multiple disks;
- Such striping is called bit-level striping.
  - For example, if we have an array of eight disks, we write bit *i* of each byte to disk *i*.
- The array of eight disks can be treated as a single disk with sectors that are eight times the normal size,
- More important, that have eight times the access rate.
- Parallelism in a disk system, as achieved through striping, has two main goals:
  1. Increase the throughput of multiple small accesses (that is, page accesses) by load balancing.

Improvement in Performance via Parallelism

- With multiple disks, we can improve the transfer rate as well (or instead) by striping data across the disks.
- In its simplest form, data striping consists of splitting the bits of each byte across multiple disks;
- Such striping is called bit-level striping.
  - For example, if we have an array of eight disks, we write bit *i* of each byte to disk *i*.
- The array of eight disks can be treated as a single disk with sectors that are eight times the normal size,
- More important, that have eight times the access rate.
- Parallelism in a disk system, as achieved through striping, has two main goals:
  1. Increase the throughput of multiple small accesses (that is, page accesses) by load balancing.
  2. Reduce the response time of large accesses.

RAID Levels I

- Mirroring provides high reliability, but it is expensive.

RAID Levels I

- Mirroring provides high reliability, but it is expensive.
- Striping provides high data-transfer rates, but it does not improve reliability.

RAID Levels I

- Mirroring provides high reliability, but it is expensive.
- Striping provides high data-transfer rates, but it does not improve reliability.
- Numerous schemes to provide redundancy at lower cost by using the idea of disk striping combined with “parity” bits have been proposed.

RAID Levels I

- Mirroring provides high reliability, but it is expensive.
- Striping provides high data-transfer rates, but it does not improve reliability.
- Numerous schemes to provide redundancy at lower cost by using the idea of disk striping combined with “parity” bits have been proposed.
- These schemes have different cost-performance trade-offs and are classified according to levels called **RAID levels** (see Figs. 14 & Figs. 15).

RAID Levels II

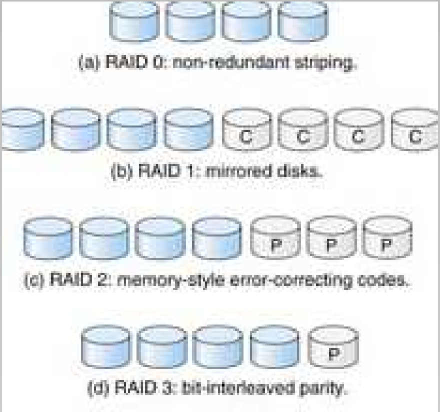


Figure: RAID levels 0-3.

RAID Levels III

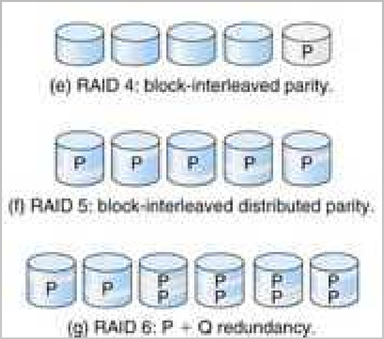


Figure: RAID levels 4-6.

In the figures, *P* indicates error-correcting bits, and *C* indicates a second copy of the data.
In all cases depicted in the figures, four disks' worth of data are stored, and the extra disks are used to store redundant information for failure recovery.