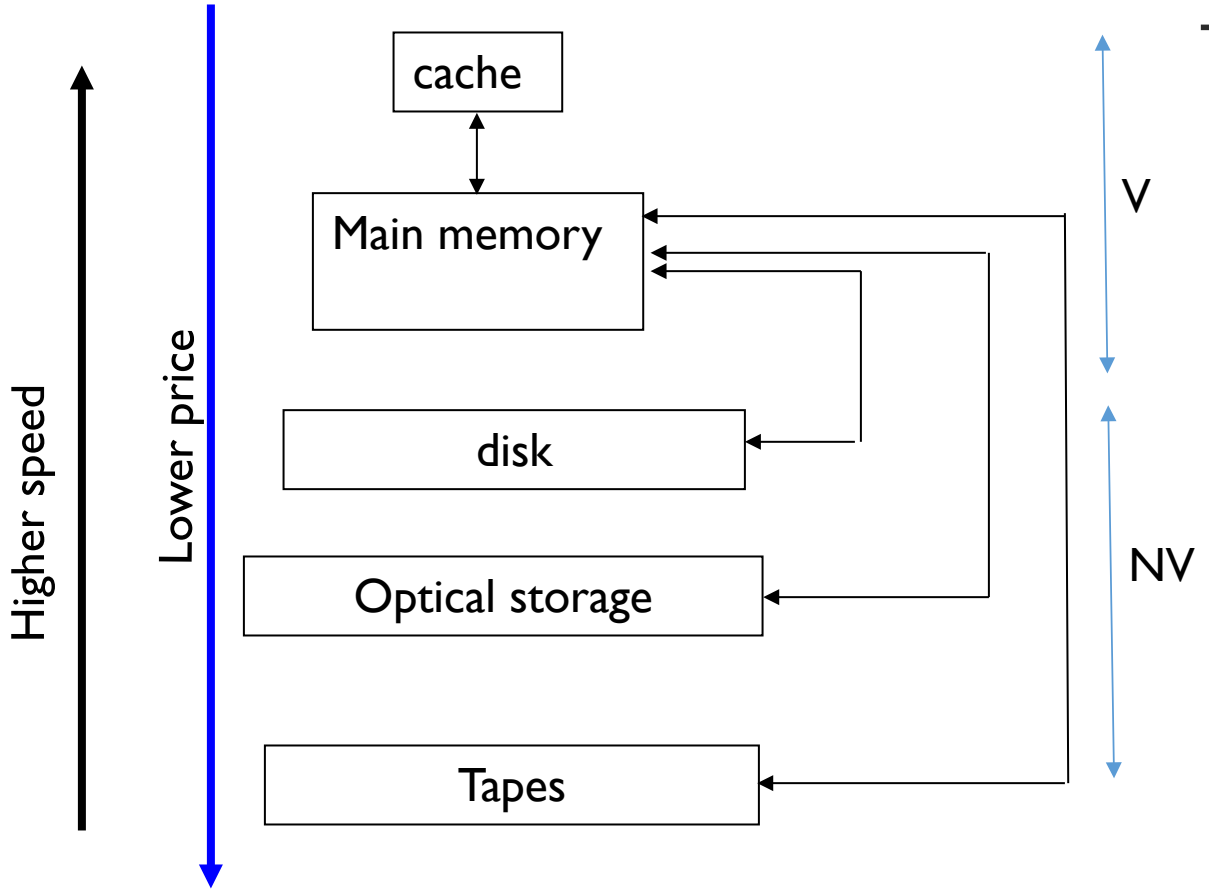
BBM371- Data Management

Lecture 2: Storage Devices

18.10.2018

Memory Hierarchy



Traveling the hierarchy:

1. speed (higher=faster)
2. cost (lower=cheaper)
3. volatility (between MM and Disk)
4. Data transfer (Main memory the "hub")
5. Storage classes (P=primary, S=secondary, T=tertiary)

Storage Devices

- Direct Access Storage Devices

  Hard disk: Huge vol., fast, expensive

  USB, CD, DVD: Transportable, less vol., slow, cheaper

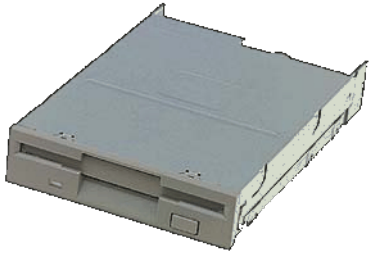
Hard Disk - random access storage

- Serial Access Storage Devices

  Tapes: Largest Size, Cheaper, Transportable, Slow

Tape Drive – sequential storage

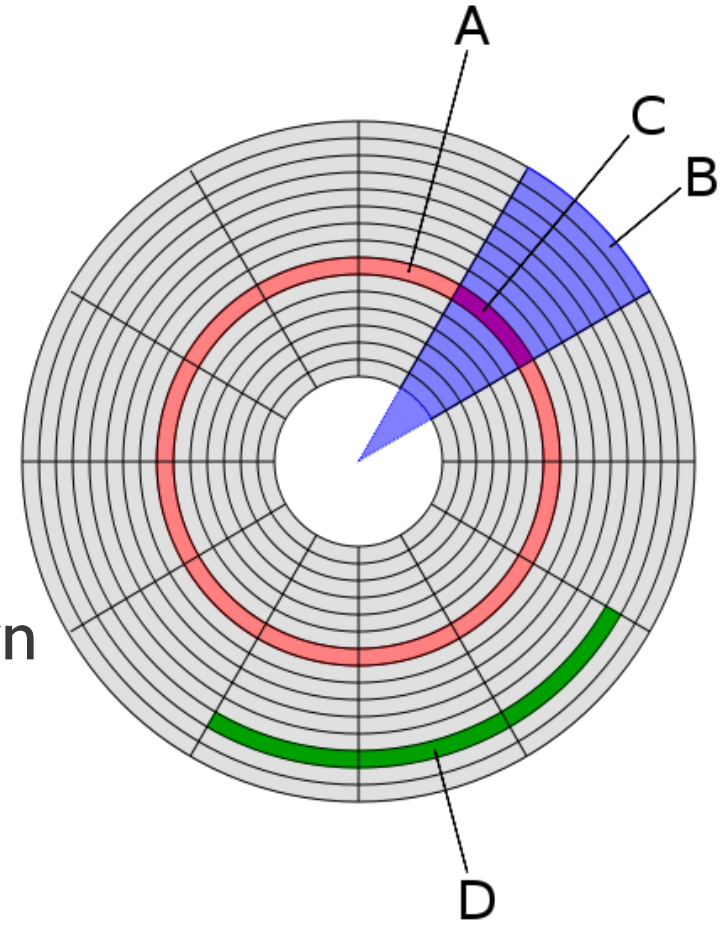
CD ROM / DVD



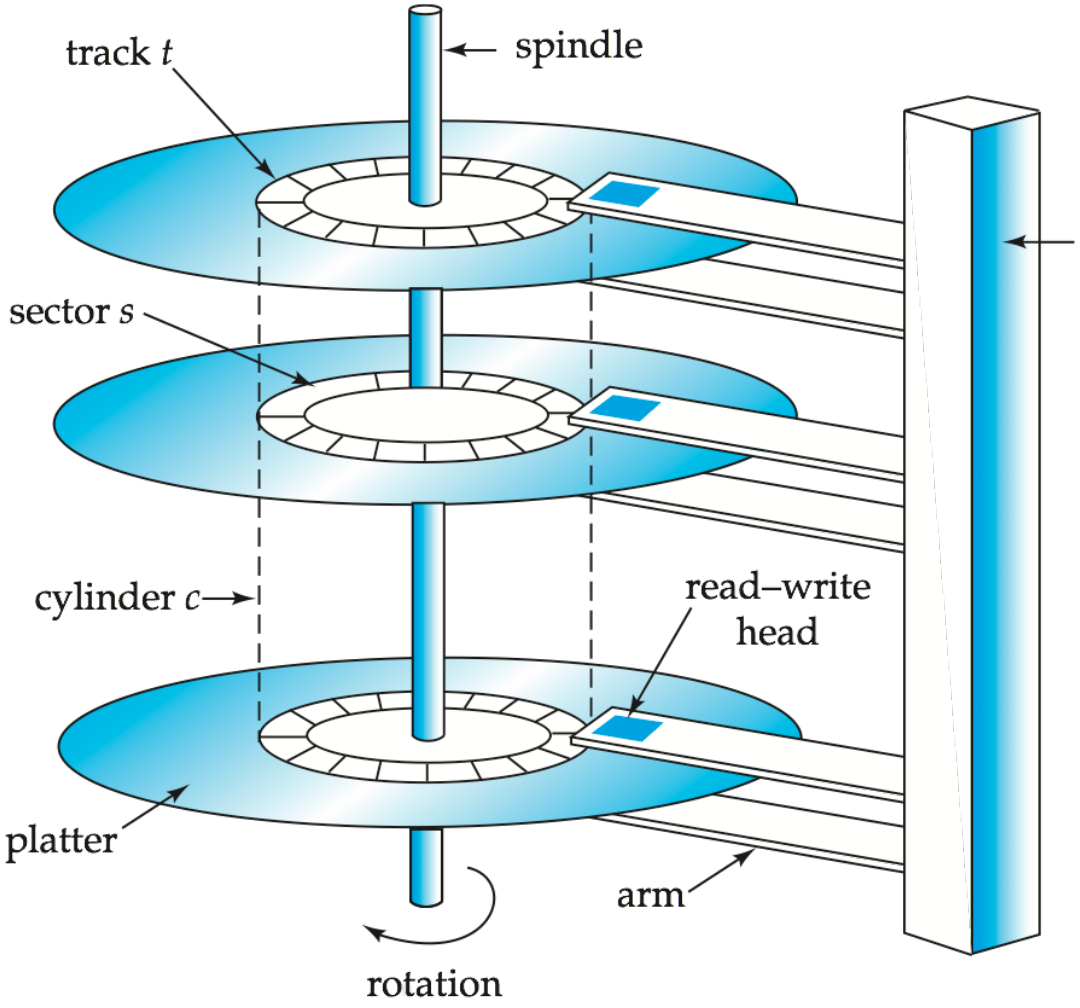
Floppy Disk Drive – random-access storage

Organization of Disk

- Platters : May use both sides for storing data
- Track : A circular path where the data is stored magnetically (Shown as A)
- Sector : A subdivision of track. (C)
  - Normally sector means portion of disk enclosed by two radii and arc. (B)
- Cluster : Unit of data transferred, defined in terms of multiple sectors (e.g. 3 sectors) (Shown as D)
  - This is usually a file system related setting



From Physical to Conceptual



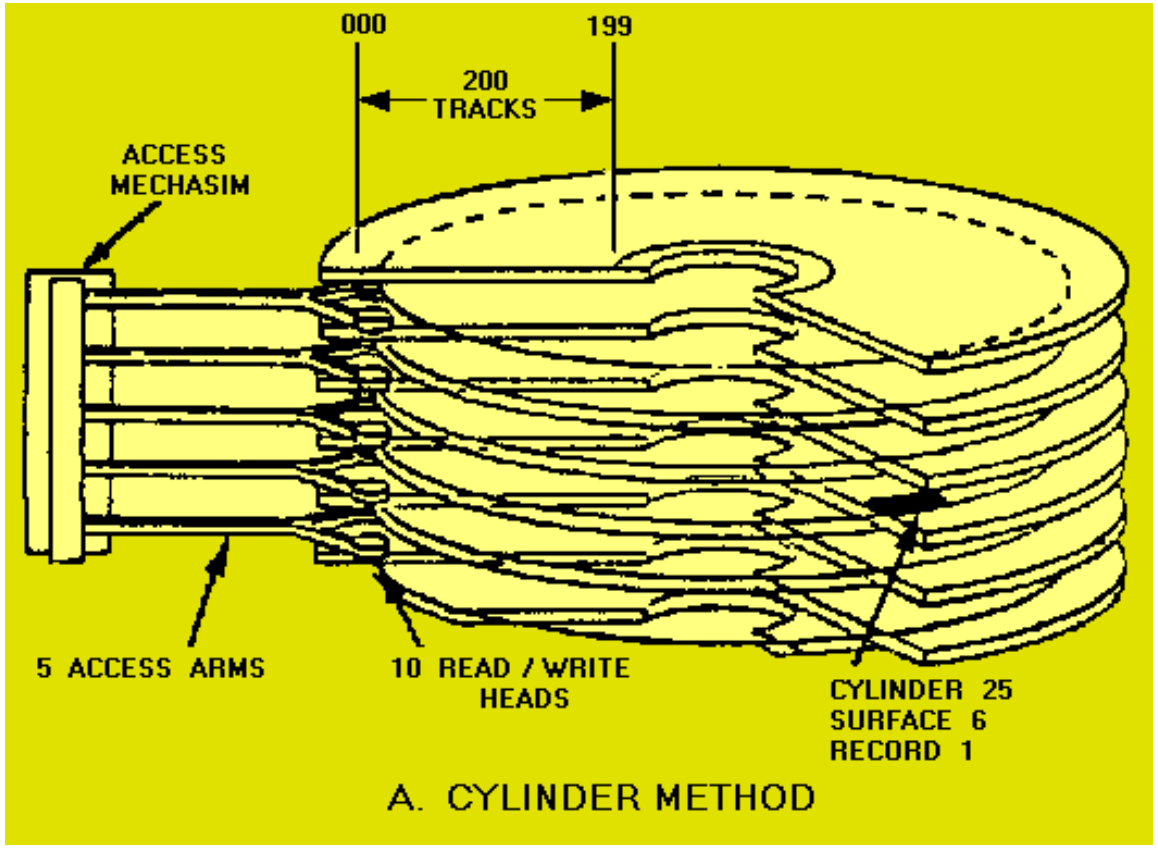
Sector: Minumum track partition for data storage

Cluster: Group of sectors

Track: One cyle on the platter

Cylinder: A semantic shape that consists of same level tracks on the platters

From Physical to Conceptual



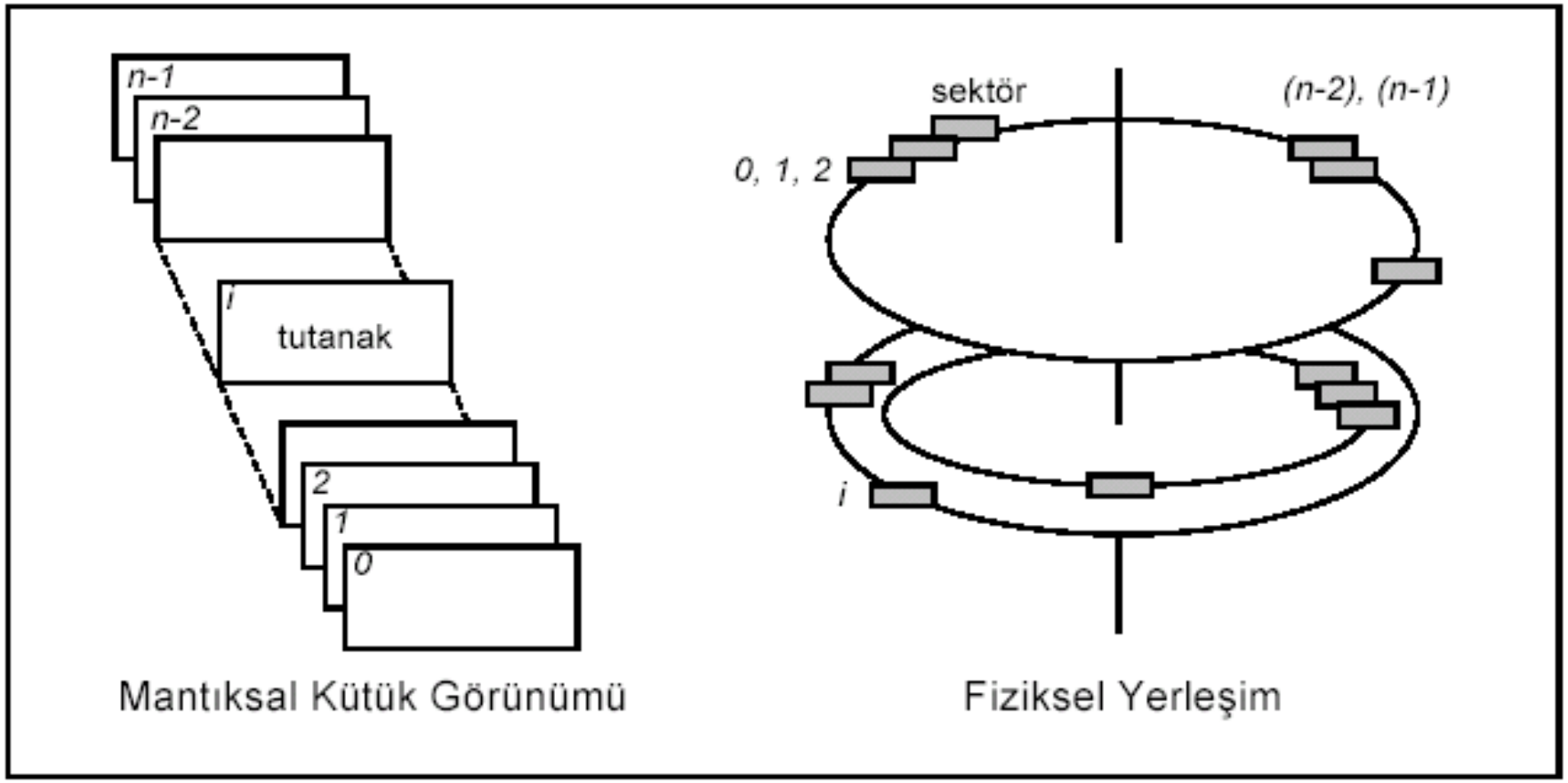
Sector: Minumum track partition for data storage

Cluster: Group of sectors

Track: One cyle on the platter

Cylinder: A semantic shape that consists of same level tracks on the platters

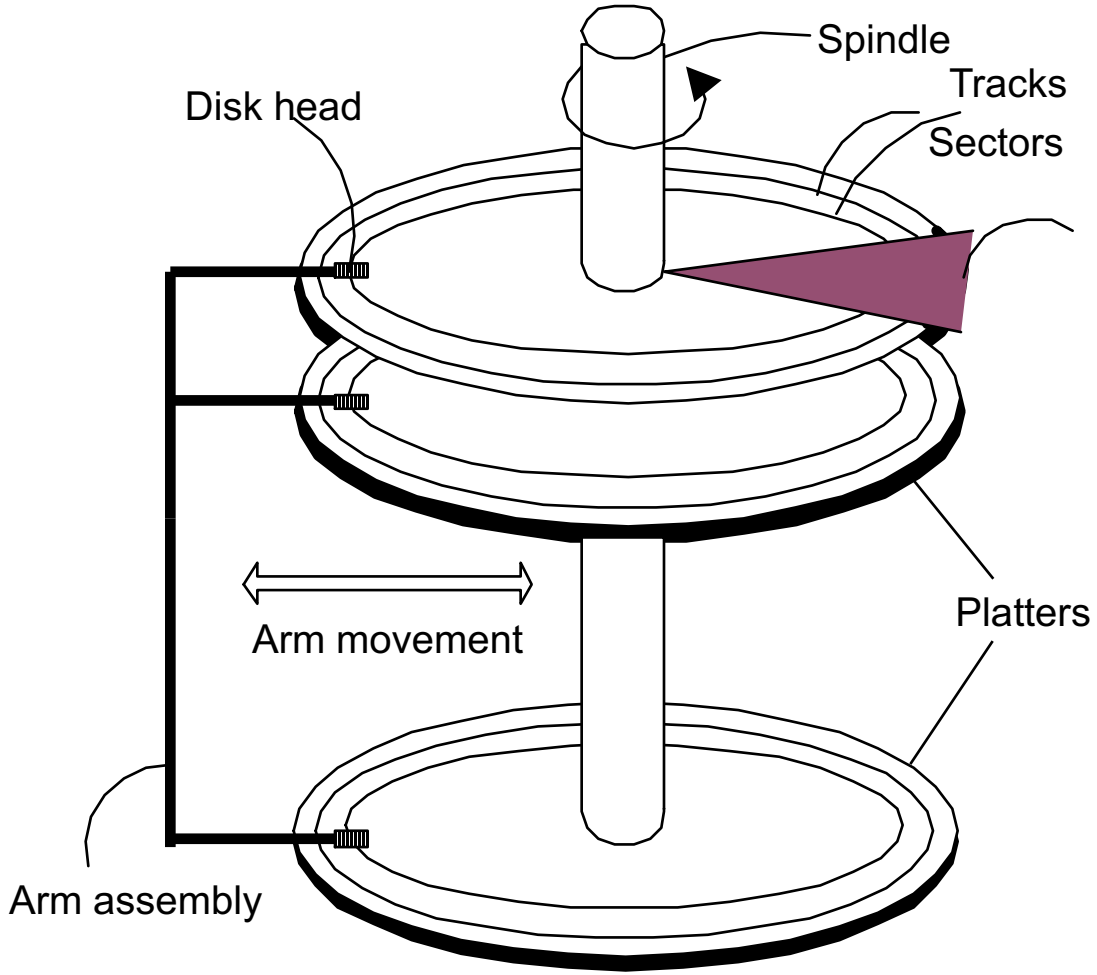
Relationship between Logical & Physical Models



Çizim 6.1. Kavramsal Kütük Modeli ve Fiziksel Yerleşimi

Components of a Disk

- The platters spin, to position for the desired sector
- The arm assembly is moved inwards or outwards to position the head on a desired track
- Only one head reads/writes at any one time
- Block size is a multiple of sector size (which is fixed)



Blocking

- Physical Block
  - Minimum data transferred between primary and secondary memory
- Logical Block
  - Minimum set of records transferred between primary and secondary memory
- Blocking Factor
  - The number of logical record per logical block
- Assume, one cluster=one block

Blocking (cont.)

- Entire physical blocks are moved from secondary to the continuous section of primary memory called as buffer
- A request for a disk block already in buffer does not require a disk transfer
- Speeding-up database operations depends on arranging data so that when a record in a block is needed, rest of the records on the same block will also be needed soon.
- Choosing too big block size will transfer too much unnecessary records, cluttering the buffer
- Choosing too small block size will guarantee a disk operation per each record request

Choosing Block Factor

- Not too large not too few
- Affected by
  - Storage media
  - Operating system
  - Record length
    - Fixed
    - Variable

Logical READ

Reading a record:

- The physical block(s) are transferred to the buffer (in primary memory)
  - Only perform physical disk transfer if not already in the buffer.
  - If it is in buffer this step is skipped (cache hit!)
- The record is extracted from the block
- Data in the record is transferred to variables of the program

Can you specify how a logical write is performed?

Access

- Access path : used to find a target record
  - Search mechanism
  - File organization
  - Secondary storage media

- Length of an access path:
  - The number of physical blocks accessed in the process of accessing a record

File design considerations

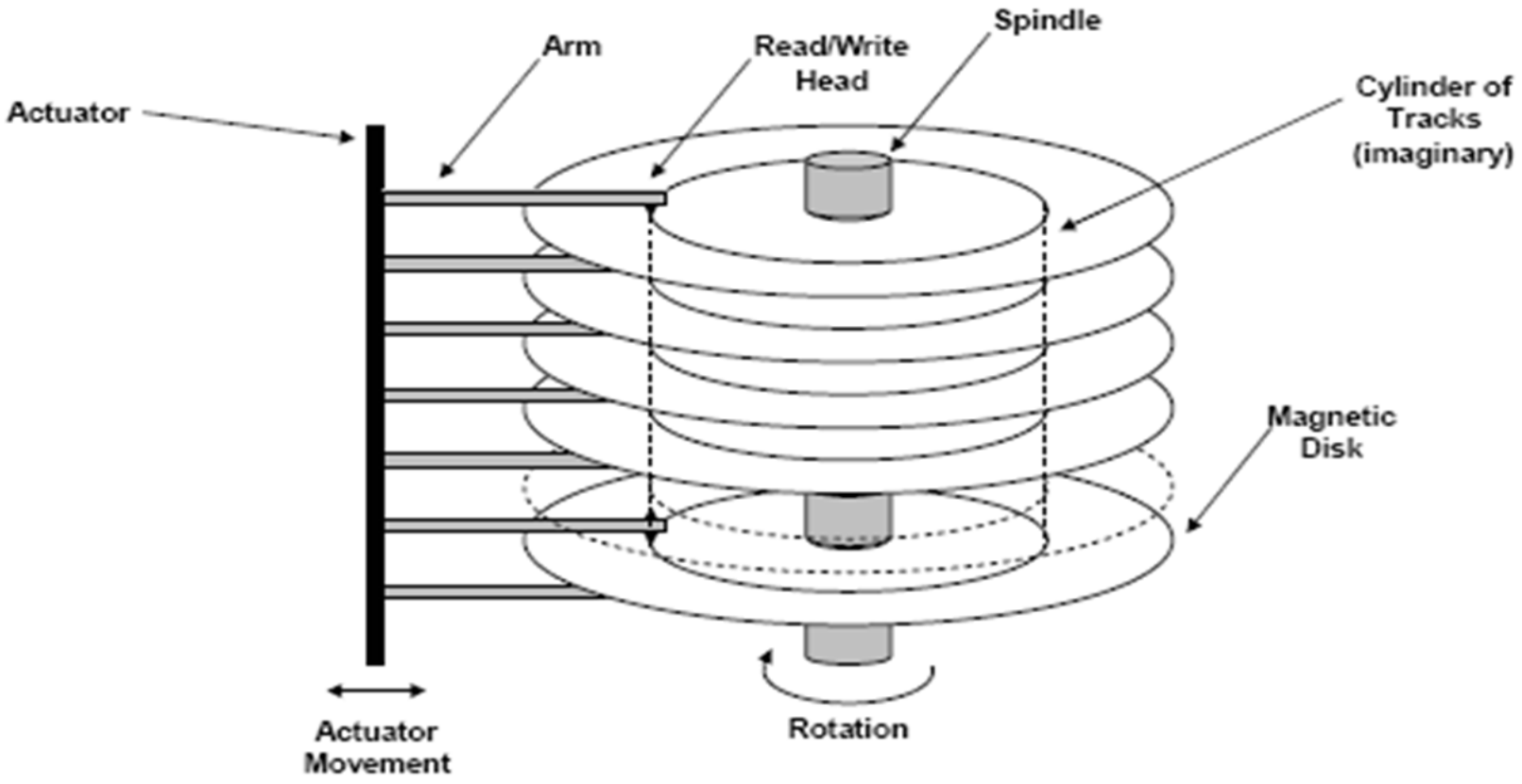
- Selection of blocking factor
- Organization of physical blocks in secondary storage
- Design of access method
- Select primary key
- File Growth

Types of Operations

- Retrieve_All
- Retrieve_One
- Retrieve_Next
- Retrieve_Previous
- Insert_One
- Update_One
- Retrieve_Few

What is the significance of cylinders?

- All information under a cylinder can be accessed without moving the arm that holds read-write heads.



Estimating Capacity of Disk

- The total capacity of disk is a function of cylinder, track & sector:

```
Track capacity=num of sectors per track * bytes per
sector

Cylinder Capacity = num of tracks per cylinder *
track capacity

Driver capacity = Number of Cylinder * Cylinder
Capacity
```

Example

Question: Consider a disk with sector size=512 Bytes

2000 tracks per surface (of a platter), with 5 double sided platters

50 sectors per track. What is the capacity of disk?

Example

Question: Consider a disk with sector size=512 Bytes

2000 tracks per surface (of a platter), with 5 double sided platters

50 sectors per track. What is the capacity of disk?

Capacity of track: 50*512 = 25600

Capacity of a surface: 2000*25600 = 51200000

Capacity of disk: 512000000*5*2

* How many cylinders do we have?

* Can we use 256 bytes, 2048, 51200 as block size?
  - Too small, we can, Too big (for track)

Performance Measures of Disks

Measuring Disk Speed:

- Access time :
  - Seek time: time to move the arm over the correct track
    - ~5ms from track-to-track,
  - Rotational latency: time it takes to align the head with the sector to be accessed.
    - For average you can use half of the time it takes to complete a full-rotation (e.g. 8.3ms)
  - Data transfer rate: The time required to retrieve the data under the read/write head.

Analogy to taking a bus:

1. Seek time: time to get to bus stop
2. Latency time; time spent waiting at bus stop
3. Data transfer time: time spent riding the bus

Cost of Access

Access Time = Seek Time + Rotational Delay

Time to Read = Seek Time + Rotational Delay + Transfer time

How can you calculate seek time, rotational delay and transfer time?

Answer

- Seek time: usually provided with the specs. of the disk
- Rotational delay: On average half of the time required for a complete revolution. Revolutions per minute (rpm) can be used to determine this
- Transfer time = (Num. of sectors transferred / Num. of sectors on a track) * rotation time

Example

ST3120022A : Barracuda 7200.7

Capacity	:	120 GB
Interface	:	Ultra ATA/100
RPM	:	7200 RPM
Seek time	:	8.5 ms avg
Latency time?	:	

- 7200/60 = 120 rotations/sec
- 1 rotation in 8.3 ms => So, Average Latency = 4.16 ms

Example 2

Disc specs:

- 20 plate, 800 track/plate, 25 sector/track, 512 byte/sector, 3600 rpm
- 7 ms from track to track, 28 ms avg. seek time, 50 msec max seek time

Avg. Rotational time =

Disk capacity =

Time to read track =

Time to read cylinder =

Time to read whole disc =

from one cylinder to another =

Example 2

Disc specs:

- 20 plate, 800 track/plate, 25 sector/track, 512 byte/sector, 3600 rpm
- 7 ms from track to track, 28 ms avg. seek time, 50 msec max seek time

Avg. Rotational time	=1/2 * (60000/3600) = 16.7/2 = 8.3msec
Disk capacity	= 25 * 512 * 800 * 20 = 204.8 MB
Time to read track	= 1 rotation = 16.7 msec
Time to read cylinder	= 20 * 16.7 = 334msec
Time to read whole disc	= 800 * time to read cylinder + 799* time to pass from one cylinder to another =800*334 + 799*7msec =267 sec + 5.59 sec = 272.59 sec

Example 3

- Disc specs: avg. seek time 8 msec, 10.000 rpm, 170 sector/track, 512 byte/sector

Time to read one sector =

Time to read 10 sequential blocks =

Time to read random 10 block =

Example 3

- Disc specs: avg. seek time 8 msec, 10.000 rpm, 170 sector/track, 512 byte/sector

Time to read one sector = avg. seek time + rotational delay + time to transfer one sector

= 8 + (0.5 * 60.000 / 10.000) + (6* 1/170) = 8 + 3+ 0.035 = 11.035 msec

Time to read 10 sequential blocks = avg. seek time + rotational delay + 10 * time to transfer one sector = 8 + 3 + 10 * 0.035 = 11.35 msec

Time to read random 10 block= 10 * (avg. seek time + rotational delay + time to transfer one sector)= 10 * (8 + 3 + 0.035) = 113.5 msec

Your turn: Exercise

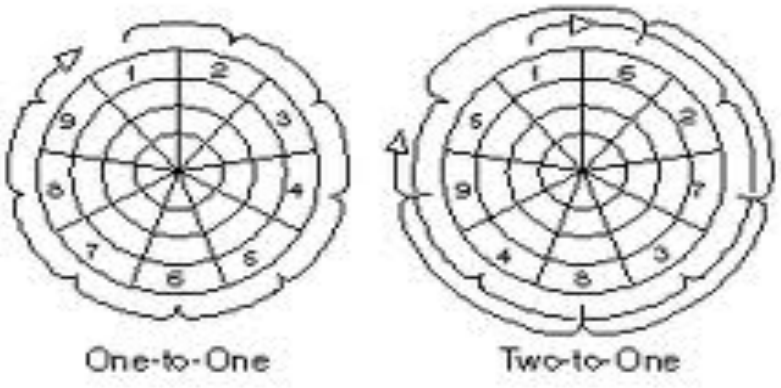
You have total 200.000 records stored in heap file and length of each record is 250 byte.

Here are the specifications of disk used for storage: 512 byte/sector, 20 sector/track, 5 sector/cluster, 3000 track/platter, one sided total 10 plate, rotational time 7200 rpm, seek time 8 msec. Records do not span between sectors

- Blocking factor of heap file: 10
- Number of blocks to store whole file: 20.000 blocks
- Number of blocks to estimate disk capacity: 120.000 blocks
- Cylinder number to store the data with cylinder based approach : 500
- Time to read one block : 14,225 ms
- Time to read file from beginning to end (in worst case): 284.500sec, 4741,67 min, 79,02 hour

Concepts Related to Sector

- Interleaving



- Extent

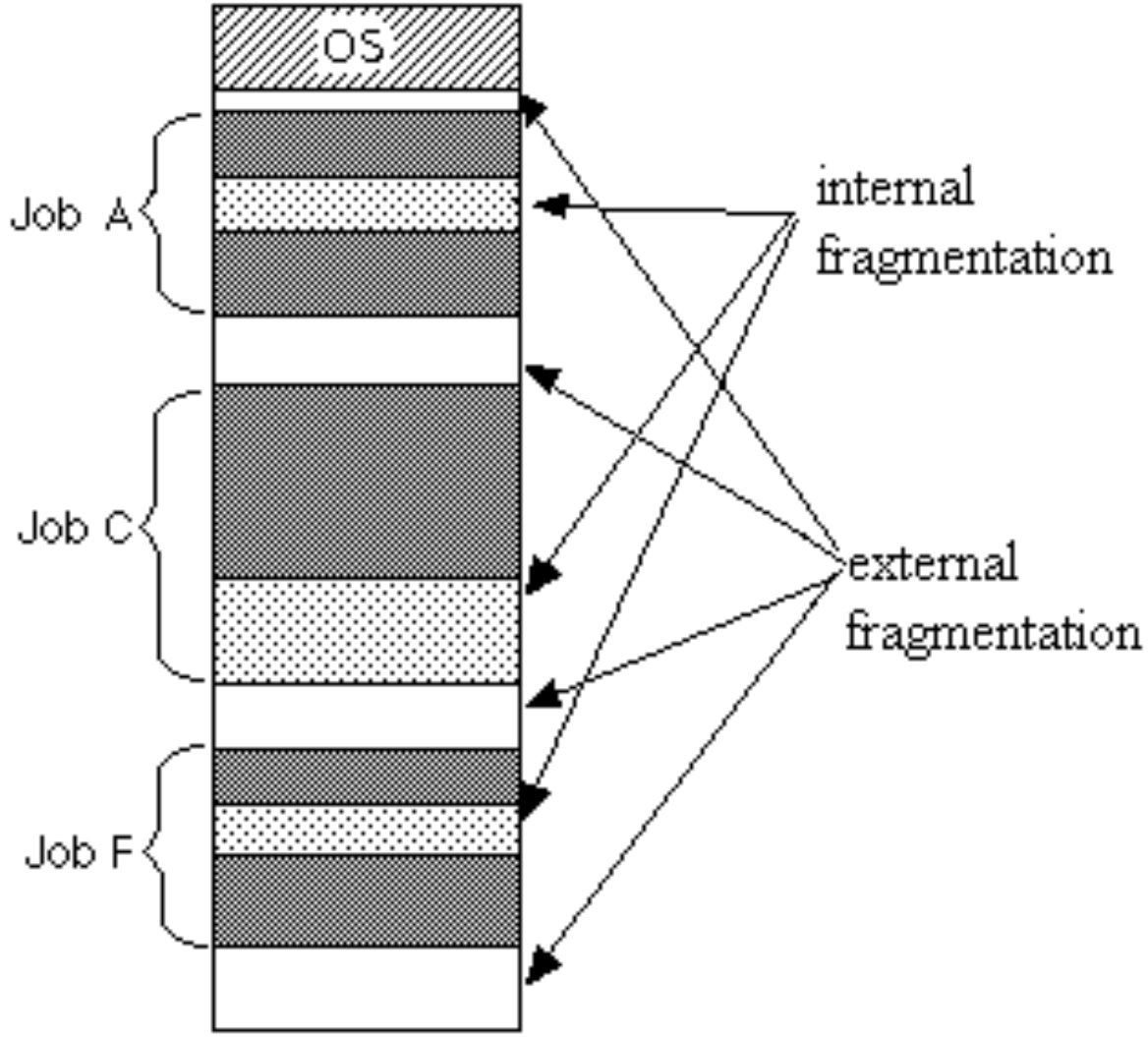
An **extent** is a contiguous area of storage in a computer file system, reserved for a file. When a process creates a file, file-system management software allocates a whole extent. When writing to the file again, possibly after doing other write operations, the data continues where the previous write left off. This reduces or eliminates file fragmentation and possibly file scattering too.

External Fragmentation

- External fragmentation arises when free memory is separated into small blocks and is interspersed by allocated memory.
- The result is that, although free storage is available, it is effectively unusable because it is divided into pieces that are too small individually to satisfy the demands of the application. The term "external" refers to the fact that the unusable storage is outside the allocated regions.
- For example, consider a situation wherein a program allocates 3 continuous blocks of memory and then frees the middle block. The memory allocator can use this free block of memory for future allocations. However, it cannot use this block if the memory to be allocated is larger in size than this free block.
- External fragmentation also occurs in file systems as many files of different sizes are created, change size, and are deleted. The effect is even worse if a file which is divided into many small pieces is deleted, because this leaves similarly small regions of free spaces.

0x0000	0x1000	0x2000	0x3000	0x4000	0x5000	Comments
						Start with all memory available for allocation.
A	B	C				Allocated three blocks A, B, and C, of size 0x1000.
A		C				Freed block B. Notice that the memory that B used cannot be included for an allocation larger than B's size.

Internal vs External Fragmentation



Disk Failures

- **Intermittent Failures:** An attempt to read or write a sector is unsuccessful.
- Solution: Checksums
  - A simple form is based on the parity of the bits in the sector.
  - A sector computes an error checking code (called checksum) and stores next to sector
  - Example: If the sequence of bits in a sector were 01101000, then there is an odd number of 1's, so the parity bit is 1.
- For the parity information
  - Even number of 1s will have parity 0
  - Odd number of 1s will have parity 1
- Example: Data is 1010111
- If error changes data to 1110111 parity should be 0, but was 1. Detected the error!

Mirroring as a Redundancy Technique

- **Disk crashes:** Data is permanently destroyed
- Solution: Mirroring
- Say a bank has k different hard drives
  - If each hard drive is error free 99% of the time
  - The probability that k hard drives will not have errors at the same time is:

    Availability = $(0.99)^k$, so for example if k=20 then it is 0.817

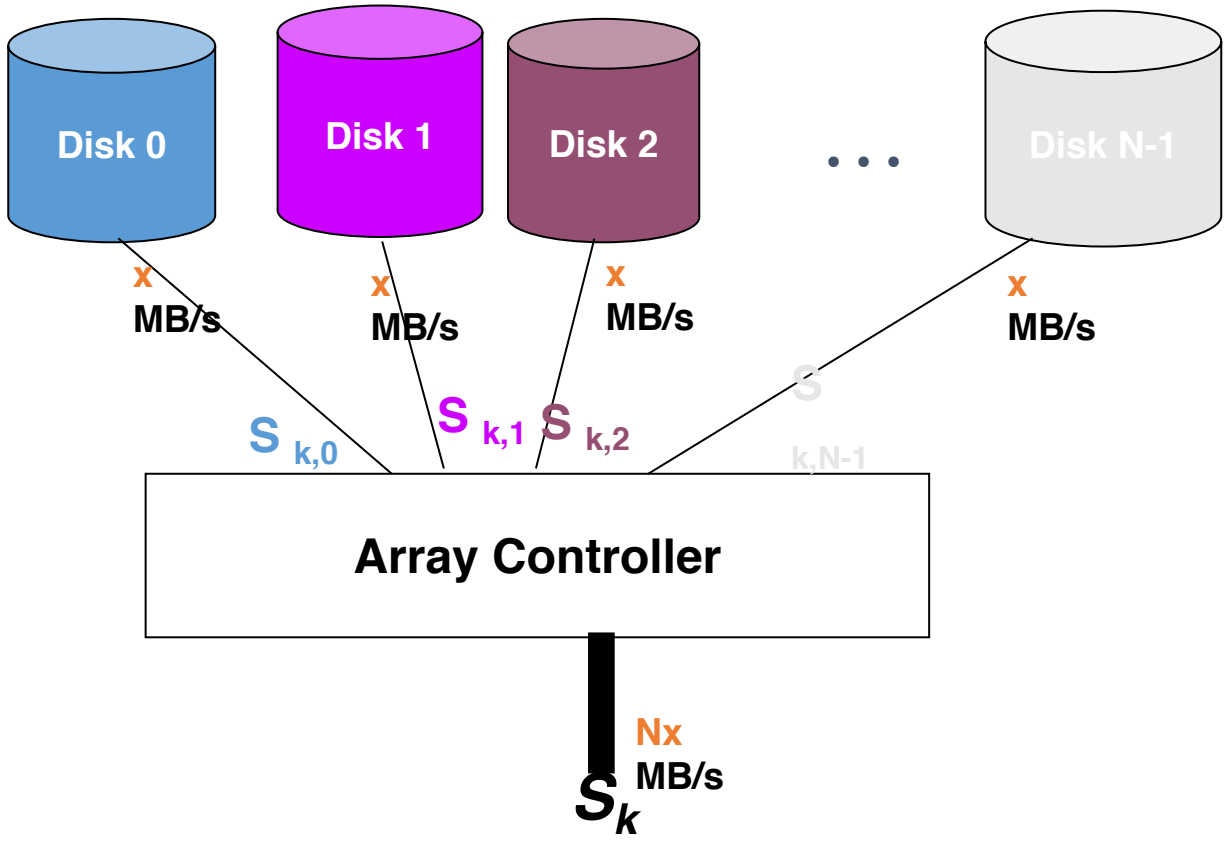
- If we keep multiple copies of the same data in *r* disks:
  - The probability of failure is 0.01
  - The probability of hard drives failing at the same time is $(0.01)^r$, if we assume that failures are independent.
  - Even if r=2, the probability of losing the data drops to 0.0001

Redundant Array of Inexpensive Disks (RAID)

- Disk Array: Arrangement of several disks that gives abstraction of a single, large disk.
- Goals: Increase performance and reliability.
  - Data striping: Data is partitioned
  - Partitions are distributed over several disks.
- Note that reliability decreases by reducing storage space utilization. We use some hard drive space for redundant data.

- We extend the idea of parity checks through redundancy!
  - The common term for this class of strategies is RAID!

RAID


Disk 0
Disk 1
Disk 2
Disk N-1
x MB/s
x MB/s
x MB/s
x MB/s
S k,0
S k,1
S k,2
S k,N-1
Array Controller
Nx MB/s
S k

RAID

- RAID Level 1: We mirror each disk, one as a data disk and one as a redundant disk
- RAID Level 2: Use only one redundant disk that store parity blocks

- What happens if there are two or more disk crashes at the same time?

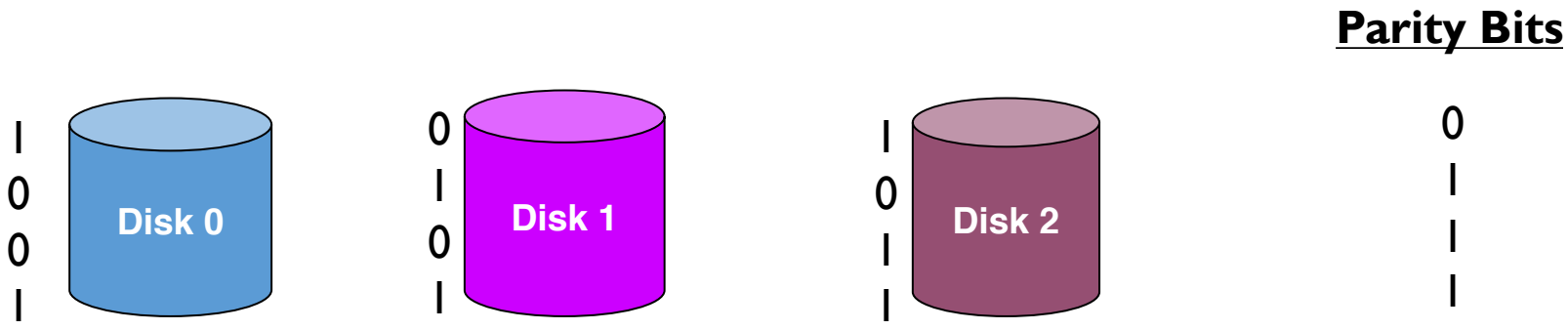
- RAID Level 5: Each disk is a redundant disk for some of the blocks
  - Example: If there are 4 disks, the disk 0 is redundant for blocks 4, 8, 12, and so on; disk 1 is redundant for blocks 1, 5, 9, and so on.

- RAID Level 6: It is designed for multiple disk crashes. It uses Hamming code.

RAID Error Correction

Data Striping by 4 bits, lets assume we have one parity for the 3 disks

100101011011



If Disk 0 Crashes, we can recover the bits by simply XOR all the others with parity bits.
Disk 0 = Disk 1 XOR Disk 2 XOR Parity