

Armazenamento Secundário (part 1)

Prof. Dr. Lucas C. Ribas

Disciplina: Estrutura de Dados II

Departamento de Ciências de Computação e Estatística



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”



IBILCE / UNESP - CÂMPUS DE SÃO JOSÉ DO RIO PRETO



- Armazenamento secundário
 - HDs
 - Organização de informações em disco
 - Sistemas de Arquivos



- Primeiro tipo de armazenamento secundário?
 - Papel!
 - Cartões perfurados
- HDs, CD-ROM, floppy disks, memórias flash, fita, etc.
- HD se distingue dos demais meios
 - Capacidade
 - Velocidade



- HDs evoluíram quase tão radicalmente quanto os processadores
 - Primeiro HD tinha menos de 5Mb, por U\$35.000
 - HDs melhores tinham cerca de 10Mb, mais de U\$100 por Mb
 - Atualmente menos de 1 centavo por Mb
 - IBM teve um papel importante!



- No início, mais em sistemas corporativos

- 1.70m de altura e de comprimento, quase 1 tonelada
- Chamado “unidade de disco”
- 5 MB

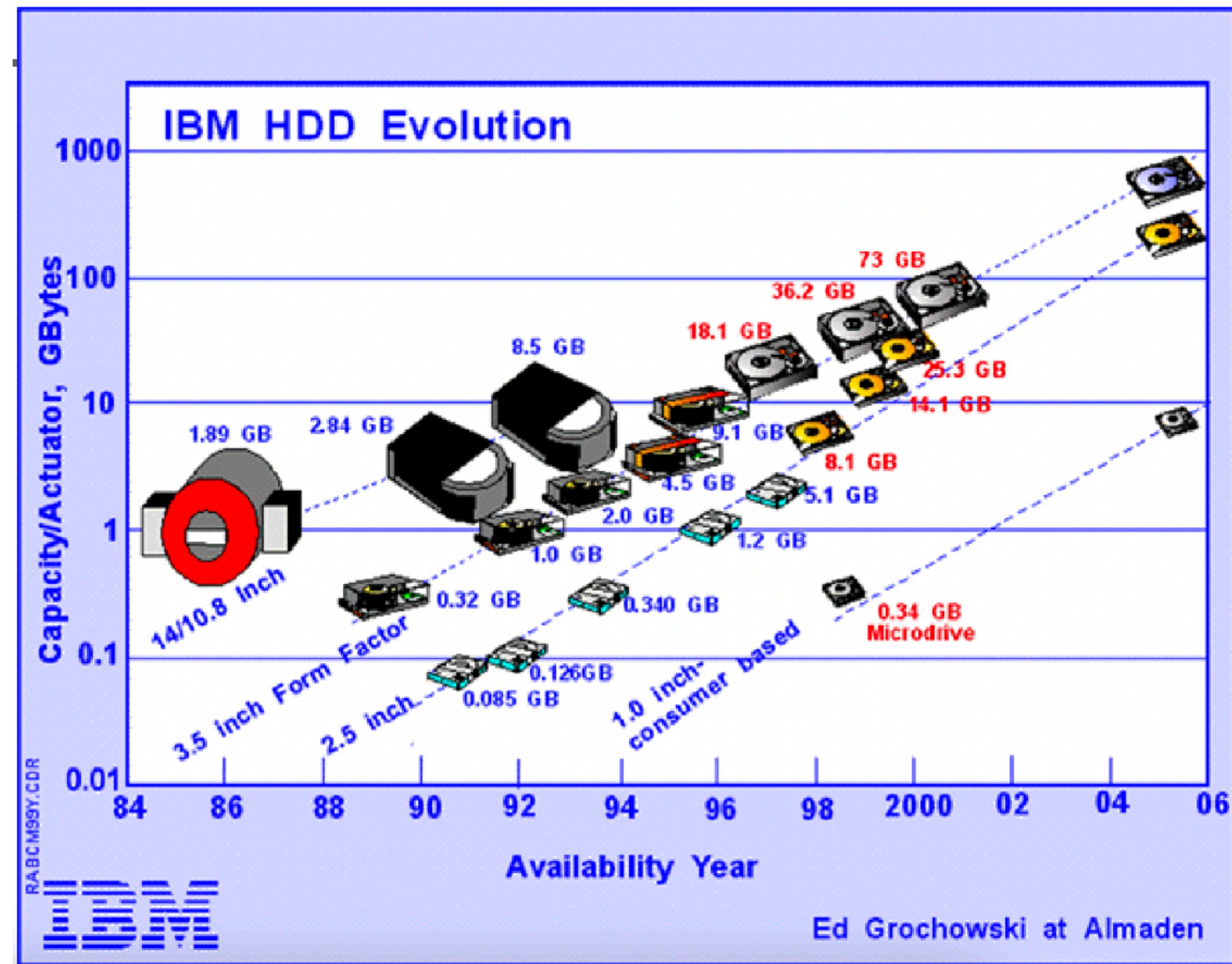


IBM 350 (1956)



● Papel fundamental em vários aspectos do computador

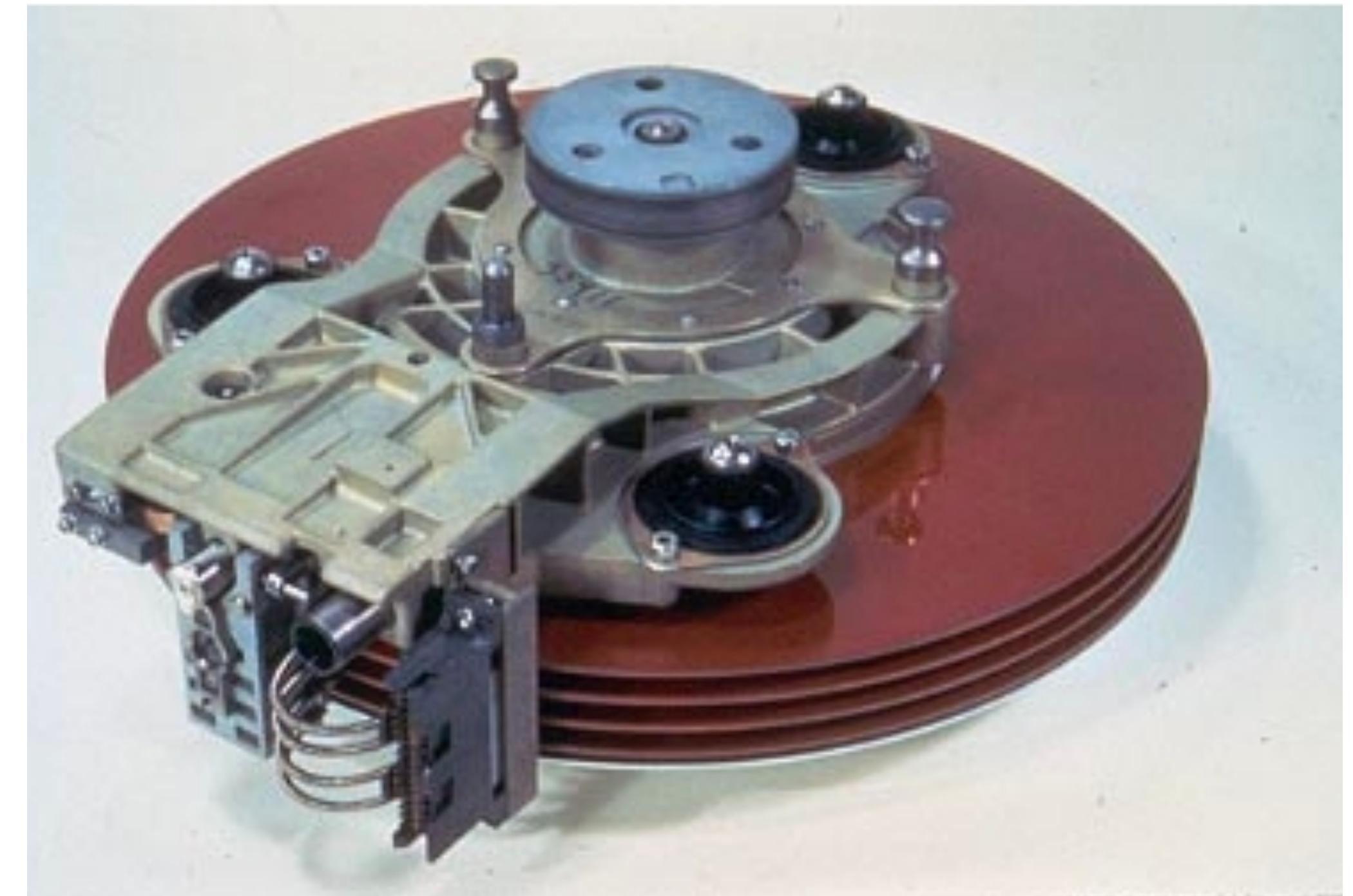
- Desempenho
 - Velocidade de acesso ao HD, boot, carregamento de programas, multitarefa
- Capacidade
 - Mais dados, programas maiores e mais complexos
- Confiabilidade
 - “A importância de um dispositivo se mede pelo impacto de sua perda”





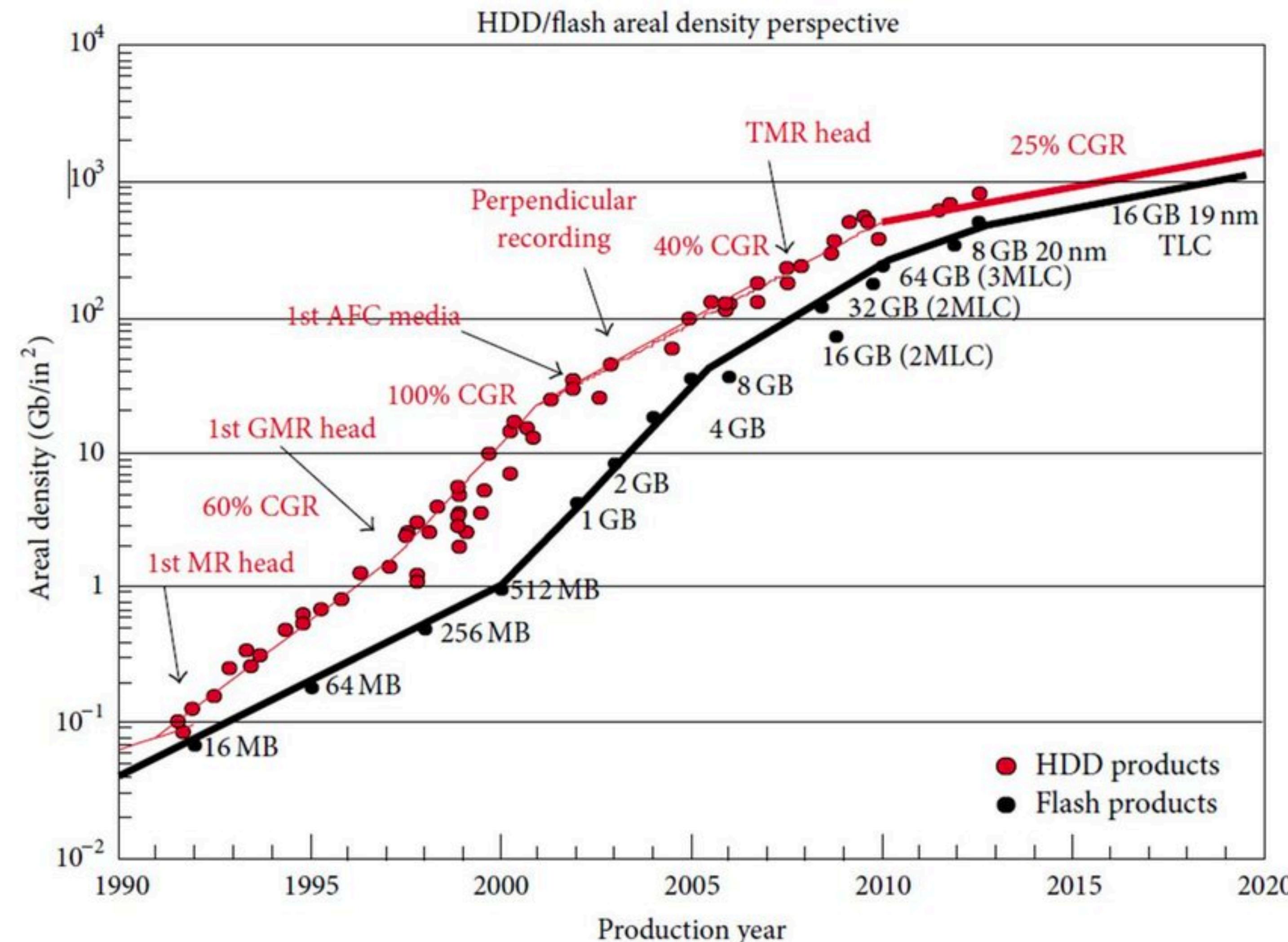
○ HD

- Em 1973, IBM lançou o que é considerado o pai dos HDs modernos
 - [Winchester](#)





- No início, as **cabeças de leitura dos discos encostavam** neles
 - Necessário para que os dispositivos eletrônicos antigos pudessem ler os campos magnéticos
- Grande avanço: **cabeças de leitura “flutuam”**
 - Quanto mais próximas as cabeças do disco, melhor
- Densidade de área, capacidade e desempenho melhoram a cada ano
- Aumentar densidade:
 - Mais dados no mesmo espaço
 - Mais trilhas, que pode passar a ter mais setores
 - **Problemas:** menor espaço para bit, enfraquece seu sinal magnético
 - Mídia de melhor qualidade (para dados estáveis) e cabeça de leituras mais sensíveis



Hai Li. Storage Physics and Noise Mechanism in Heat-Assisted Magnetic Recording, 2016



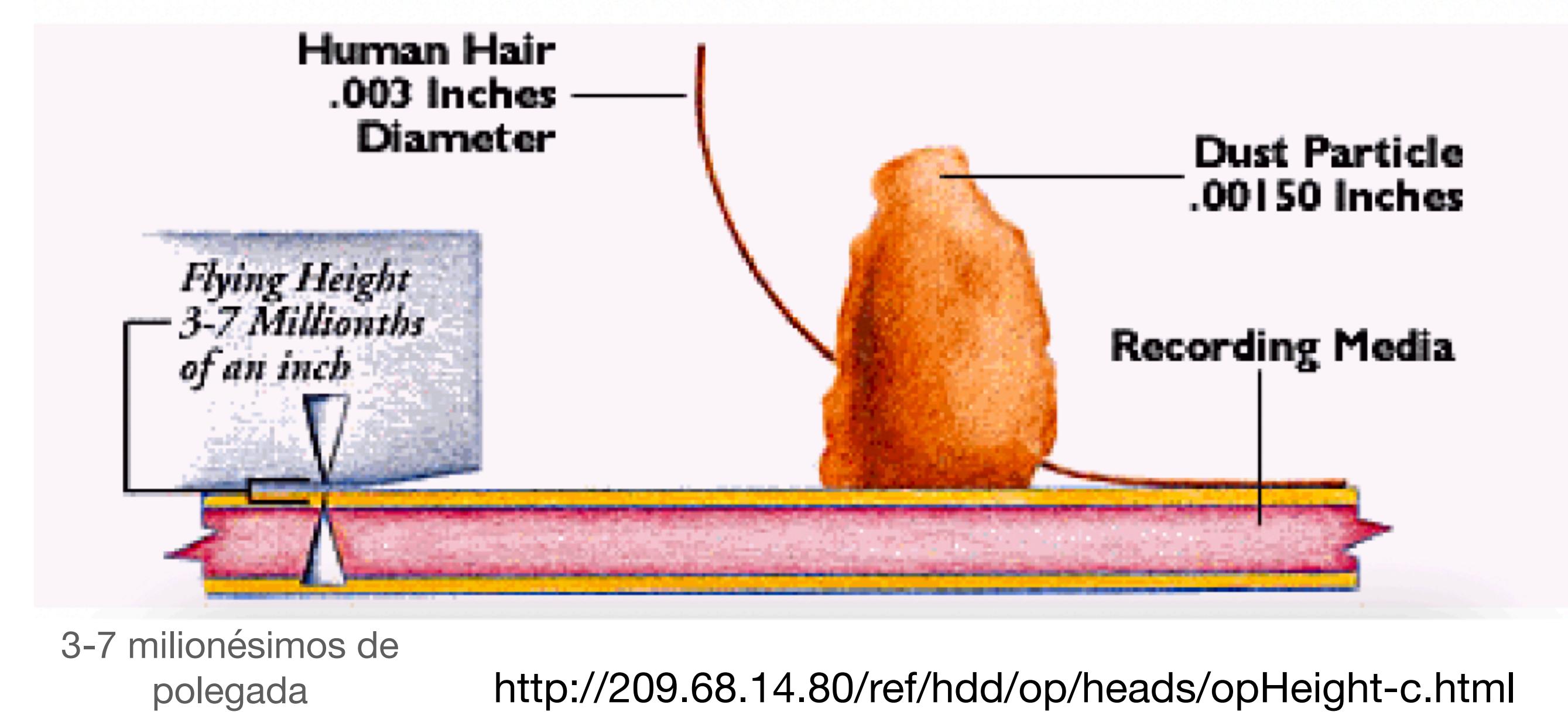
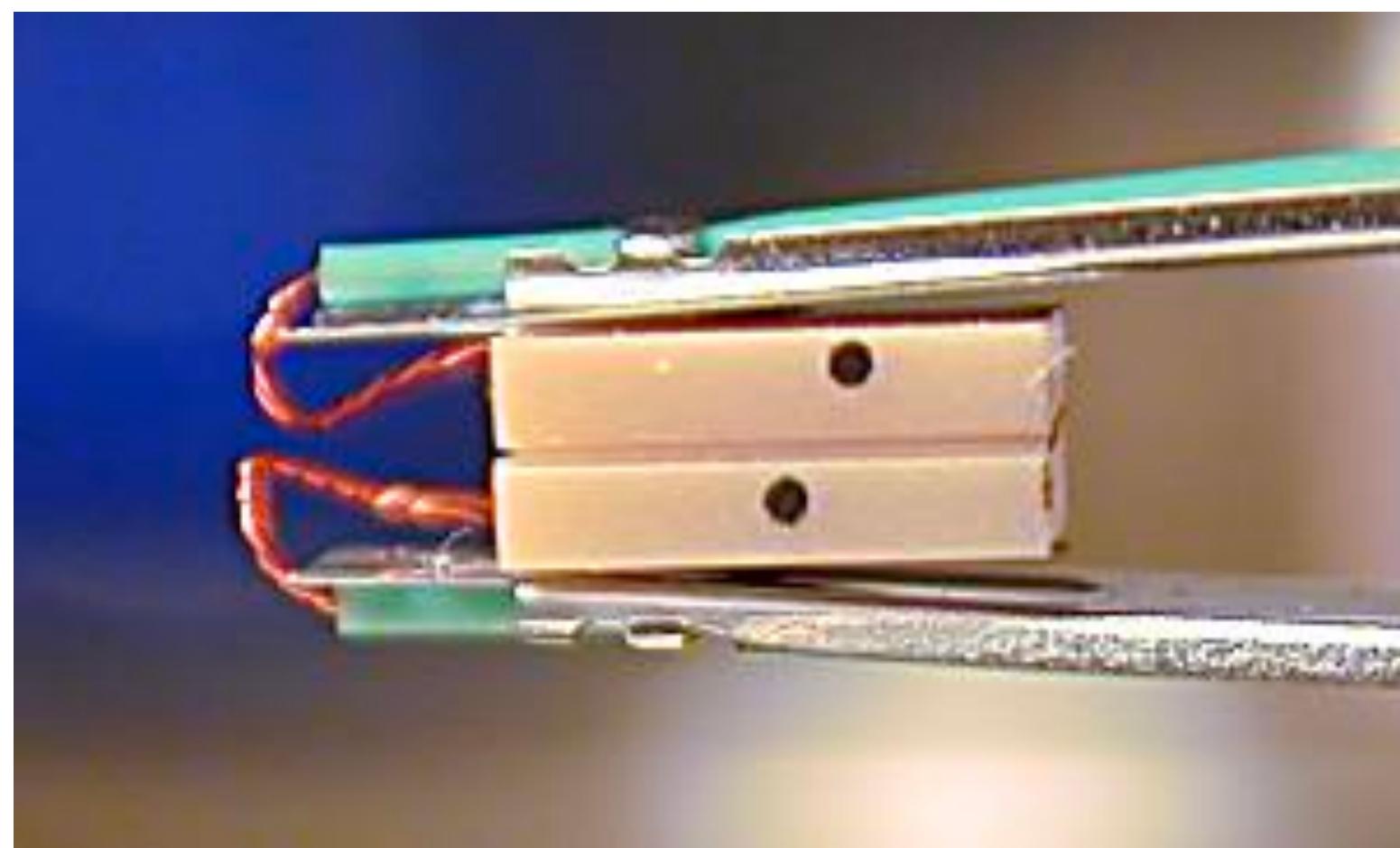
● Componentes importantes

- Discos (podem haver vários)
 - Substrato sólido (alumínio, vidro/cerâmica)
 - Superfície magnética: “ferrugem” no passado, filme magnético no presente, moléculas orgânicas no futuro
- Cabeças de leitura: lêem e escrevem nos discos enquanto eles giram (~10.000 RPM atualmente)
 - Convertem entre sinais elétricos e pulsos magnéticos
- A informação lida é armazenada em um buffer, de onde é transferida para a memória



- Distância em relação ao disco

- Flutuam em função do colchão de ar gerado quando o disco gira
- Não há vácuo dentro do disco!
- é montado em uma sala limpa contendo ar especialmente filtrado



<http://209.68.14.80/ref/hdd/op/heads/opHeight-c.html>

- HDs funcionam em altitudes altíssimas (mais de 3.000m)?



São normalmente projetados para funcionar a altitudes de até 3.000 metros acima do nível do mar. Em altitudes muito elevadas, a pressão do ar é menor, comprometendo a criação do colchão de ar.

Para casos extremos (uso militar, por exemplo), existem HDs pressurizados

Organização da informação no disco



- **Disco:** conjunto de ‘pratos’ empilhados
 - Dados são gravados nas superfícies desses pratos
- **Superfícies:** são organizadas em trilhas
- **Trilhas:** são organizadas em setores
- **Cilindro:** conjunto de trilhas na mesma posição

Organização da informação no disco

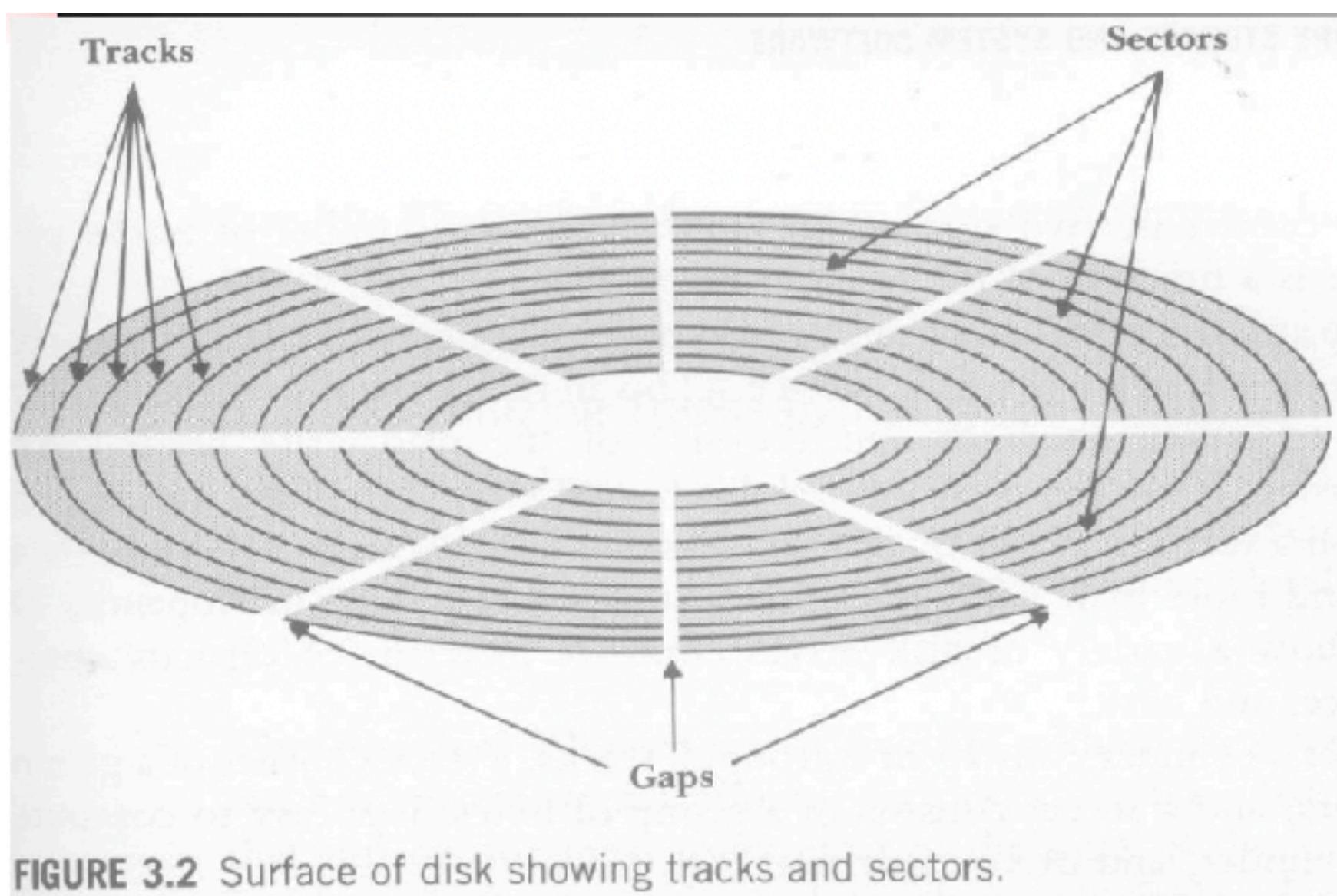


FIGURE 3.2 Surface of disk showing tracks and sectors.

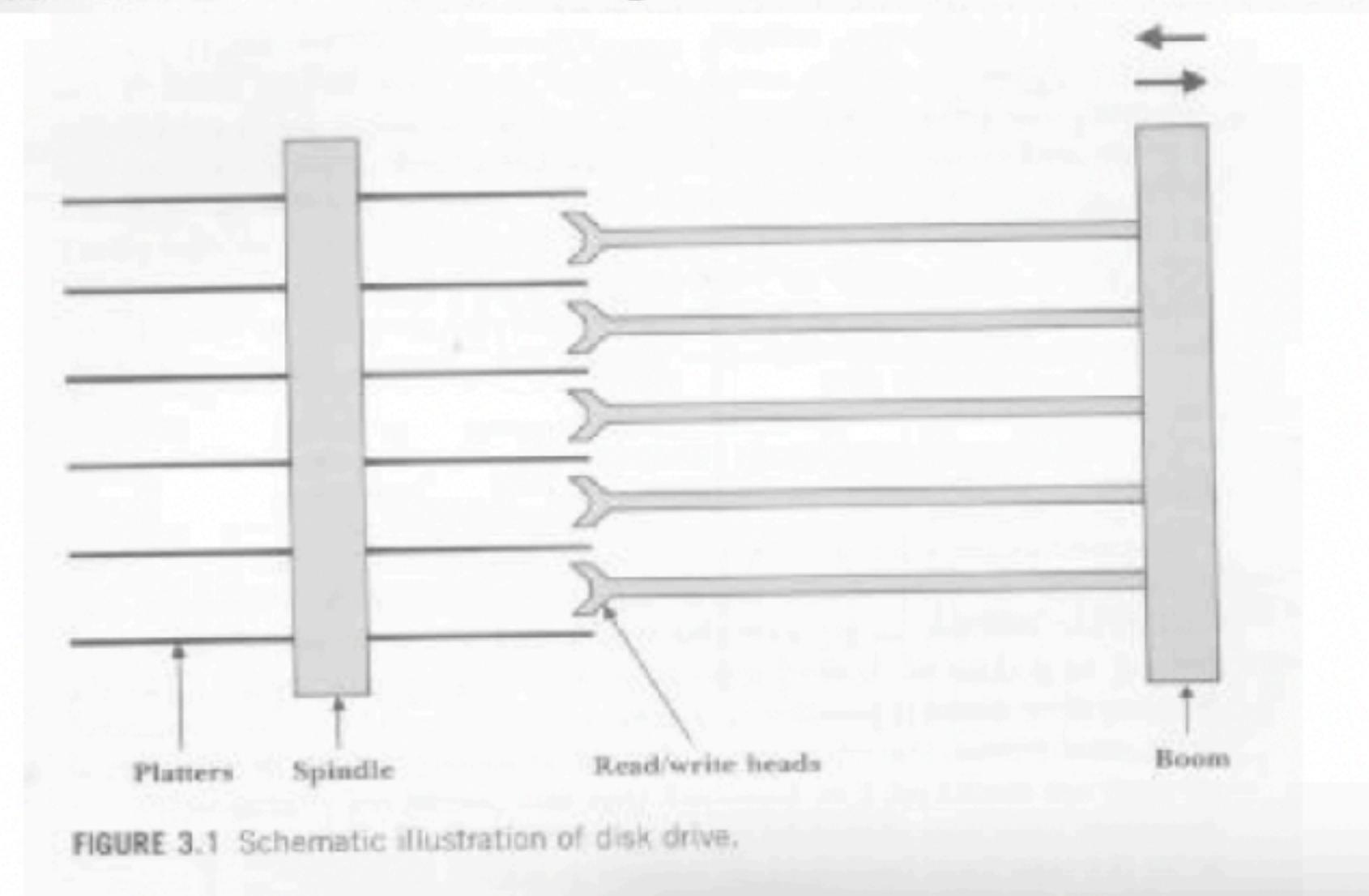
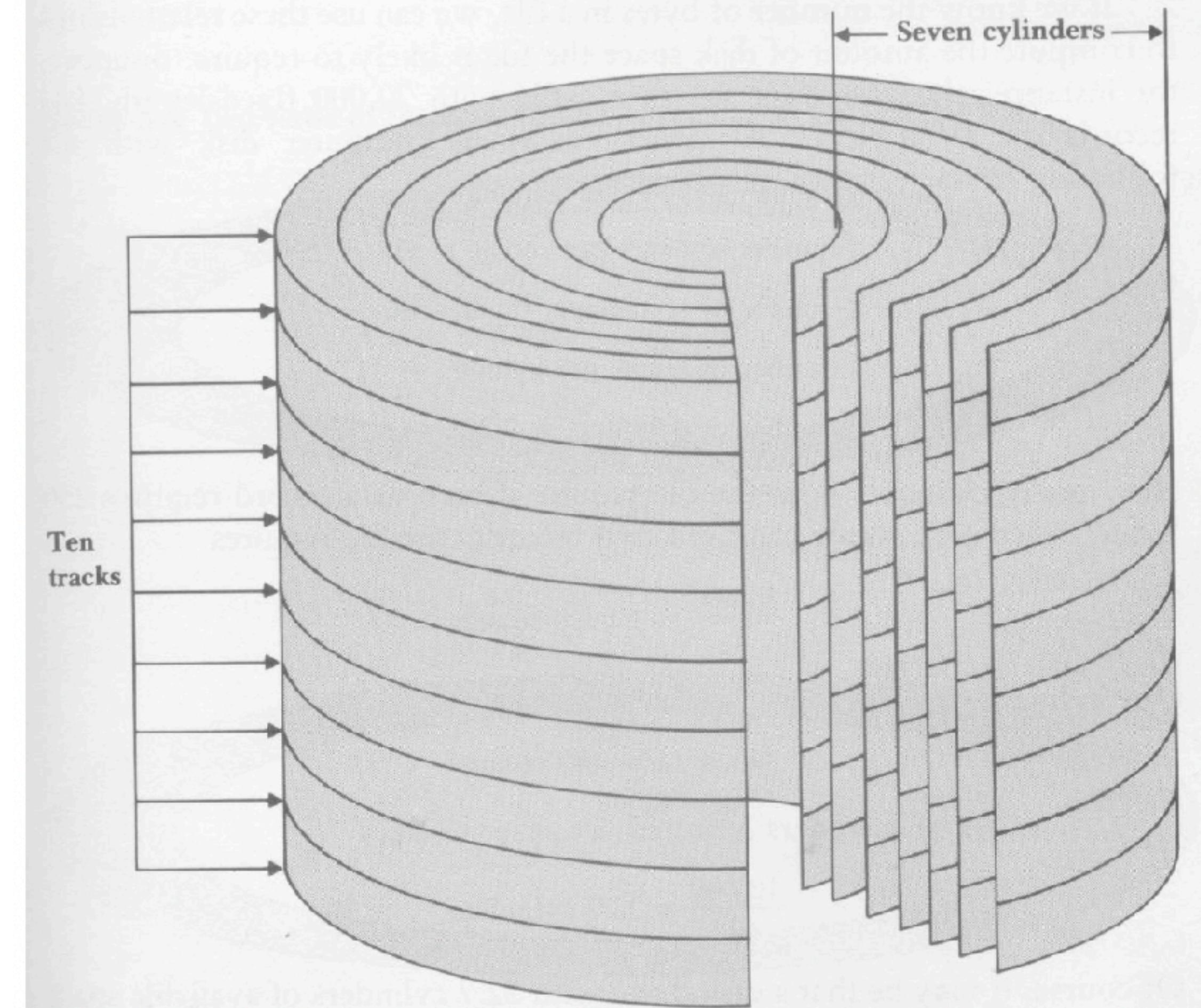
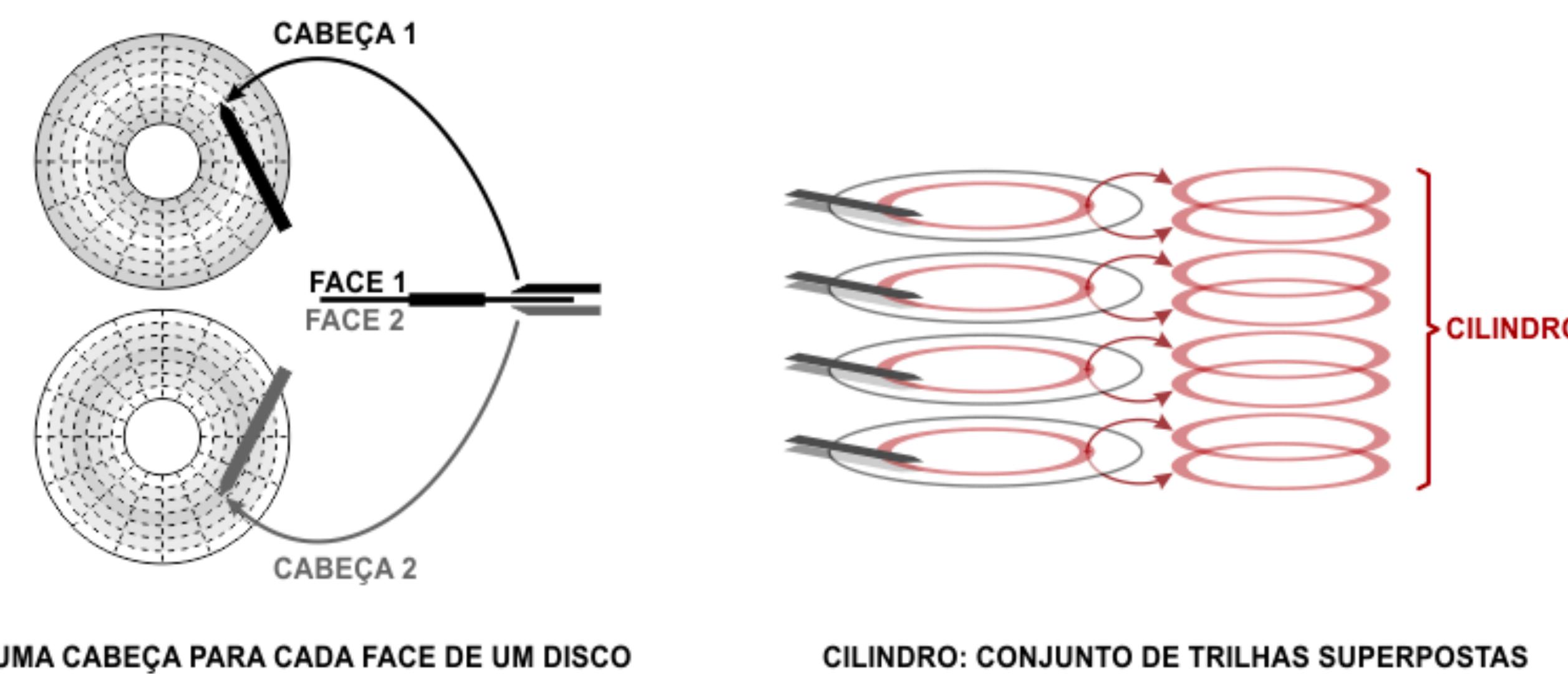
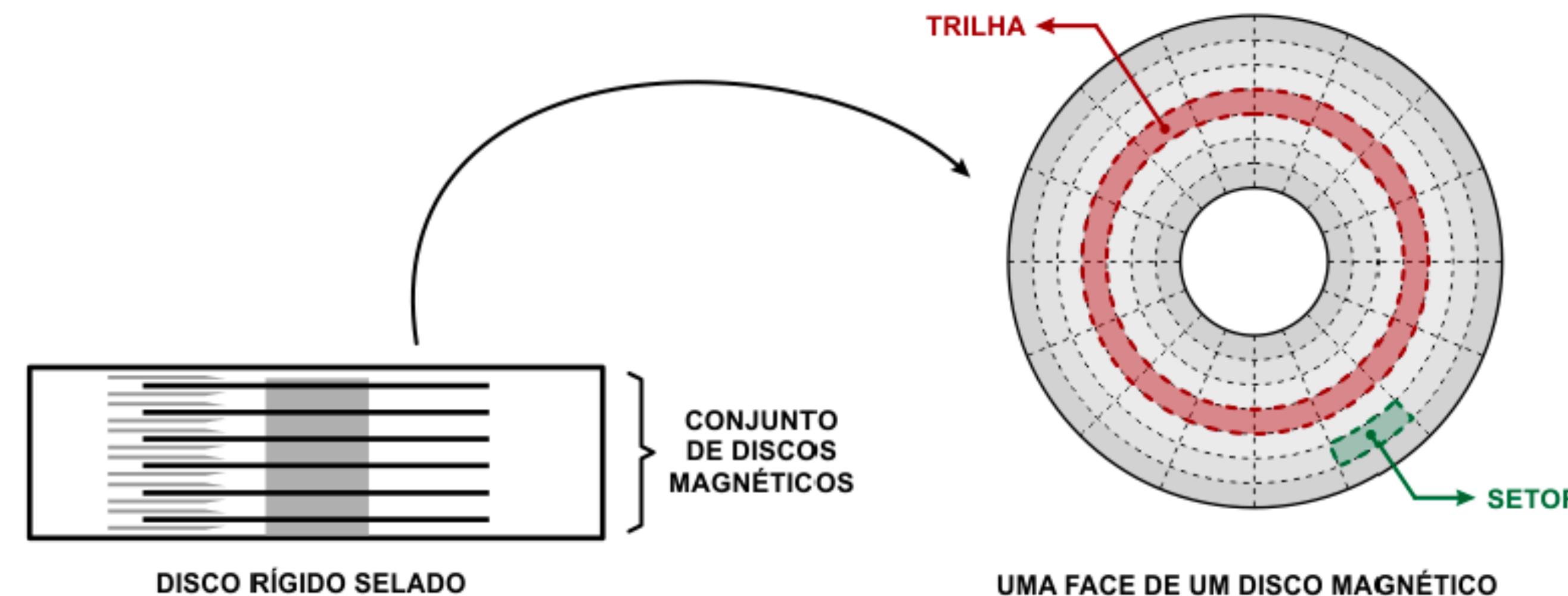


FIGURE 3.1 Schematic illustration of disk drive.

FIGURE 3.3 Schematic illustration of disk drive viewed as a set of seven cylinders.



Organização da informação no disco





- Um **setor** é a menor porção endereçável do disco
- Exemplo:
 - Read(fd,&c,1): lê 1 byte na posição corrente
 - S.O. determina qual a **superfície**, **trilha** e **setor** em que se encontra esse byte
 - O conteúdo do setor é carregado para uma memória especial (buffer de E/S) e o byte desejado é lido do buffer para a RAM. Se o setor necessário já está no buffer, o acesso ao disco torna-se desnecessário



- Movimento de posicionar a cabeça de L/E sobre a trilha/setor desejado
- O conteúdo de todo um cilindro pode ser lido com 1 único *seeking*
- É o movimento **mais lento** da operação leitura/escrita
- **Deve ser reduzido!**

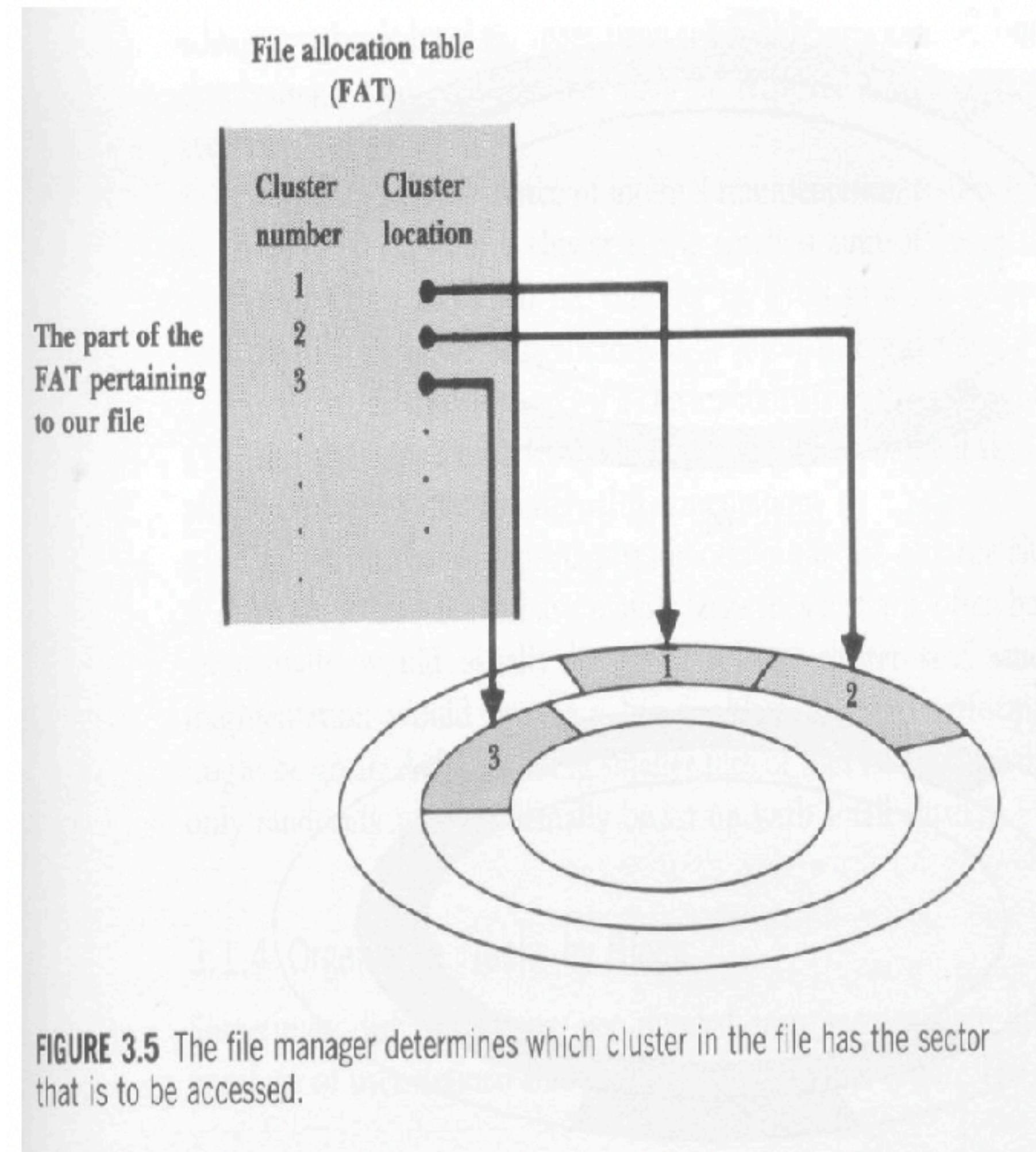


- Conjunto de setores logicamente contíguos no disco
- Um arquivo é visto pelo S.O. como um grupo de *clusters* distribuído no disco
 - Arquivos são alocados em um ou mais *clusters*

FAT - File Allocation Table

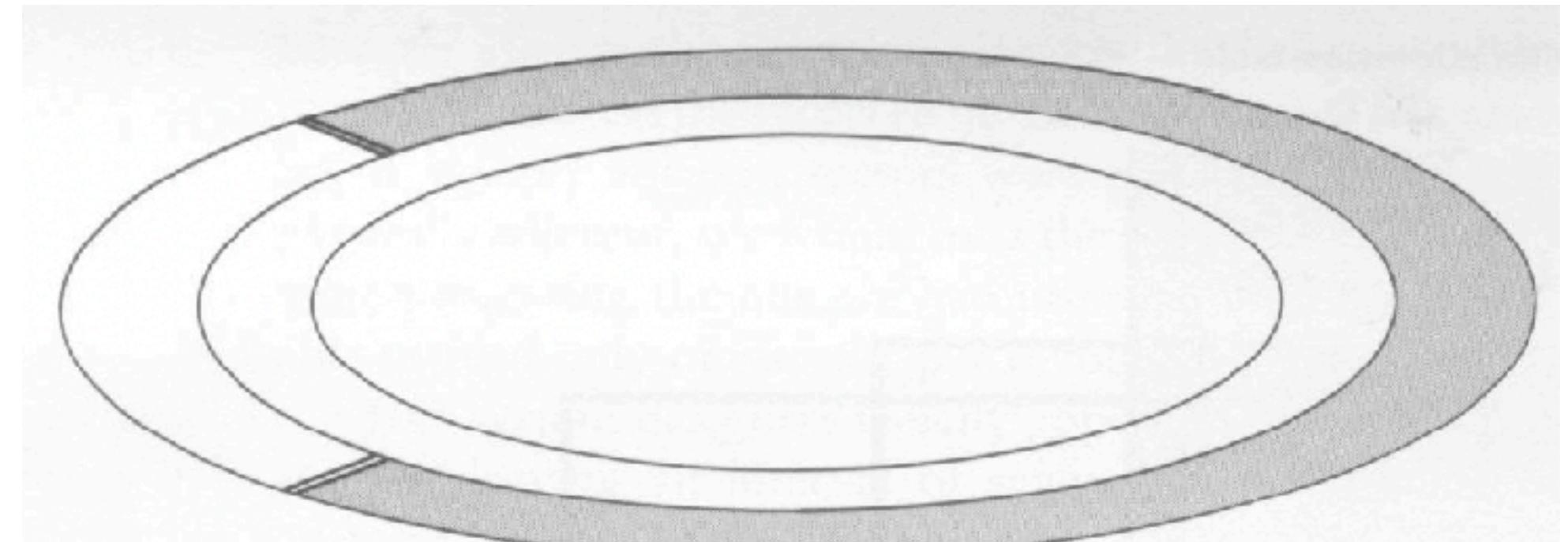


- Cada entrada na tabela dá a localização física do *cluster* associado a um certo arquivo lógico
- **1 seeking para localizar 1 cluster**
 - Todos os setores do *cluster* são lidos sem necessidade de seeking adicional

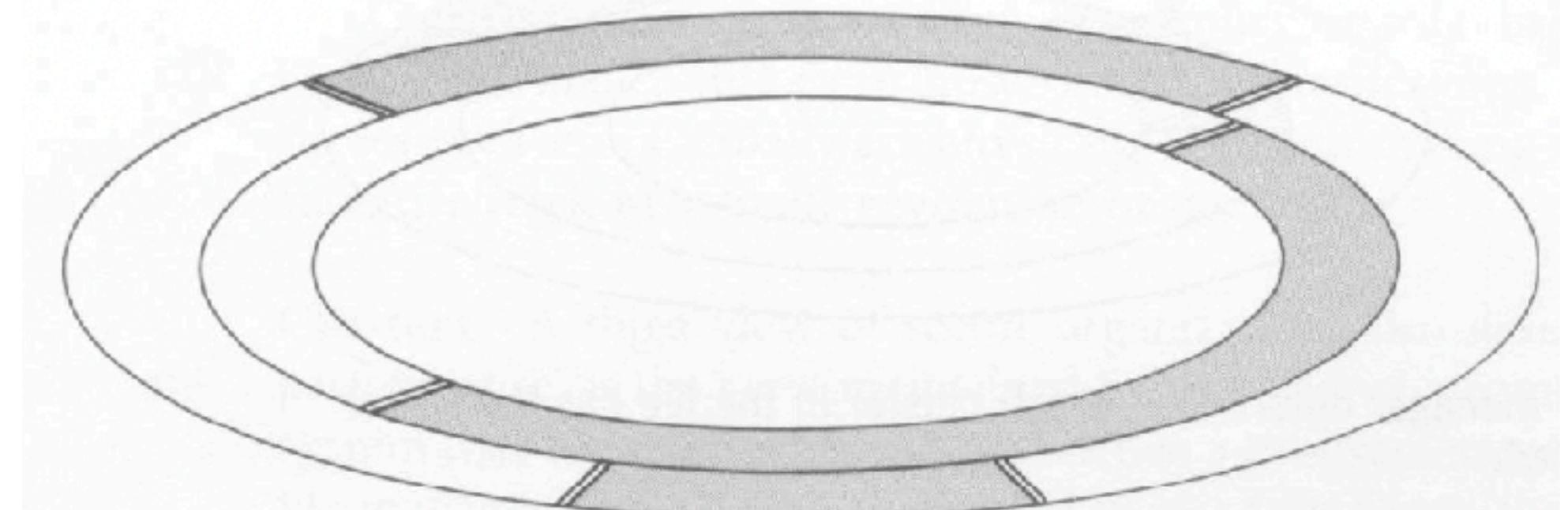




- Sequência de clusters consecutivos no disco, alocados para o mesmo arquivo
- 1 seeking para recuperar 1 extent
- A situação ideal é um arquivo ocupar 1 extent
 - Frequentemente isso não é possível e o arquivo é espalhado em vários extents pelo disco



(a)



(b)

FIGURE 3.6 File extents (shaded area represents space on disk used by a single file).

Capacidade do disco (nominal)



○ Capacidade do setor

- n° bytes (Ex. 512 bytes)

○ Capacidade da trilha

- n° de setores/trilha * capacidade do setor

○ Capacidade do cilindro

- n° de trilhas/cilindro * capacidade da trilha

○ Capacidade do disco

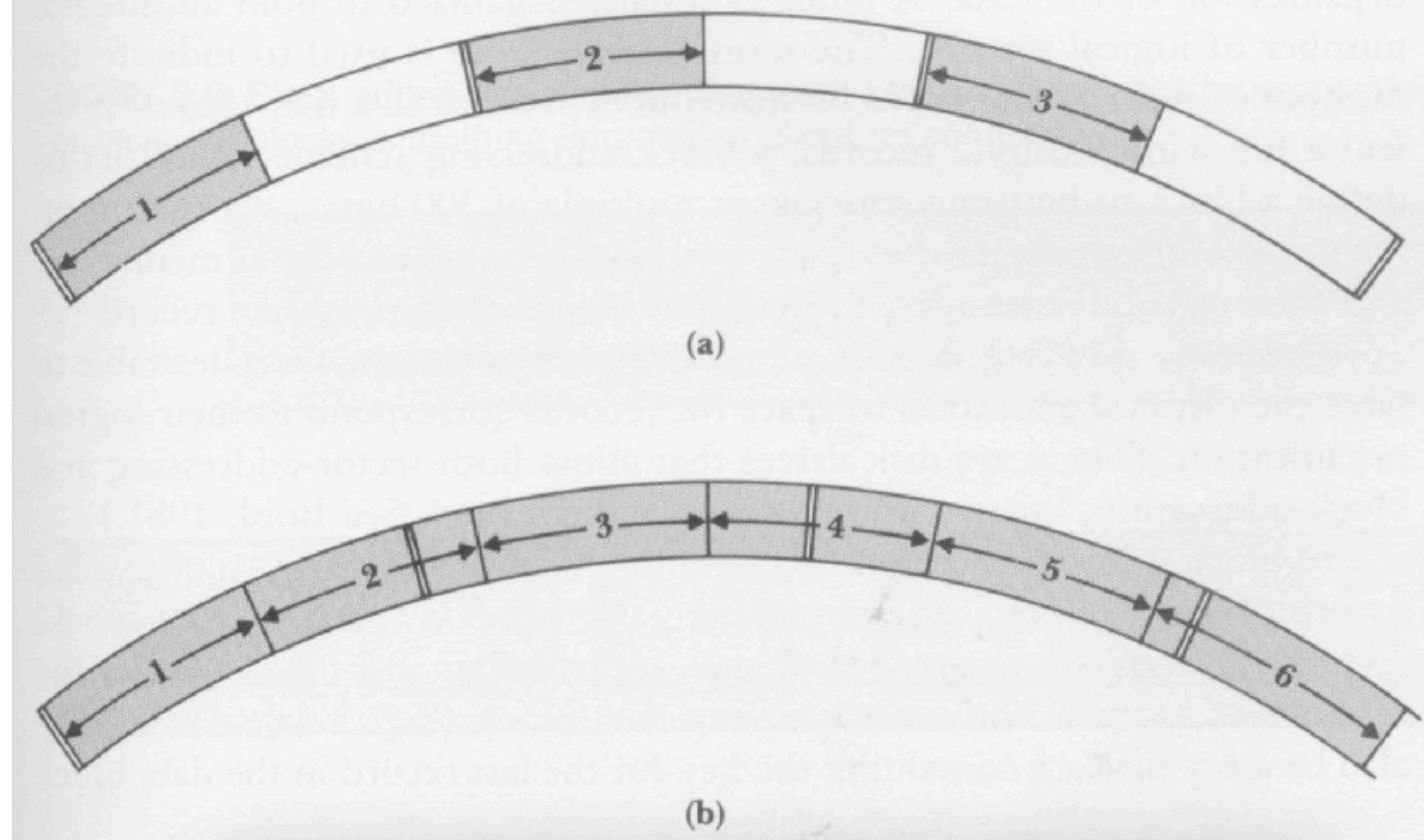
- n° de cilindros x capacidade do cilindro

Fragmentação interna



- Perda de espaço útil decorrente da organização em setores de tamanho fixo
- Ex: setor de 512 bytes, arquivos c/ registro de 300 bytes. Temos duas alternativas:
 - 1 registro por setor => fragmentação
 - Registros ocupando mais de 1 setor => acesso mais complexo

FIGURE 3.7 Alternate record organization within sectors (shaded areas represent data records, and unshaded areas represent unused space).





- A organização do disco em setores/trilhas/cilindros é **uma formatação física** (já vem da fábrica)
 - Pode ser alterada se o usuário quiser dividir o disco em partições
- É necessária uma **formatação lógica**, que ‘instala’ o sistema de arquivos no disco
 - Subdivide o disco em regiões endereçáveis
- **Sistema de arquivos:** estruturas lógicas e sub-rotinas usadas para controlar acesso aos dados em disco



- O sistema de arquivos **FAT** (Windows) não endereça setores, mas grupos de setores (*clusters*)
 - 1 *cluster* = 1 unidade de alocação
 - 1 *cluster* = n setores
- Um arquivo ocupa, no mínimo, 1 cluster
 - Unidade mínima de alocação
- Se um programa precisa acessar um dado, cabe ao sistema de arquivos do SO determinar em qual *cluster* ele está (FAT)

Fragmentação interna (*clusters*)



- Fragmentação também ocorre organizando os arquivos em *clusters*!
 - Ex: 1 cluster = 3 setores de 512 bytes, arquivo com 1 byte (quanto espaço se perdeu?)
- **Alternativa:** alguns S.O. organizam as trilhas em **blocos** de tamanho definido pelo usuário



- Espaço ocupado com informações para gerenciamento (não c/ dados), introduzidas pelo processo de formatação do disco
- O *overhead* existe tanto **em discos organizados por setor quanto em discos organizados por blocos**



- Definido automaticamente pelo SO quando o disco é formatado
- FAT (Windows): sempre uma potência de 2
 - 2, 4, 8, 16 ou 32KB
- Determinado pelo máximo que a FAT consegue manipular, e pelo tamanho do disco
 - FAT16: pode endereçar 2^{16} clusters = 65.536
- Quanto maior o *cluster*, maior a fragmentação!

Outros Sistemas de Arquivos



- FAT32 (Windows 95 e posteriores)
 - *clusters* de tamanho menor, endereça mais *clusters*, menos fragmentação
- NTFS (New Technology File System)
 - Sistemas OS/2 (IBM) e Windows NT
 - Mais eficiente: a menor unidade de alocação é o próprio setor de 512 bytes

Custo de acesso a disco



- É uma combinação de 3 fatores:
 - **Tempo de busca (seek):** tempo para posicionar o braço de acesso no cilindro correto
 - **Delay de rotação:** tempo para o disco rodar de forma que a cabeça de L/E esteja posicionada sobre o setor desejado
 - **Tempo de transferência:** tempo p/ transferir os bytes
 - **Tempo transferência=(nº de bytes transferidos/nº de bytes por trilha)*tempo de rotação**



- Os **tempos de acesso reais** são afetados não só pelas características físicas do disco
 - Também pela distribuição do arquivo no disco
 - e **modo de acesso** (aleatório x sequencial)



- Você sabe o seguinte sobre seu HD
 - Número de bytes por setor: 512
 - Número de setores por trilha: 40
 - Número de trilhas por cilindro: 11
 - Número de cilindros: 1.331
- Há um conjunto de dados composto por 20.000 registros, sendo que cada registro tem 256 bytes
- Quantos cilindros são necessários para se armazenar esses 20.000 registros?



- Você sabe o seguinte sobre seu HD
 - Número de bytes por setor: 512
 - Número de setores por trilha: 40
 - Número de trilhas por cilindro: 11
 - Número de cilindros: 1.331
 - Tamanho de cada um dos 20.000 registros: 256 bytes
- Cada setor, de 512 bytes, armazena dois registros (de 256 bytes cada)
 - Portanto, são necessários 10.000 setores
- Um cilindro tem 11 trilhas, sendo que cada uma tem 40 setores
 - Número de setores por cilindro: $11 * 40 = 440$ setores por cilindro
- Número de cilindros necessários: $10.000 / 440 = 22,7$ cilindros



- **Questão 1** Explique a diferença entre arquivo lógico e arquivo físico.
- **Questão 2** Descreva o que acontece quando um arquivo já existente é aberto por um aplicativo (p.ex., um programa) para escrita. E se o mesmo ocorrer com um arquivo não existente?
- **Questão 3** No que consiste a operação de posicionamento (*seeking*) em um arquivo? Qual a sua utilidade?
- **Questão 4** Explique o que é um cilindro, e a razão para a organização de arquivos em cilindros.
- **Questão 5** Por que os discos são considerados o gargalo de um sistema computacional? Explique como este problema pode ser minimizado.
- **Questão 6** O que são “buffers”de E/S (ou I/O)?
- **Questão 7** Descreva as operações fundamentais que podem ser realizadas em um arquivo. Descreva as funções que executam estas operações na linguagem de programação que você usa, e como elas são utilizadas. Por que existem vários modos de abrir um arquivo?



- Leitura recomendada: FOLK, M.J. File Structures, Addison-Wesley, 1992. Capítulos 3.
- **Atividade:** elabore 4 questões com as respectivas respostas sobre o tema visto na aula e referente aos capítulos do livro.
 - Essas questões devem ser entregues até dias 22/09 pelo google classroom
 - Pelo menos uma questão deve envolver SSD.
 - Obs:
 - Plágio é zero seja da internet ou colega
 - Questões tem que ser elaboradas. Não serão consideradas questões básicas. Ex. O que é...?
 - Pode ser de múltipla escolha, dissertativas etc



- FOLK, M.J. File Structures, Addison-Wesley, 1992.
- File Structures: Theory and Practice”, P. E. Livadas, Prentice-Hall, 1990;
- Contém material extraído e adaptado das notas de aula dos professores Moacir Ponti, Thiago Pardo, Leandro Cintra e Maria Cristina de Oliveira.