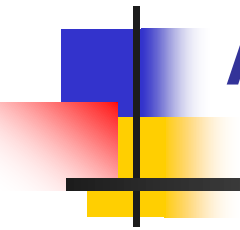
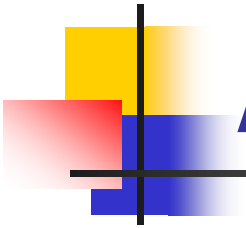
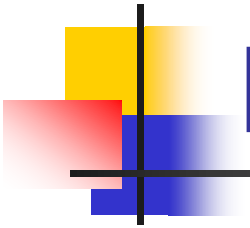


Armazenamento Secundário



Armazenamento secundário

- 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
  - Fixo



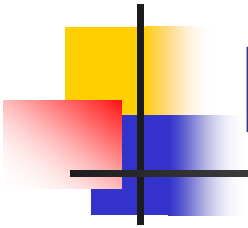
HDs

- HDs evoluíram quase tão radicalmente quanto os processadores
  - Primeiro HD tinha mentos 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!

HDs

- No início, mais em sistemas corporativos
  - 1.70m de altura e de comprimento, quase 1 tonelada
  - Chamado “unidade de disco”

IBM 350 (1956)

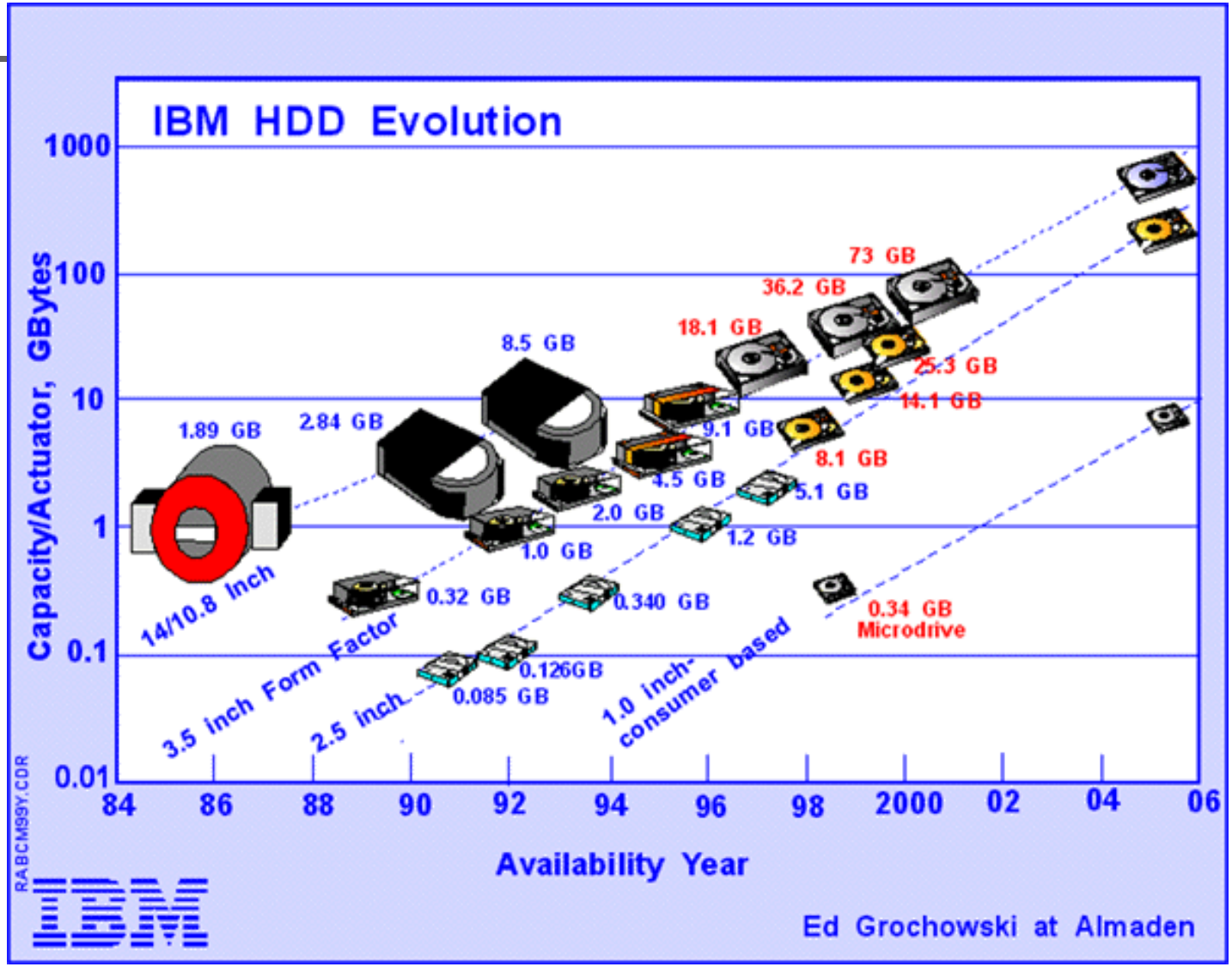


HDs

- 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"

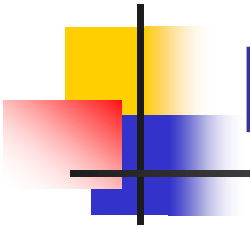
HDs

- Previsão em 2001



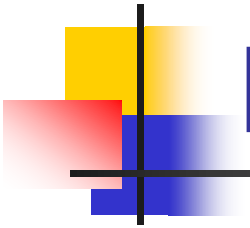
HDs

- HD
  - Em 1973, IBM lançou o que é considerado o pai dos HDs modernos
    - Winchester



HDs

- 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



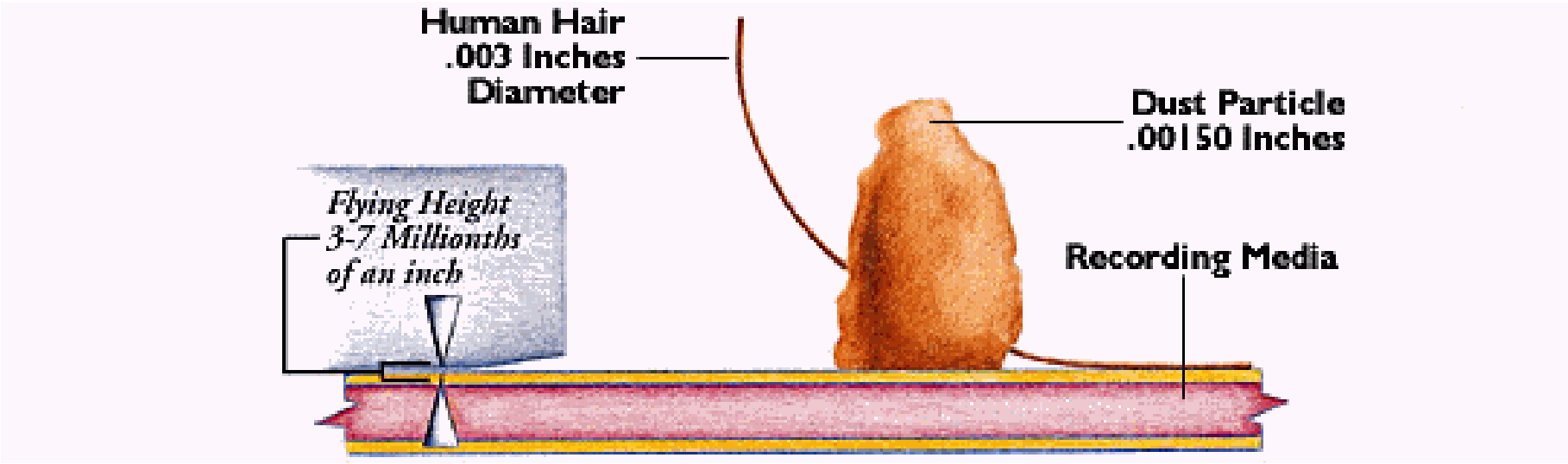
HDs

- 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


Discos
(platters)
Eixo
Cabeça de leitura
Braço de leitura
Actuator
Cabo flat
Jumpers
Porta SATA
Conector de força

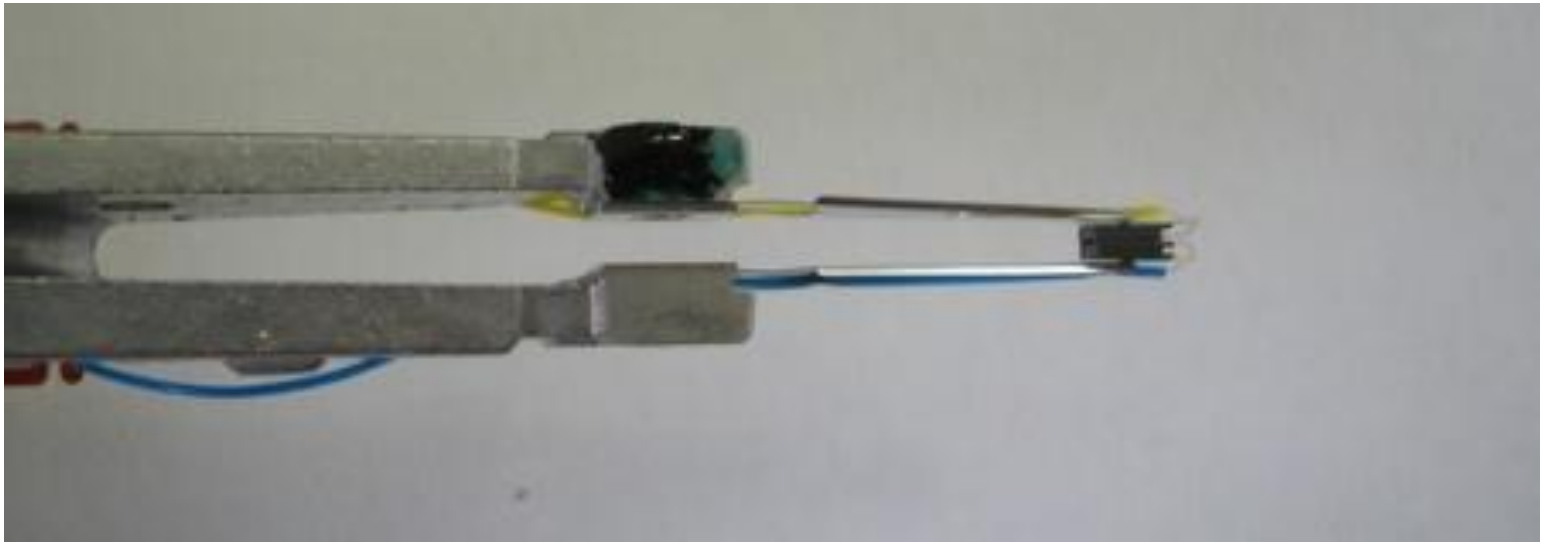
Cabeças de leitura

- Distância em relação ao disco
  - Flutuam em função do colchão de ar gerado quando o disco gira
  - Não há vácuo dentro do disco!



Cabeças de leitura

- Sem o disco

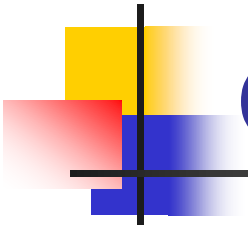


HDs

- Por que os HDs são sempre exibidos meio de lado?

HDs

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



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

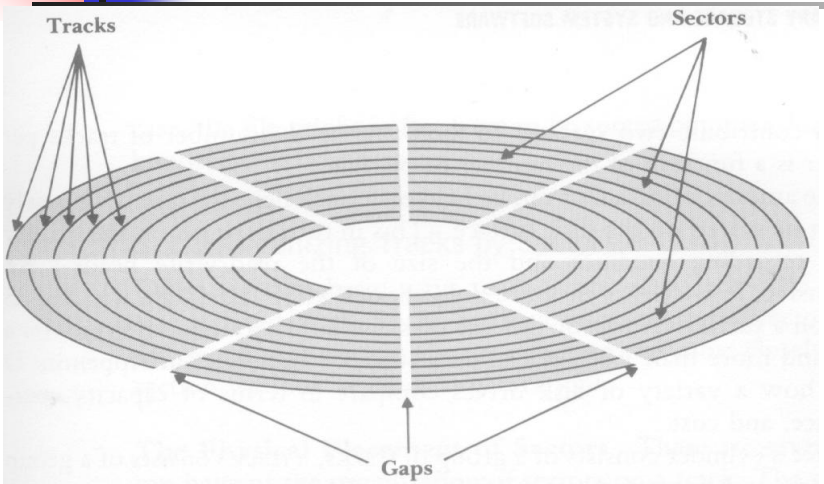


FIGURE 3.2 Surface of disk showing tracks and sectors.

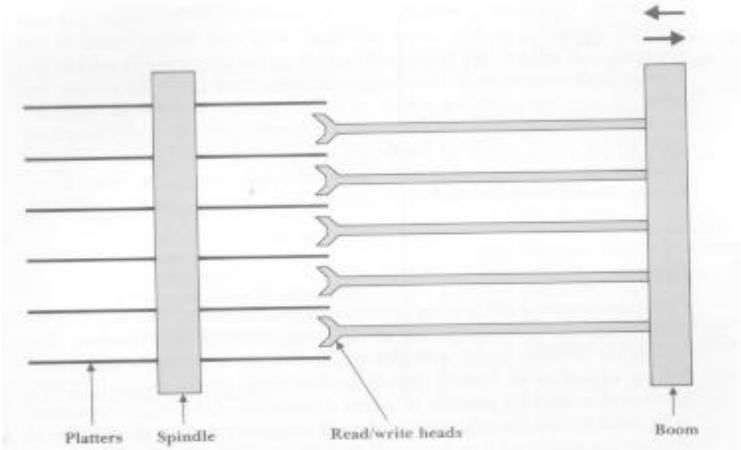
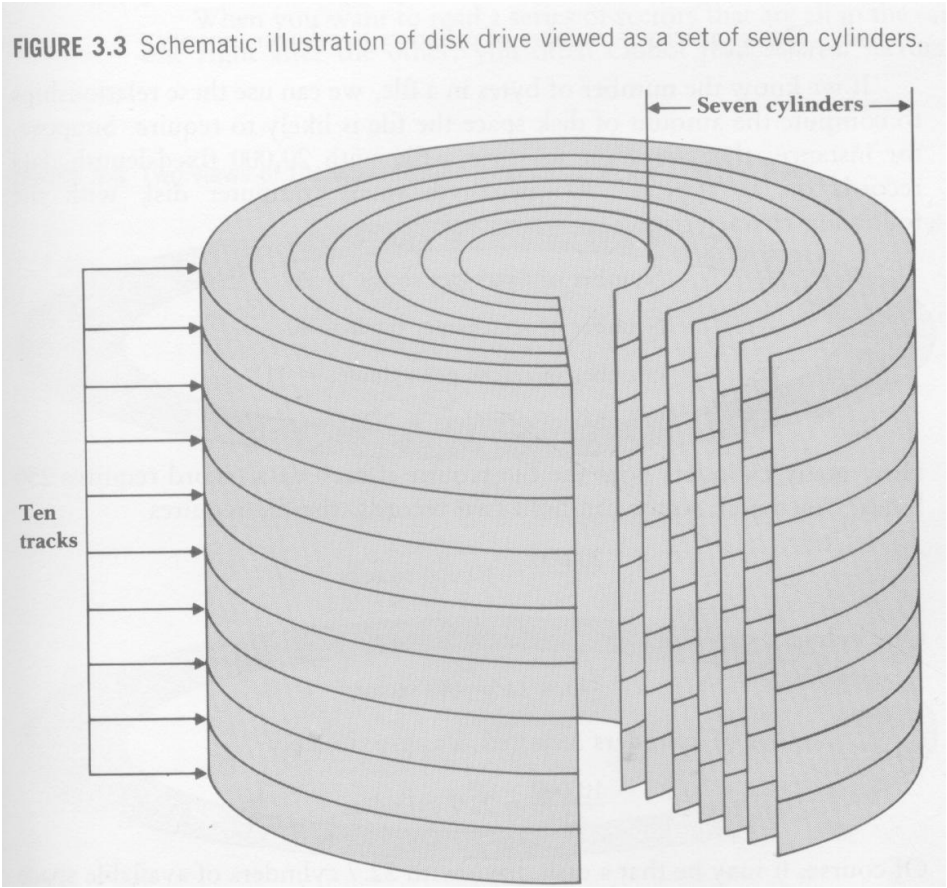
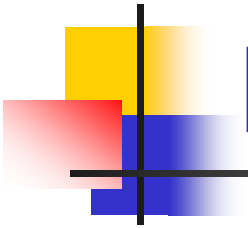


FIGURE 3.1 Schematic illustration of disk drive.

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

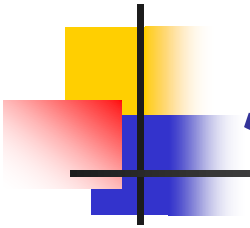




Endereços no disco

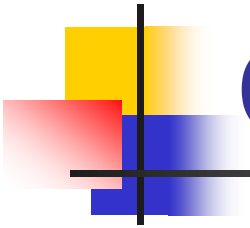
- Um **setor** é a menor porção endereçável do disco

- 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.



Seeking

- 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 ao mínimo



Cluster

- 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

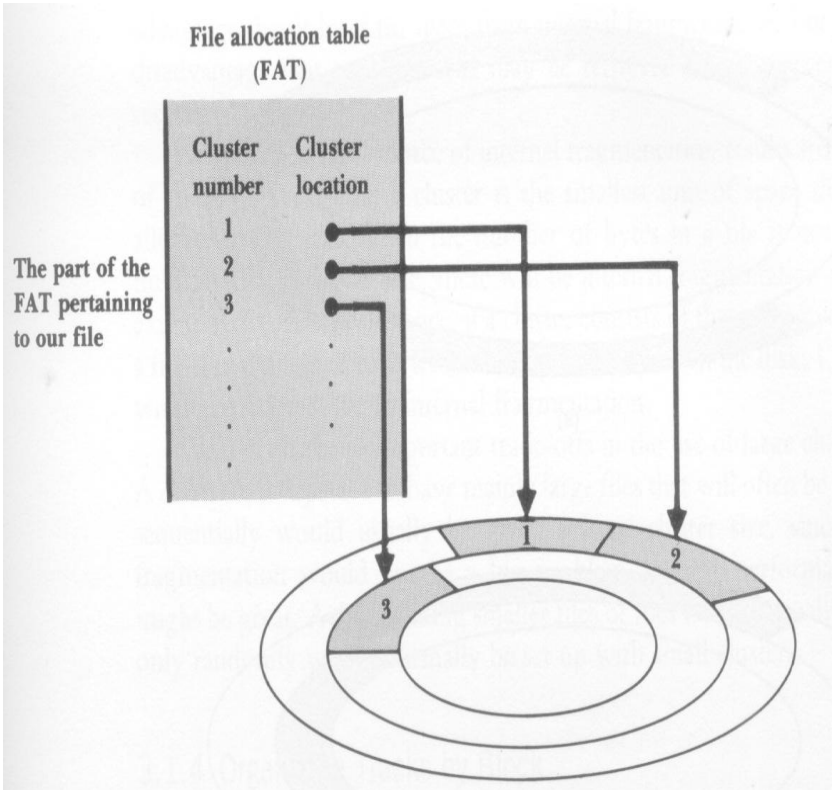
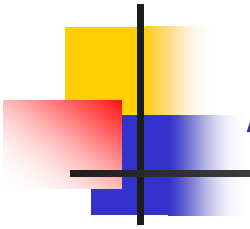


FIGURE 3.5 The file manager determines which cluster in the file has the sector that is to be accessed.



Extent

- 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

Extent

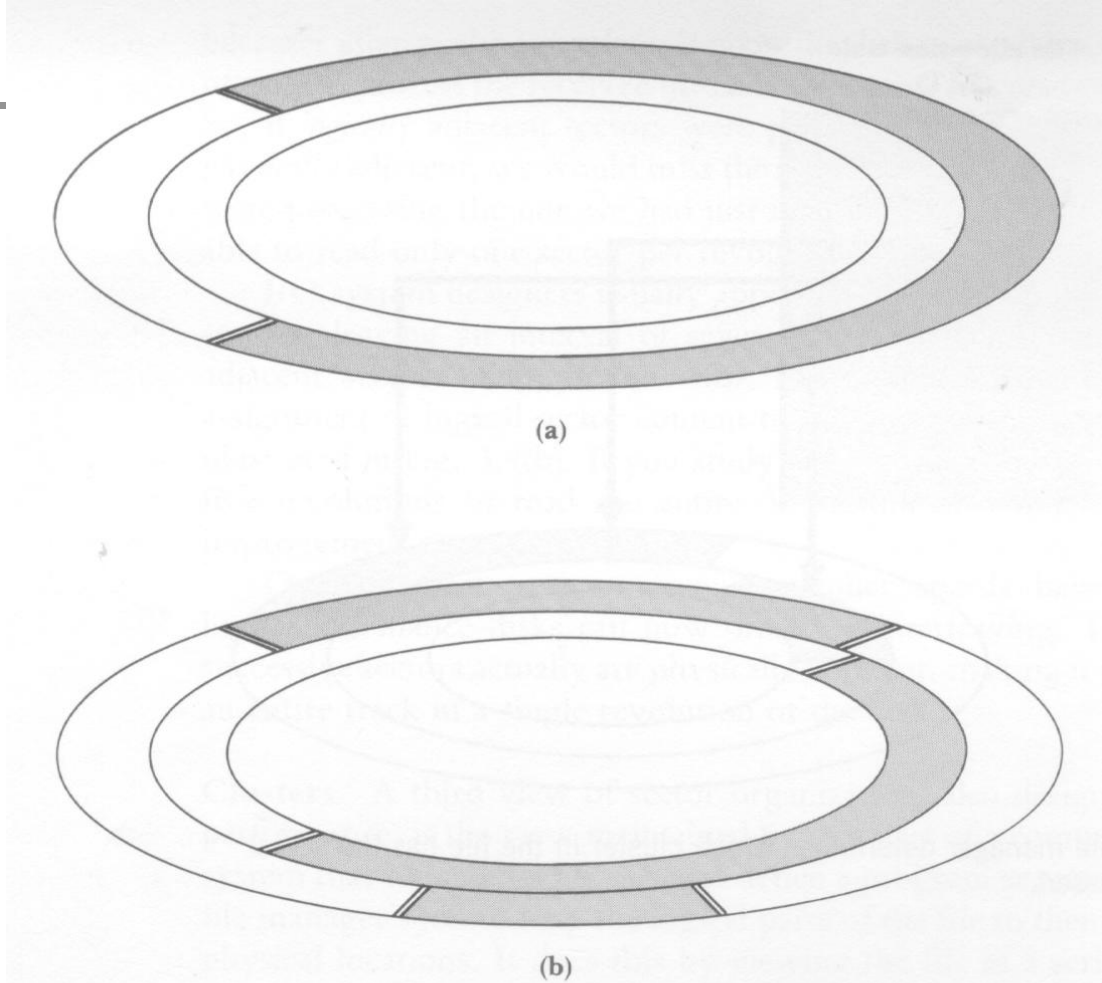
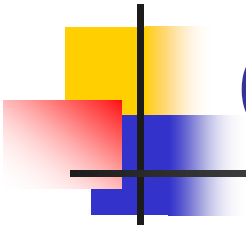
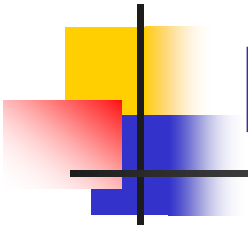


FIGURE 3.6 File extents (shaded area represents space on disk used by a single file).



Capacidade do disco (nominal)

- Capacidade do setor
  - nº bytes (Ex. 512 bytes)
- Capacidade da trilha
  - nº de setores/trilha * capacidade do setor
- Capacidade do cilindro
  - nº de trilhas/cilindro * capacidade da trilha
- Capacidade do disco
  - nº de cilindros x capacidade do cilindro



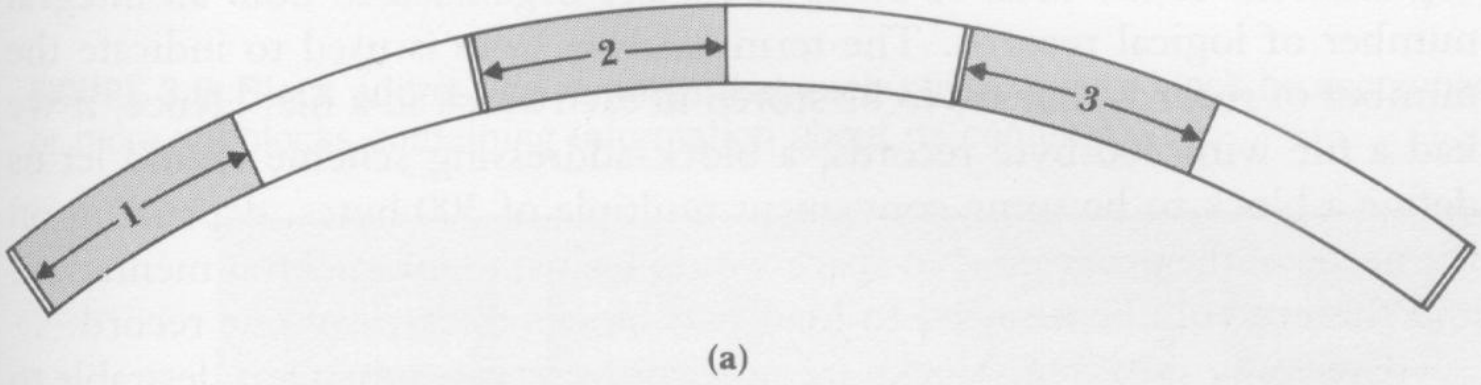
Fragmentação interna

- Perda de espaço útil decorrente da organização em setores de tamanho fixo

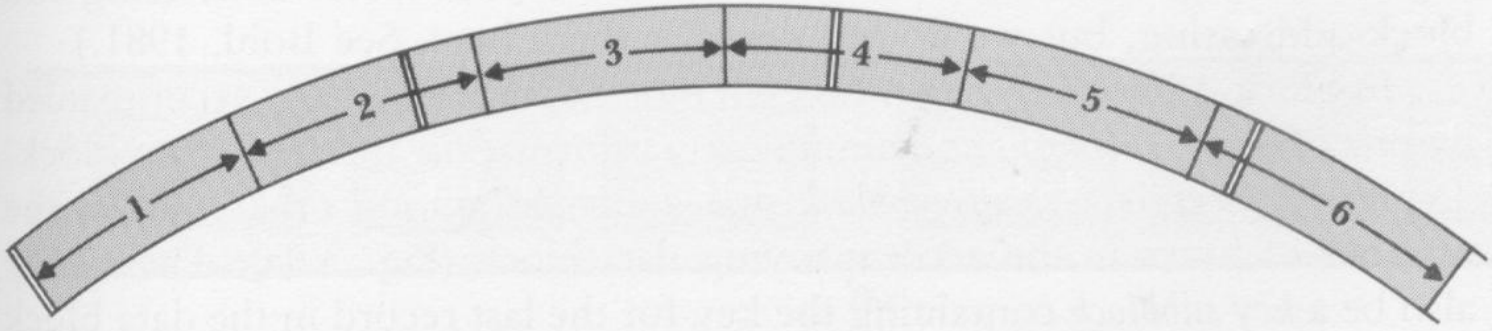
- Ex: setor de 512 bytes, arquivos c/ registro de 300 bytes. Temos duas alternativas:
  a. 1 registro por setor => fragmentação
  b. Registros ocupando mais de 1 setor => acesso mais complexo

Fragmentação interna

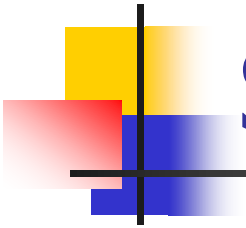
FIGURE 3.7 Alternate record organization within sectors (shaded areas represent data records, and unshaded areas represent unused space).



(a)

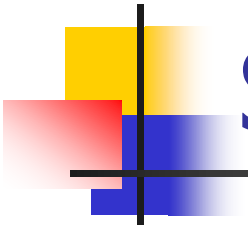


(b)



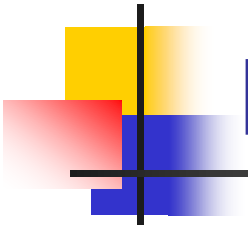
Sistema de Arquivos

- 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



Sistema de Arquivos

- 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

Setores X Blocos

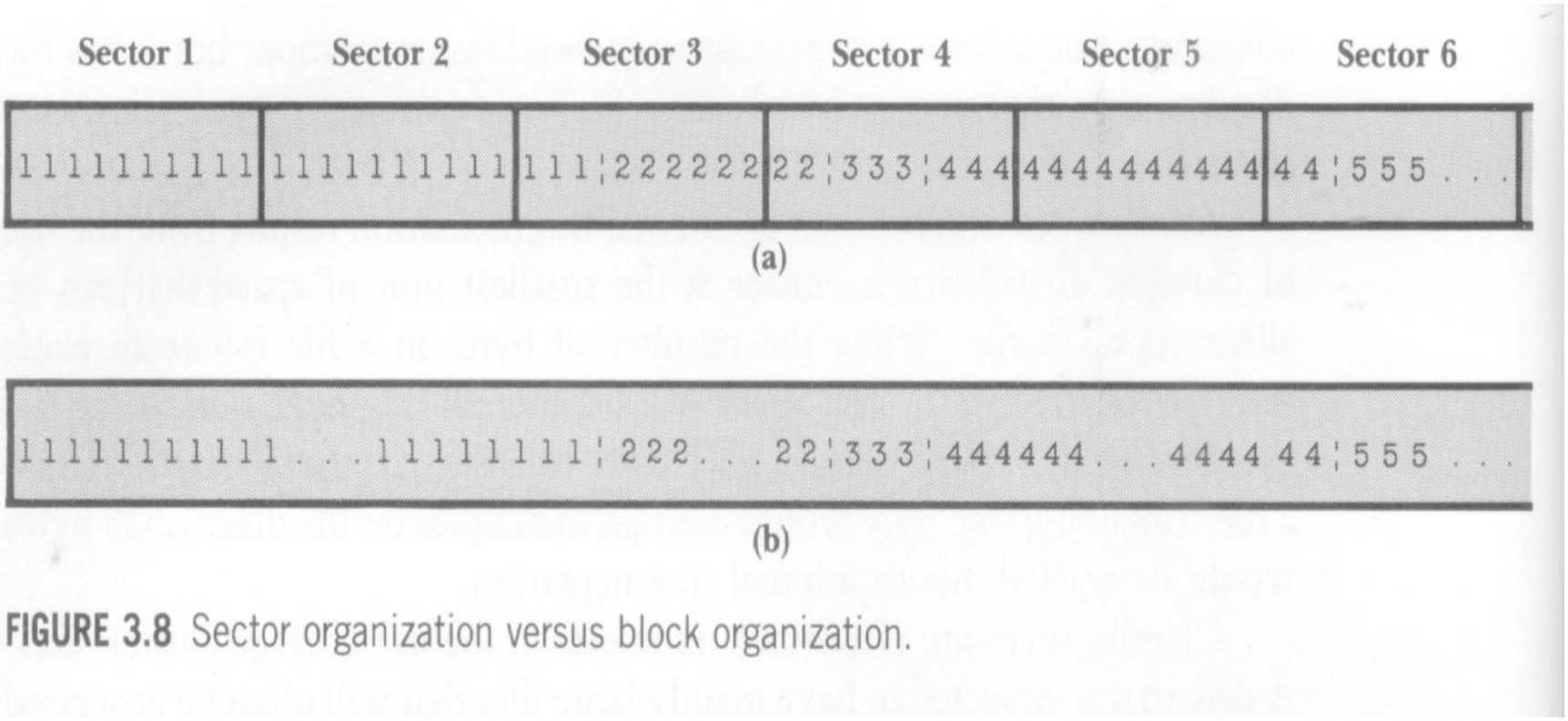
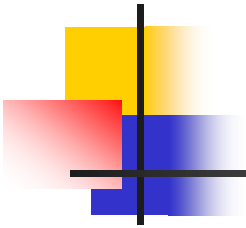


FIGURE 3.8 Sector organization versus block organization.

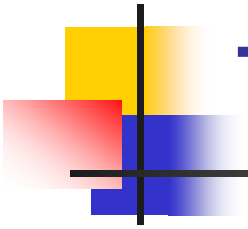
Qual a dificuldade?



Overhead

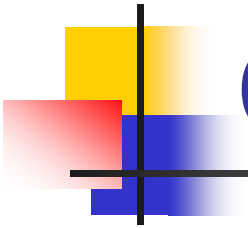
Overhead – espaço ocupado com informações para gerenciamento (não c/ dados), introduzidas pelo processo de formatação do disco

- O *overhead* existe tanto em discos organizados por setor quanto em discos organizados por blocos



Tamanho do *cluster*

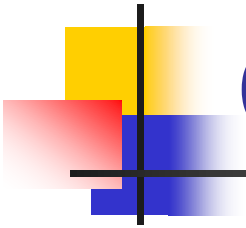
- Definido automaticamente pelo SO quando o disco é formatado
- FAT (Windows): sempre uma potência de 2
  - 2, 4, 8, 16 ou 32KB
- Determinado pelo máximo que a FAT consegue manipular, e pelo tamanho do disco
  - FAT16: pode endereçar 2^{16} clusters = 65.536
- Quanto maior o cluster, maior a fragmentação!



Outros sistemas de arquivos

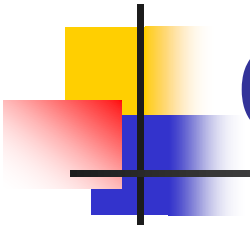
- FAT32 (Windows 95 e posteriores)
  - *clusters* de tamanho menor, endereça mais *clusters*, menos fragmentação

- 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



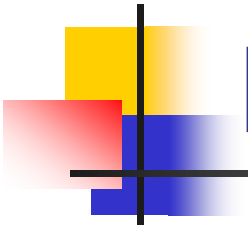
Custo de acesso a disco

- 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**



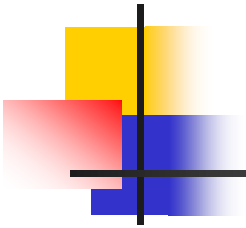
Observação

- 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)



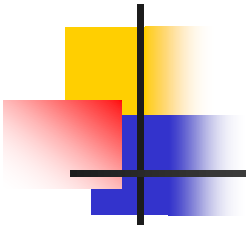
Exercício

- 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?



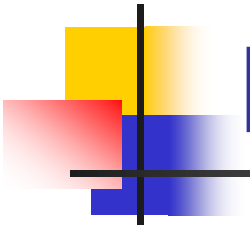
Exercício

- Dados
  - 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



Para discussão

- Você está projetando seu próprio HD e decide armazenar os arquivos em espaços contínuos, ignorando limites de setores/clusters/extents/blocos/cilindros
  - Como isso facilita o armazenamento e recuperação de dados?
  - Que problemas cria?



Leituras adicionais

- http://www.clubedohardware.com.br/artigos/313
- http://www.clubedohardware.com.br/artigos/489
- http://www.pcguide.com/ref/hdd/
- http://www.guiadohardware.net/tutoriais/como-hd-funciona/