

ARMAZENAMENTO SECUNDÁRIO, PARTE 1

Professora Rosane Minghim

Organização da informação no disco

2

- **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

3

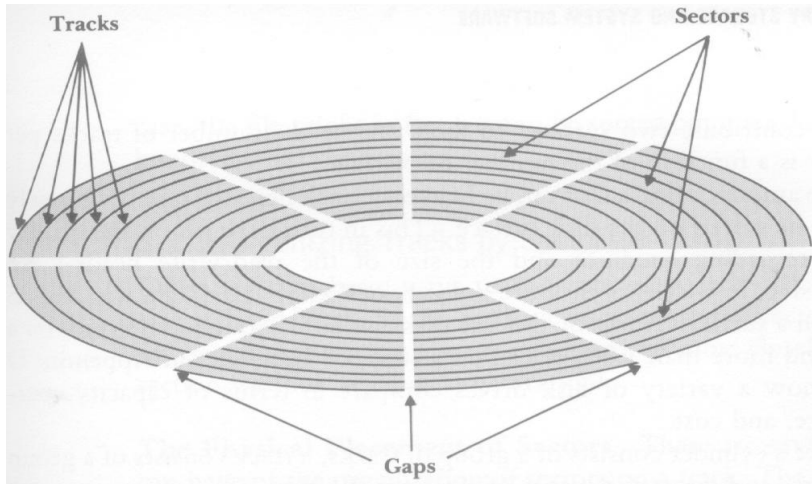


FIGURE 3.2 Surface of disk showing tracks and sectors.

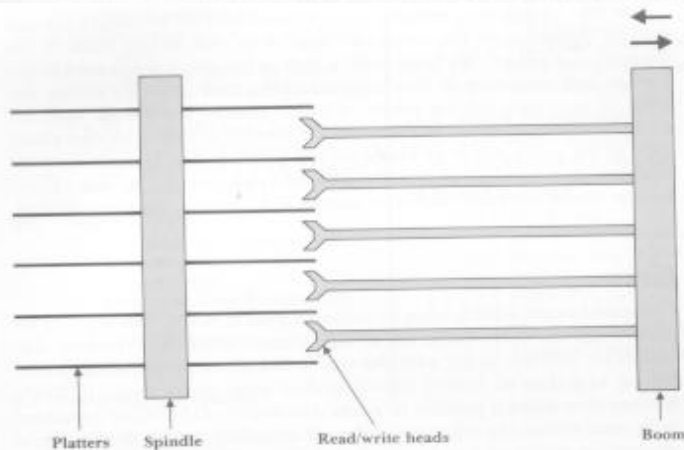


FIGURE 3.1 Schematic illustration of disk drive.

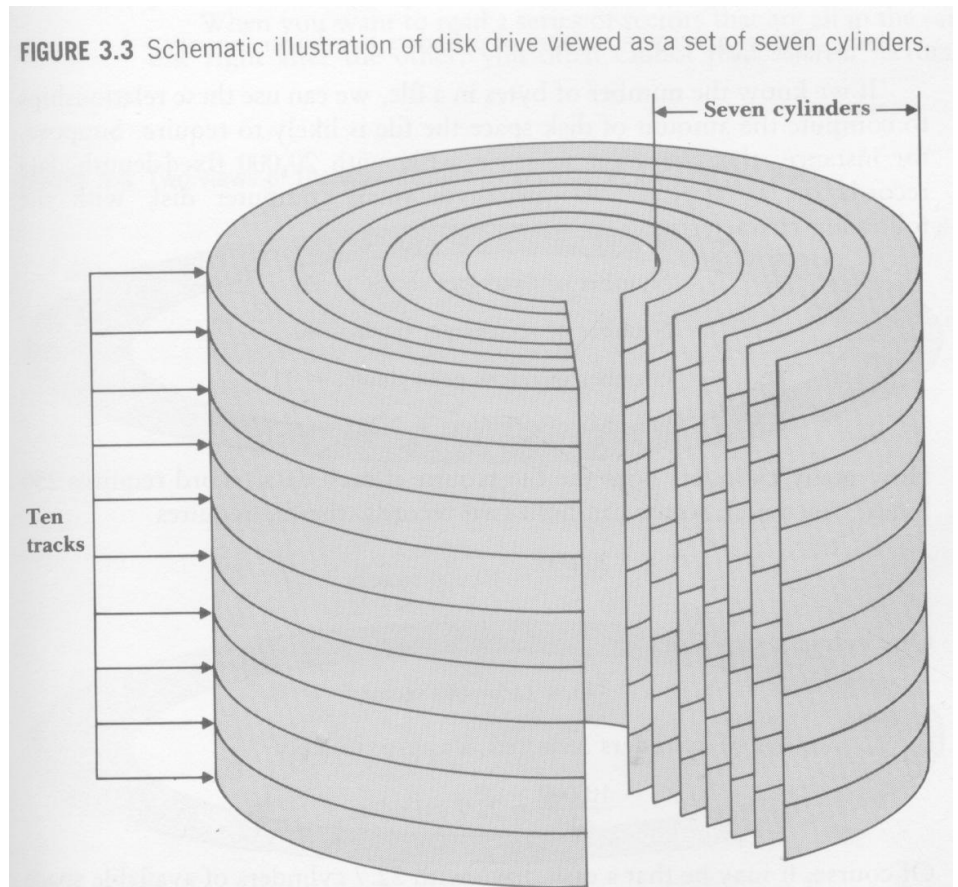


FIGURE 3.3 Schematic illustration of disk drive viewed as a set of seven cylinders.

Endereços no disco

4

- 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

5

- 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

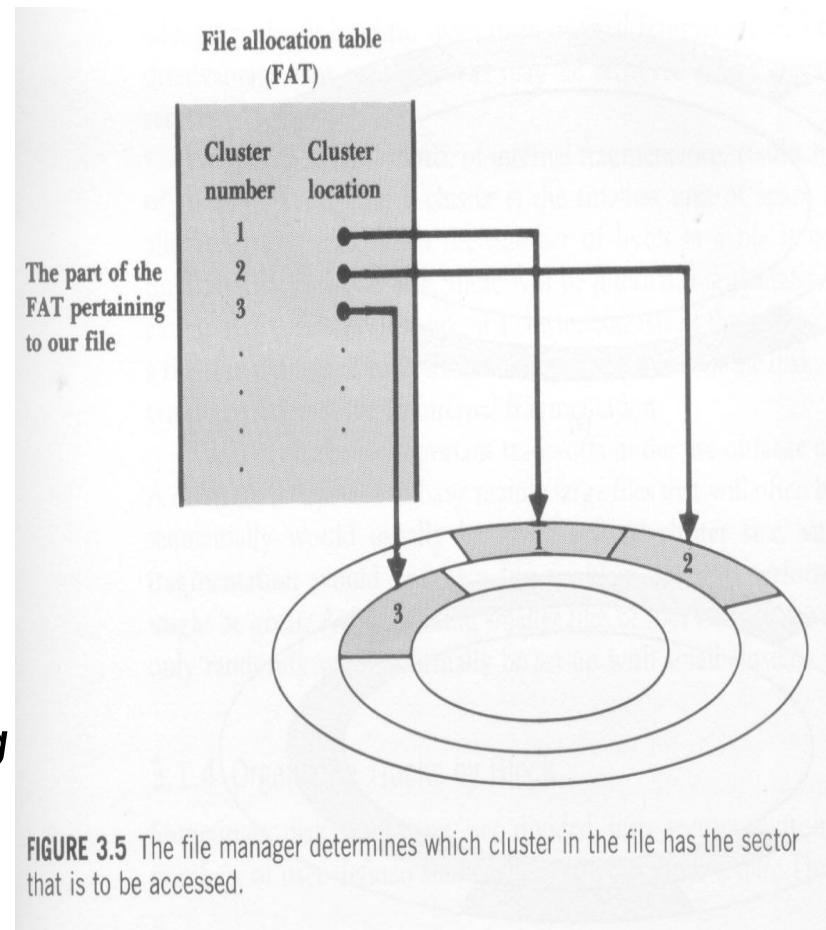
6

- 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*

7

- 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



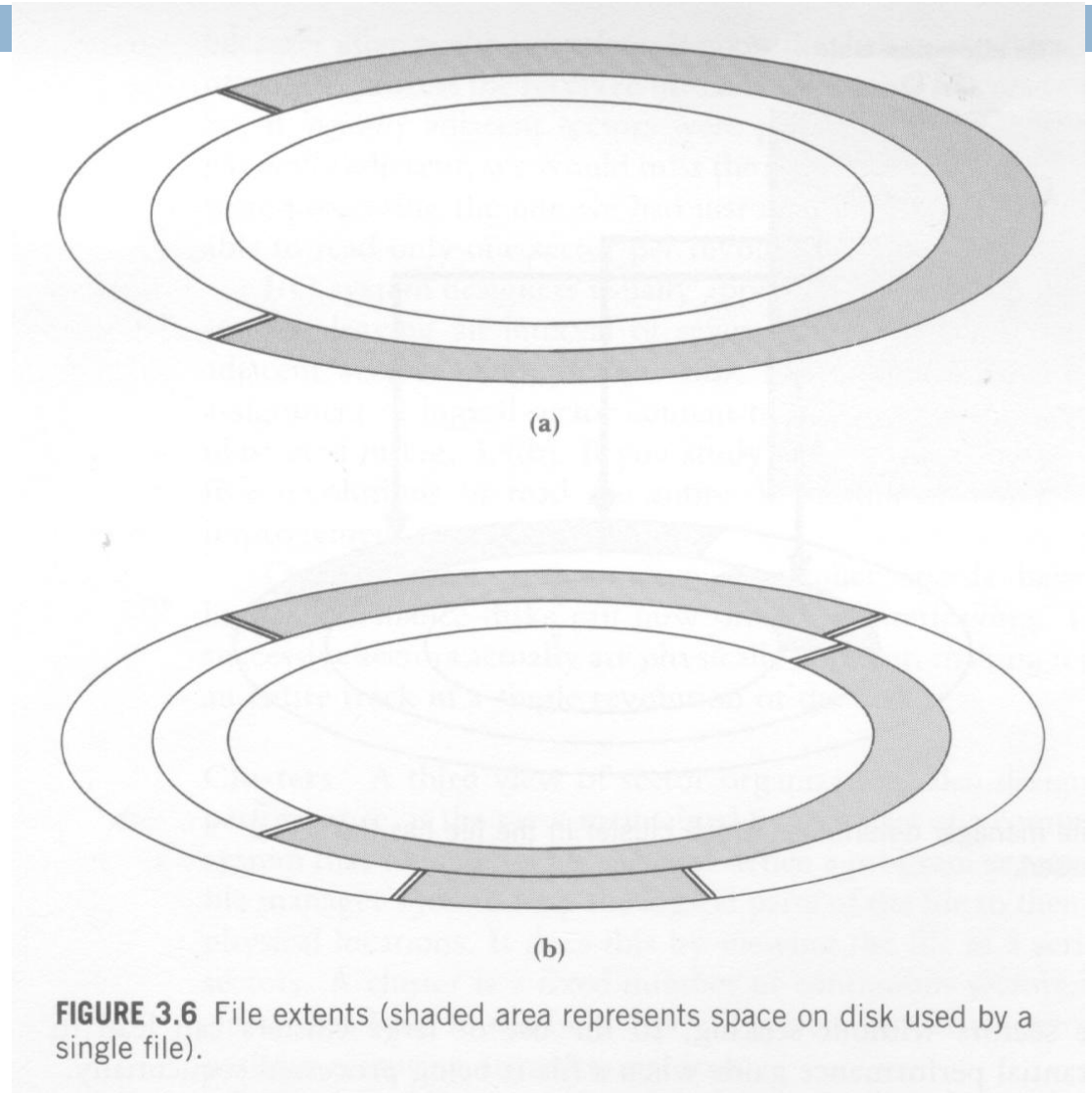
Extent

8

- Seqüência 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*
 - ▣ freqüentemente isso não é possível, e o arquivo é espalhado em vários *extents* pelo disco

Extent

9



Capacidade do disco (nominal)

10

- 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

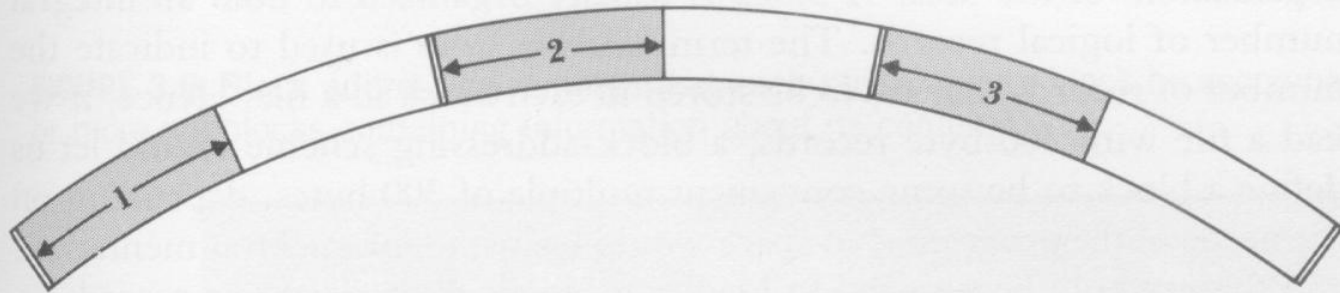
11

- 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

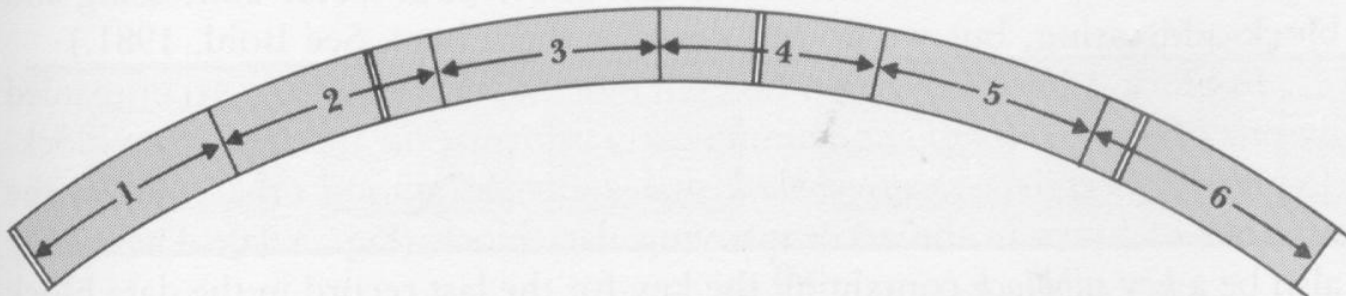
Fragmentação interna

12

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



(a)



(b)

Sistema de Arquivos

13

- 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

14

- 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)

15

- 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

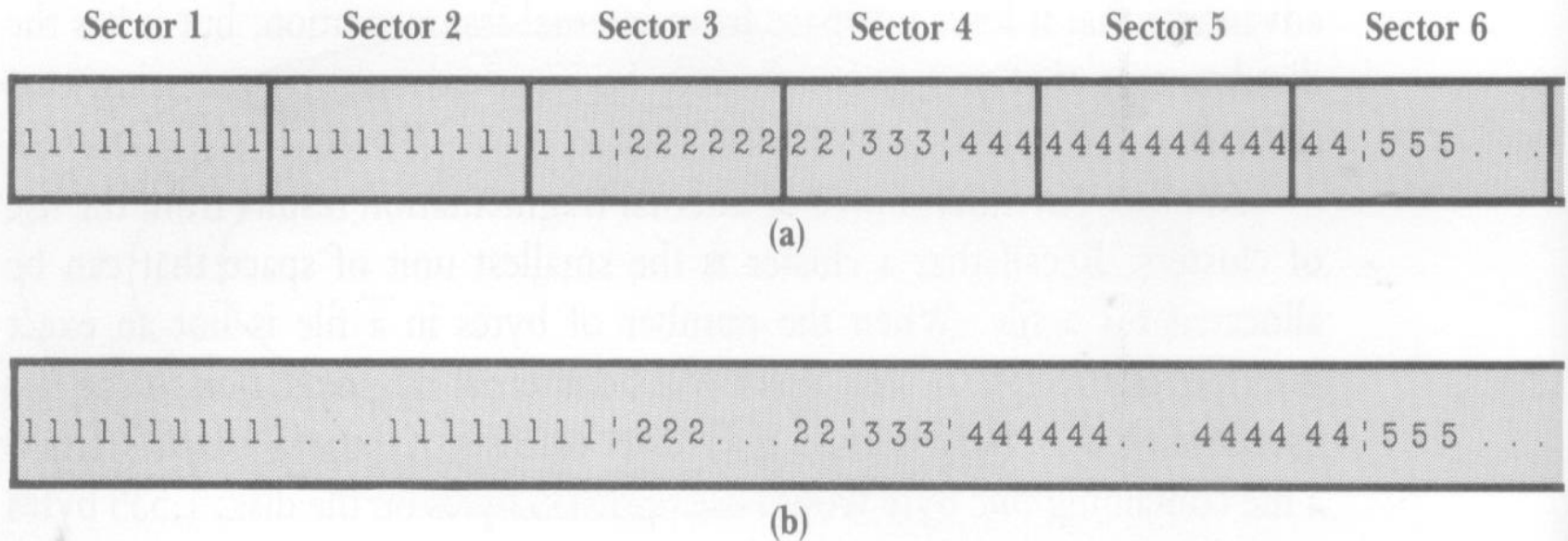


FIGURE 3.8 Sector organization versus block organization.

Overhead

17

- 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*

18

- 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

19

- 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

Mais Informações

20

□ Sobre sistemas de arquivos p/ Windows ver:

<http://www.clubedohardware.com.br/artigos/313>

<http://www.clubedohardware.com.br/artigos/489>

Custo de acesso a disco

21

- É uma combinação de 3 fatores:
 - ▣ **Tempo de busca (seek):** tempo p/ posicionar o braço de acesso no cilindro correto
 - ▣ **Delay de rotação:** tempo p/ 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=(n° de bytes transferidos/n° de bytes por trilha)*tempo de rotação**

Observação

22

- Ver <http://www.pcguide.com/ref/hdd/>
- 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 seqüencial)

FIM