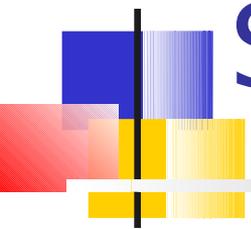
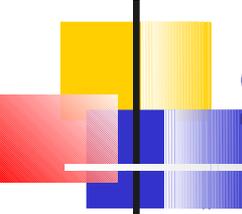


Armazenamento Secundário



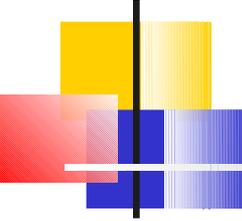
SCC-503 Algoritmos e Estruturas de Dados II

Thiago A. S. Pardo
Leandro C. Cintra
M.C.F. de Oliveira



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

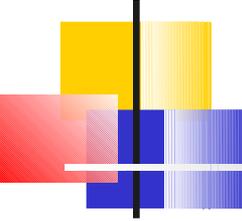
- HDs evoluíram quase tão radicalmente quanto os processadores
 - Primeiro HD tinha menos de 5Mb, por U\$35.000
 - HDs melhores tinham cerca de 10Mb, mais de U\$100 por Mb
 - Atualmente menos de 1 centavo por Mb
 - IBM teve um papel importante!

HDS

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



IBM 350 (1956)

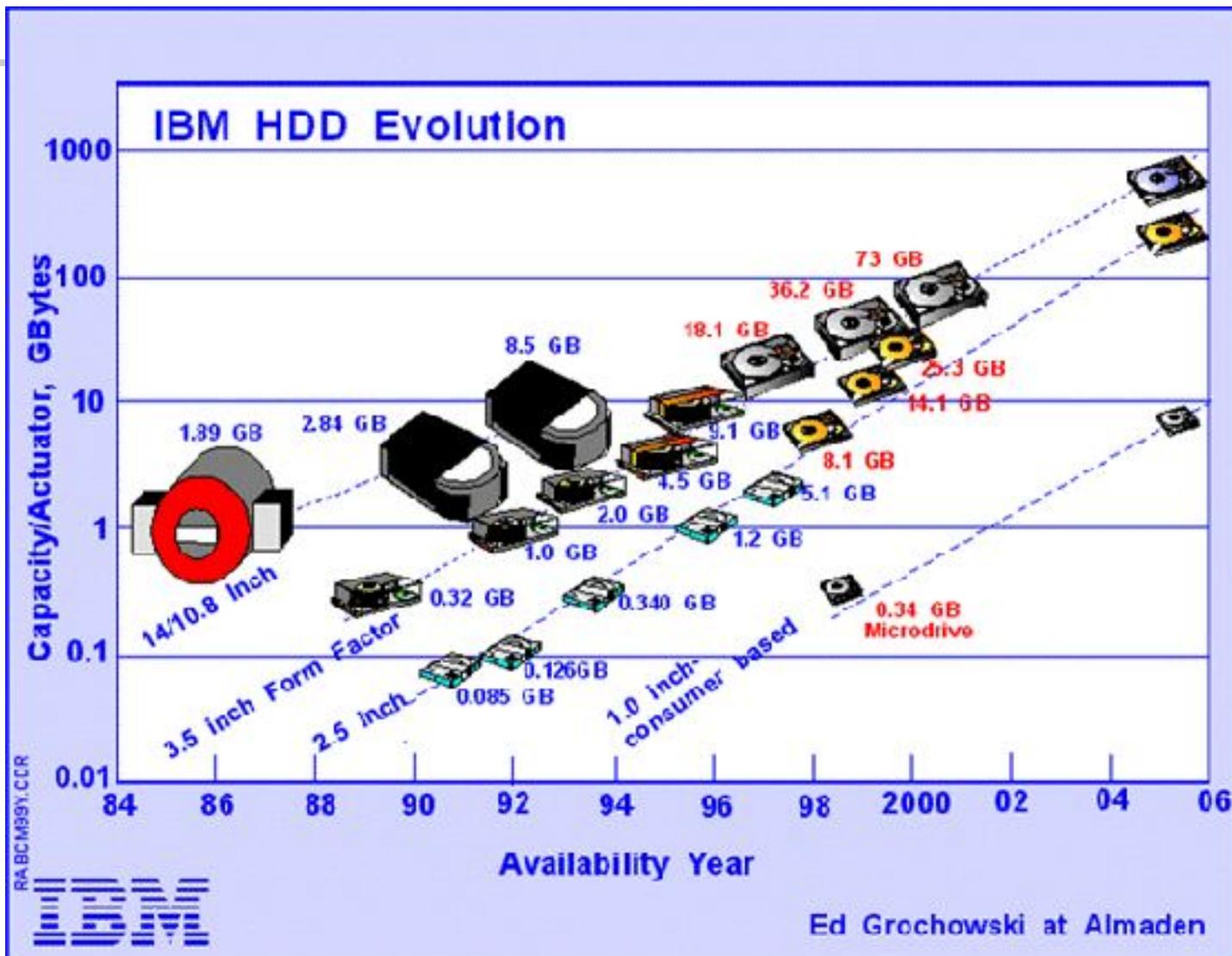


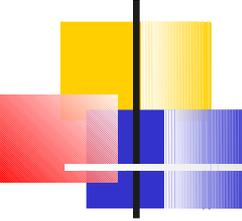
HDS

- 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”

HDDs

- Previsão em 2001
- Como estamos?



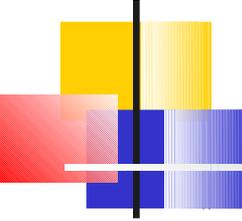


HDs

- HD

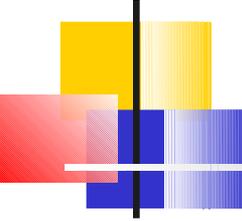
- Em 1973, IBM lançou o **Winchester**, que é considerado o pai dos HDs modernos
 - Winchester





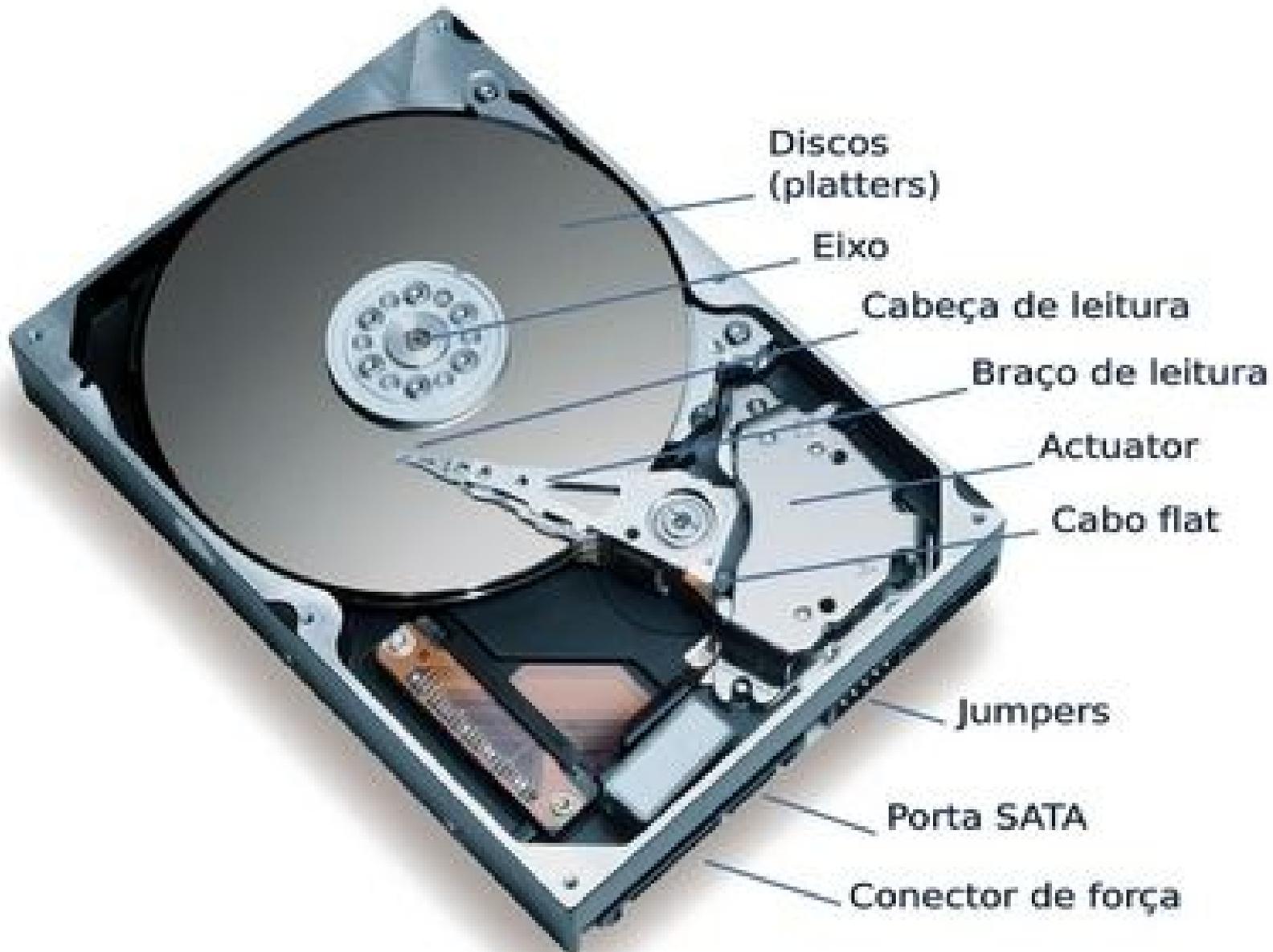
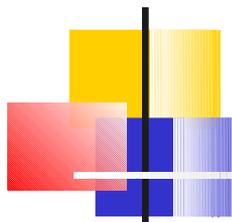
HDS

- 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



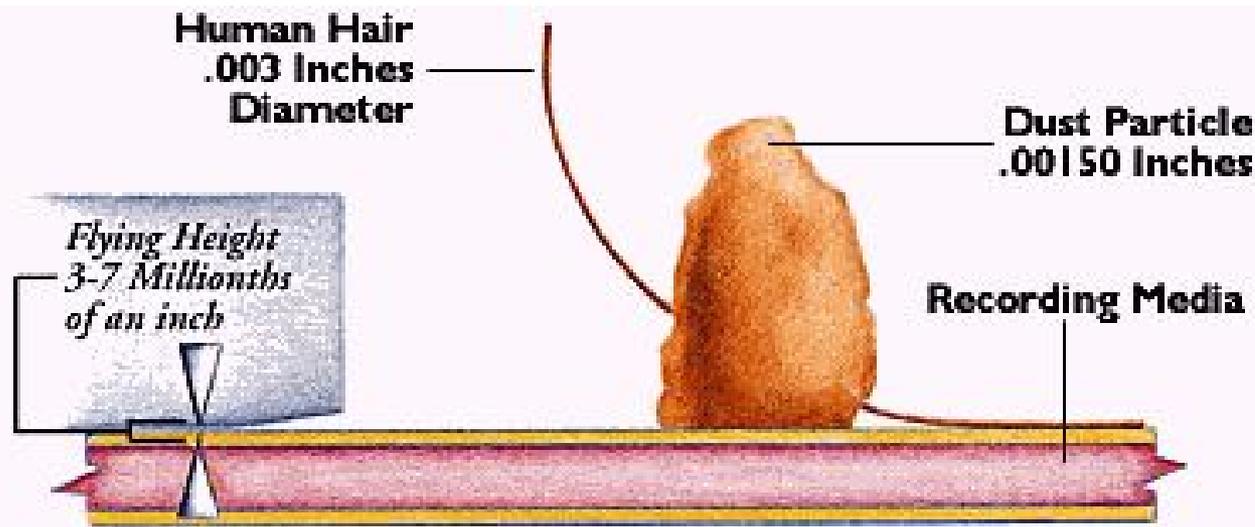
HDS

- **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 (~15.000 RPM atualmente)
 - **Convertem entre sinais elétricos e pulsos magnéticos**
 - A informação lida é armazenada em um **buffer**, de onde é transferida para a memória



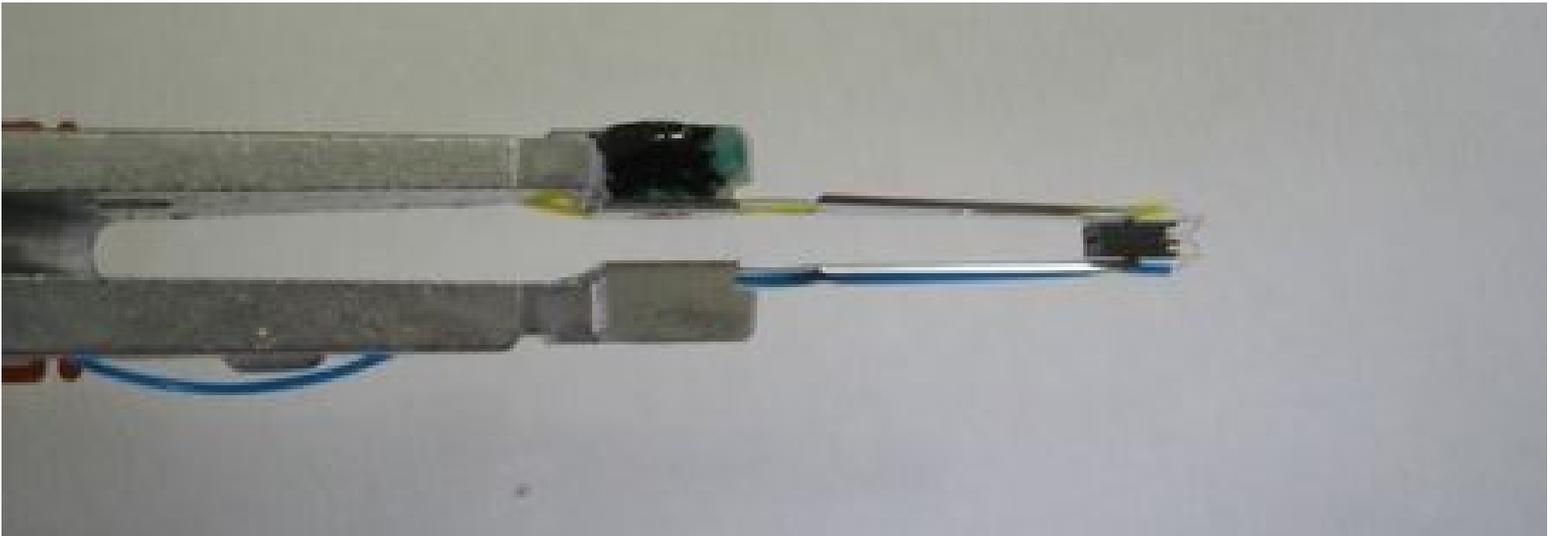
Cabeças de leitura

- 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



Interior do HD

- Entrada de ar com filtro para manter a pressão interna igual à externa



HDS

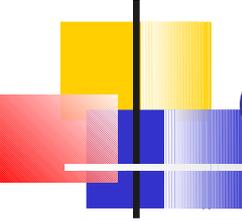
- HDs funcionam em altitudes altíssimas (mais de 3.000m)?



HDS

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





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

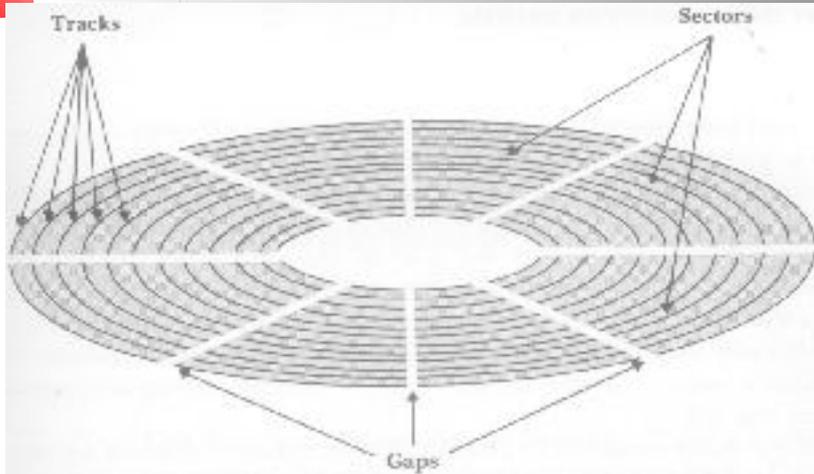


FIGURE 3.2 Surface of disk showing tracks and sectors.

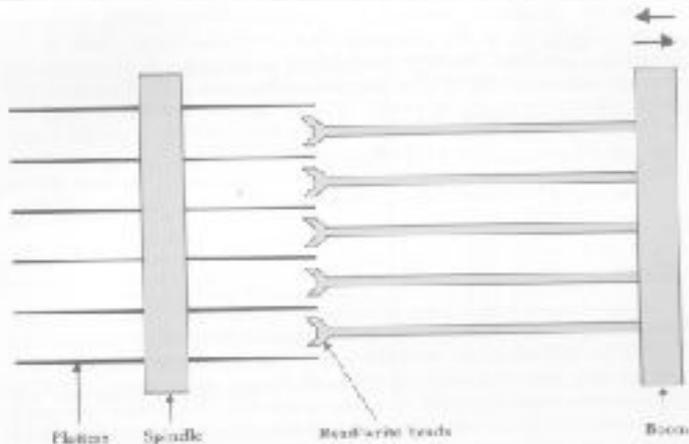
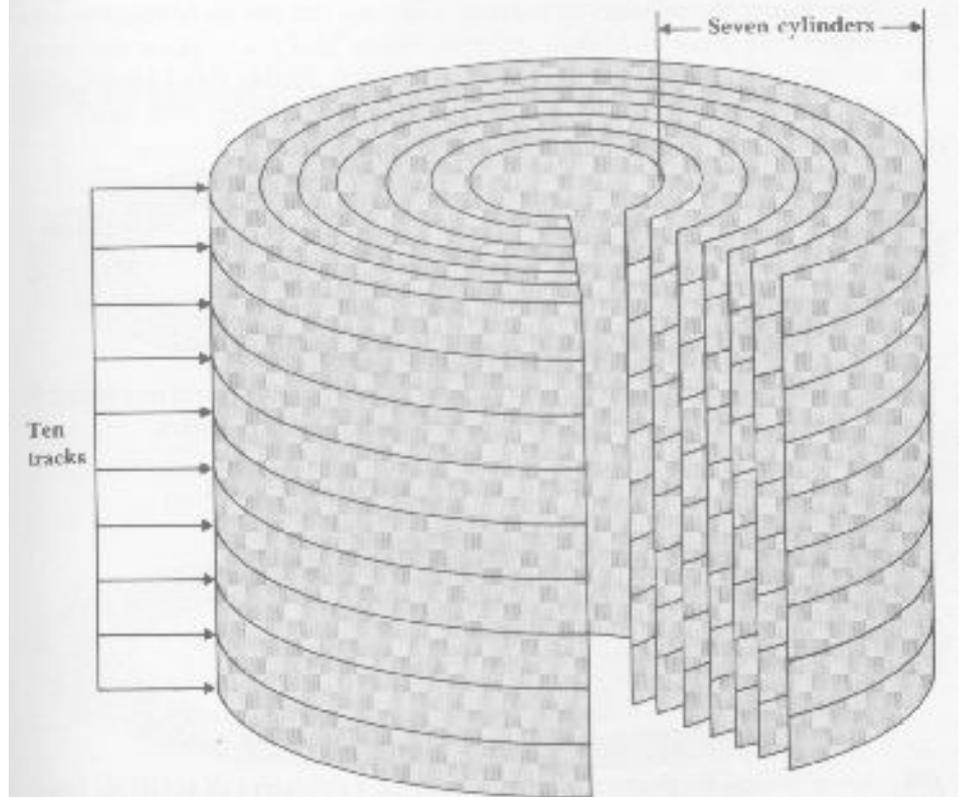
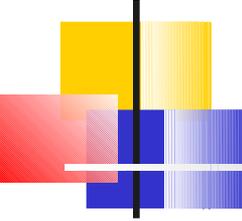


FIGURE 3.3 Schematic illustration of disk drive.

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

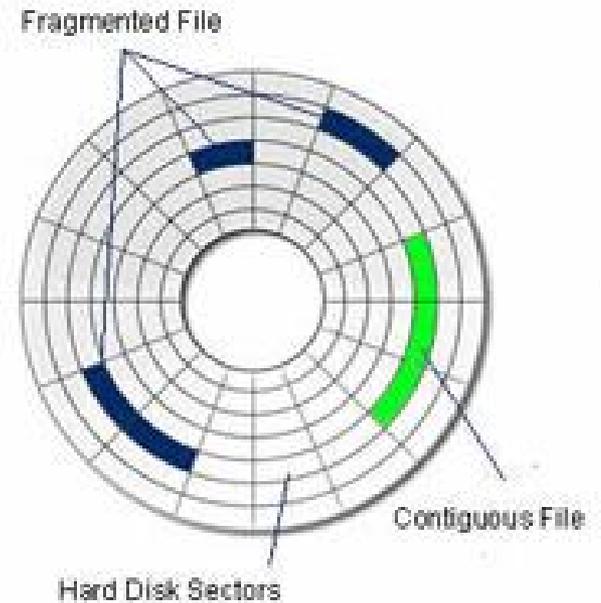




Endereços no disco

- Um **setor** é a **menor porção endereçável** do disco
- Ao ler 1 byte
 - SO 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

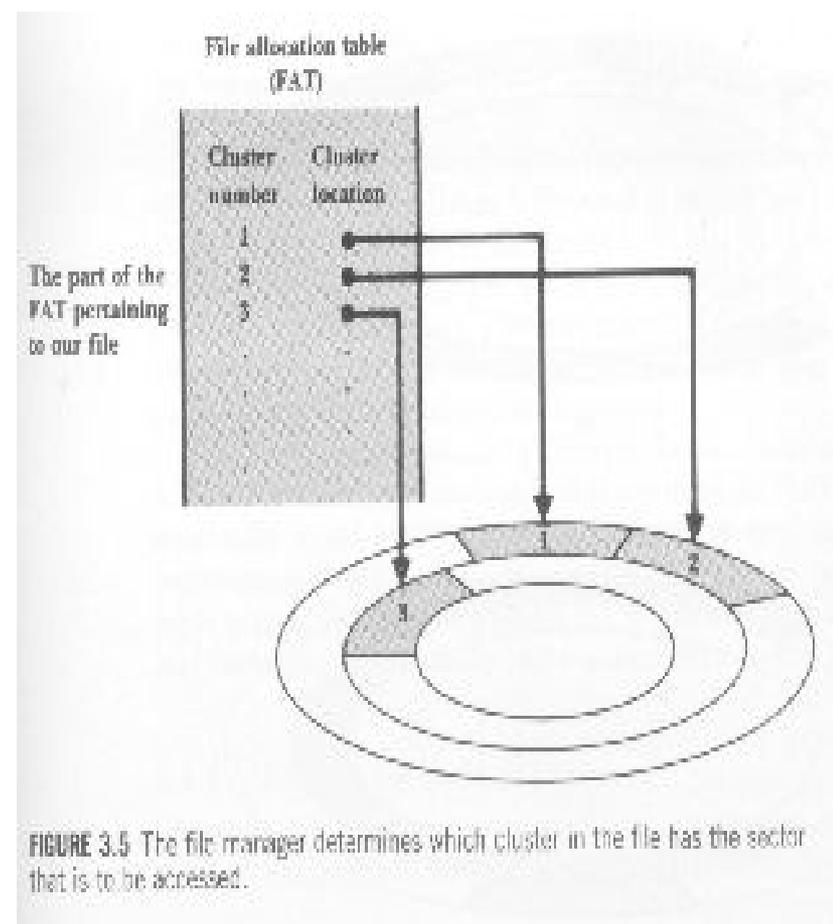
Cluster

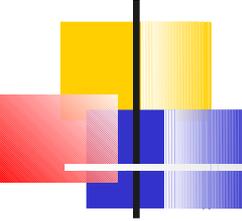


- **Conjunto de setores**
logicamente contíguos no disco
- Um arquivo é visto pelo SO como um grupo de clusters distribuído no disco
 - Arquivos são alocados em um ou mais clusters

FAT – *File Allocation Table*

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

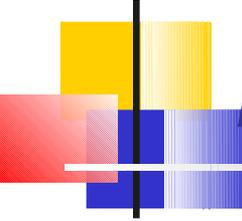




FAT – *File Allocation Table*

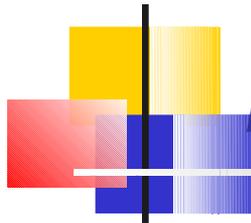
- Questão

- O que acontece se dá pau na FAT?



Extent

- 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

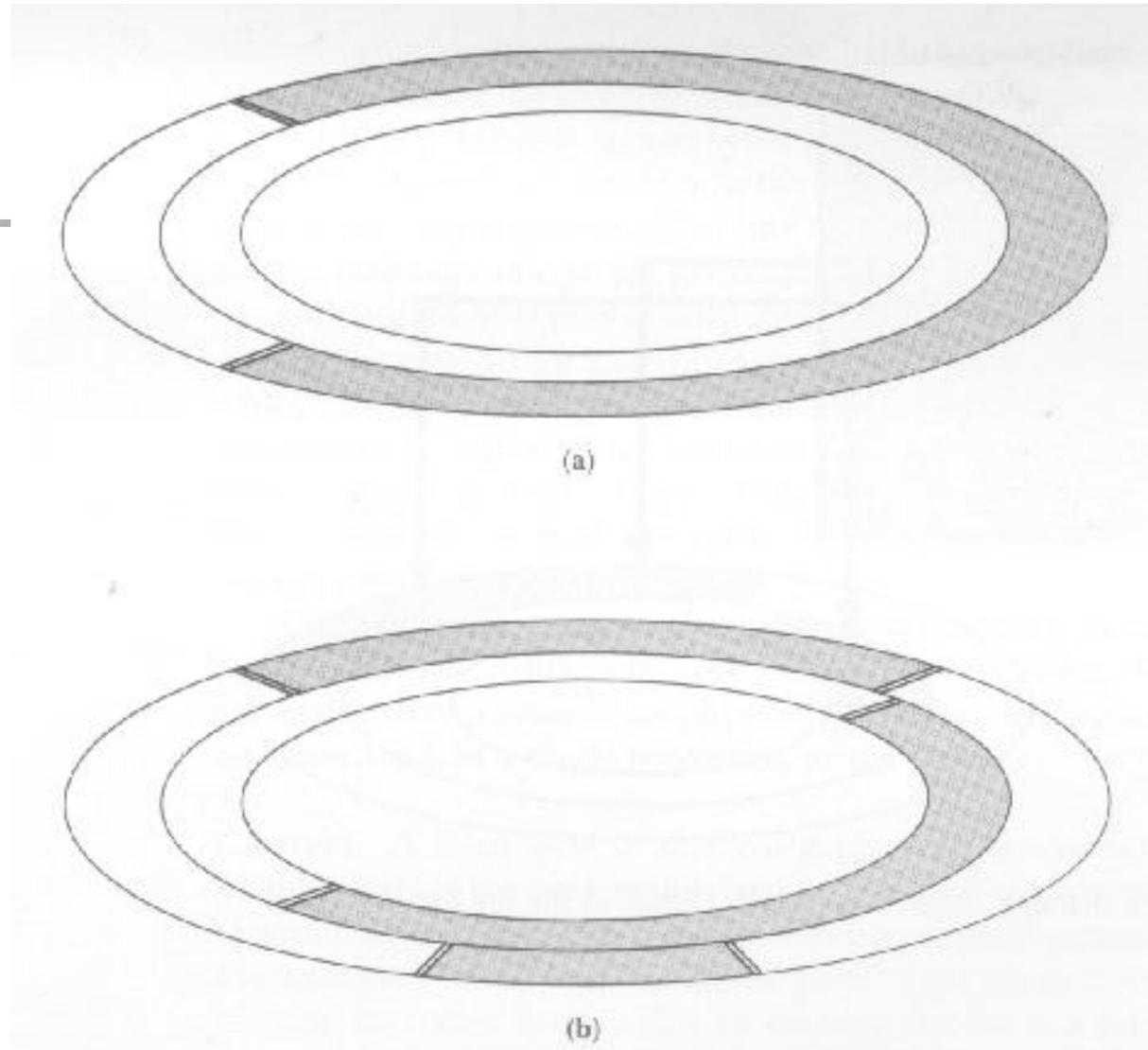
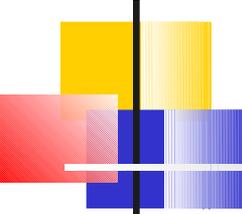
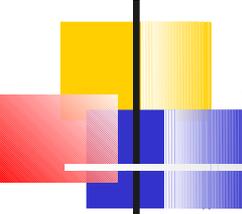


FIGURE 3.6 File extents (shaded area represents space on disk used by a single file).



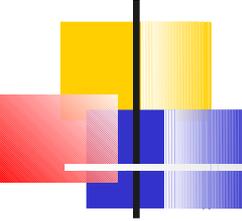
Capacidade do disco (nominal)

- Capacidade do setor
 - nº bytes (Ex. 512 bytes)
- Capacidade da trilha
- Capacidade do cilindro
- Capacidade do disco



Capacidade do disco (nominal)

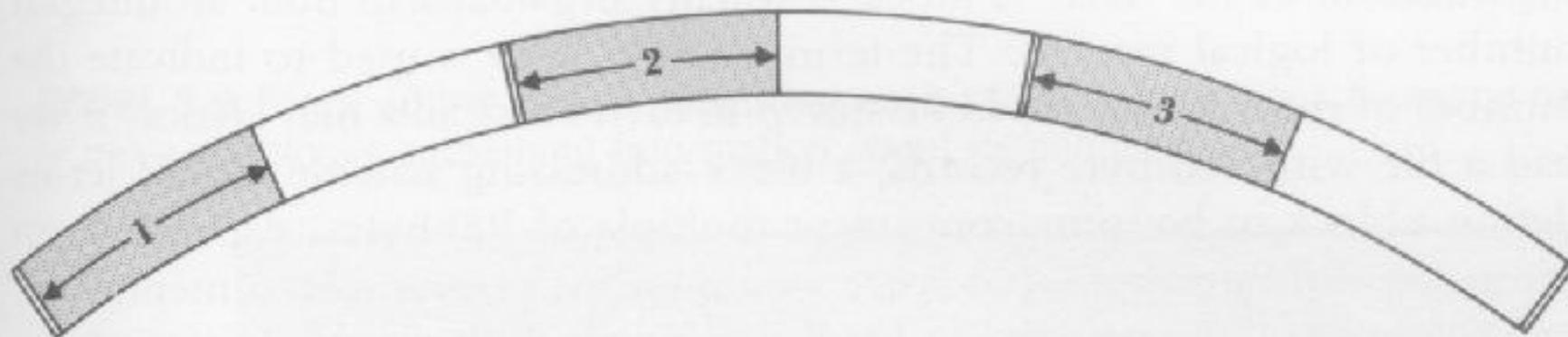
- **Capacidade do setor**
 - nº bytes (Ex. 512 bytes)
- **Capacidade da trilha**
 - nº de setores por trilha * capacidade do setor
- **Capacidade do cilindro**
 - nº de trilhas por cilindro * capacidade da trilha
- **Capacidade do disco**
 - nº de cilindros x capacidade do cilindro



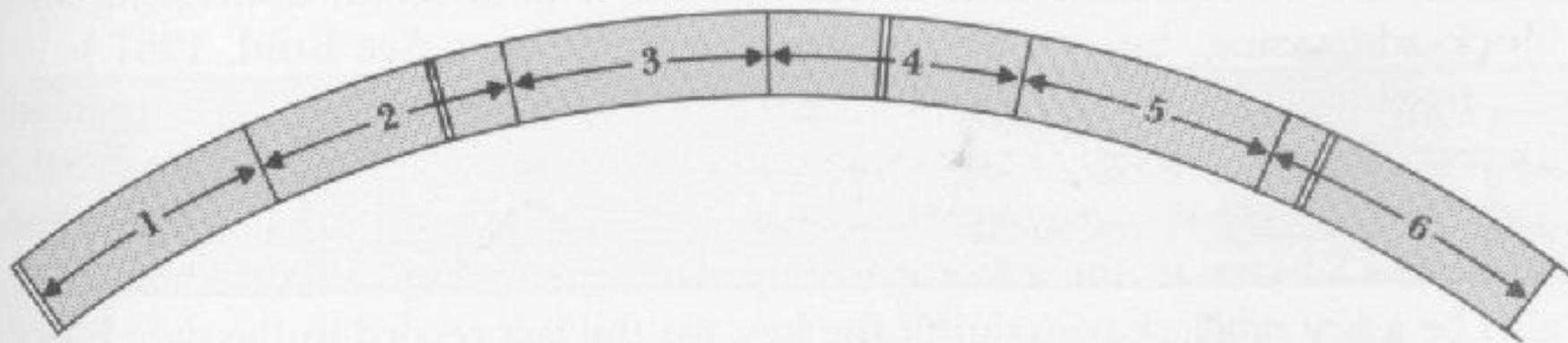
Fragmentação interna

- **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:
 - 1 registro por setor => fragmentação
 - Registros ocupando mais de 1 setor => acesso mais complexo

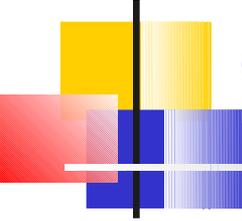
FIGURE 3.7 Alternate record organization within sectors (shaded areas represent data records, and unshaded areas represent unused space).



(a)

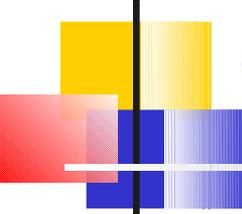


(b)



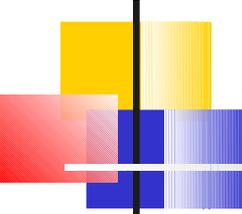
Sistema de Arquivos

- 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 9x e dispositivos de memória Flash) 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 SO organizam as trilhas em **blocos** de tamanho definido pelo usuário

Setores X Blocos

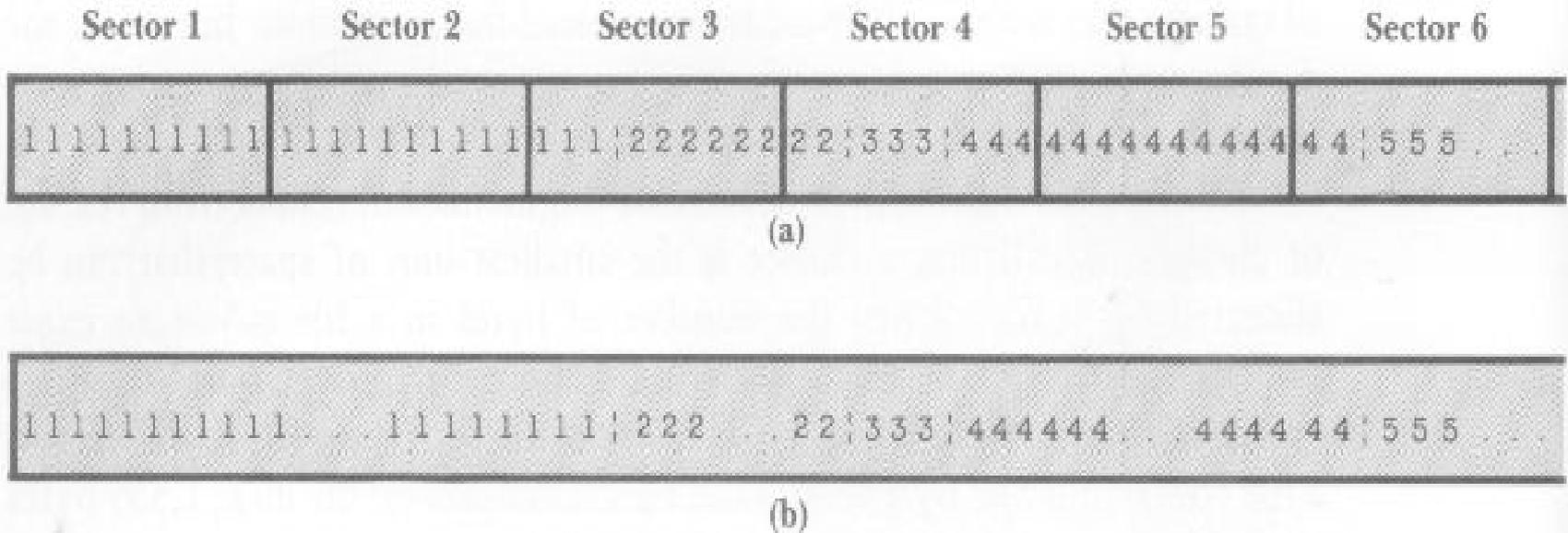
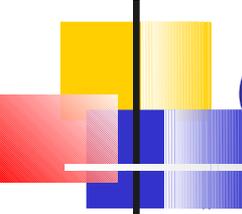
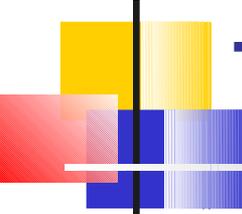


FIGURE 3.8 Sector organization versus block organization.



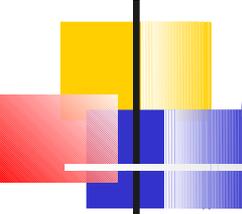
Overhead

- *Overhead* – espaço ocupado com informações para gerenciamento , introduzidas pelo processo de formatação do disco:
 - Endereços, identificadores, código de verificação de erro, marcadores de badblock.
 - não são dados “úteis”
- 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, 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!

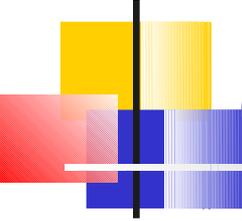


Outros sistemas de arquivos

- 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
 - Lida melhor com arquivos maiores, a depender do tamanho do volume

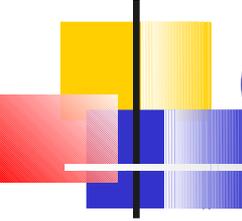
Tamanhos padrão de clusters para NTFS

Volume size	Windows NT 3.51	Windows NT 4.0	Windows 2000, Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008
7 MB–512 MB	512 bytes	4 KB	4 KB
512 MB–1 GB	1 KB	4 KB	4 KB
1 GB–2 GB	2 KB	4 KB	4 KB
2 GB–2 TB	4 KB	4 KB	4 KB
2 TB–16 TB	Not Supported*	Not Supported*	4 KB
16TB–32 TB	Not Supported*	Not Supported*	8 KB
32TB–64 TB	Not Supported*	Not Supported*	16 KB
64TB–128 TB	Not Supported*	Not Supported*	32 KB
128TB–256 TB	Not Supported*	Not Supported*	64 KB



Outros sistemas de arquivos

- ext3 e 4
 - *clusters* entre 1 e 8KB

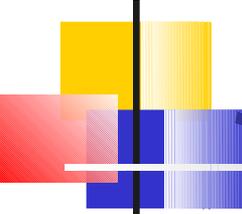


Custo de acesso a disco

- 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

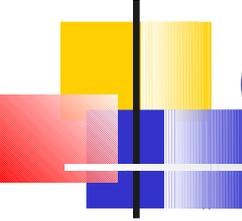
Tempo transferência =

$(n^{\circ} \text{ de bytes transferidos} / n^{\circ} \text{ de bytes por trilha}) * \text{tempo de rotação}$



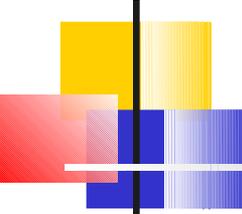
Seeking

- 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**



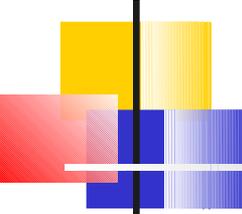
Observação

- 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 seqüencial)



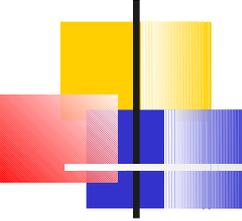
Exercício em duplas

- 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?



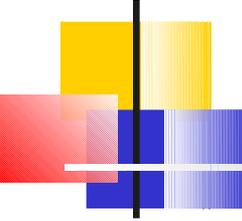
Exercício: resolução

- 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



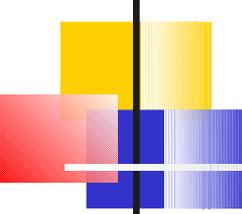
Para discussão

- Você está projetando seu próprio HD e decide armazenar os arquivos em espaços contínuos, ignorando limites de setores/clusters/extents/blocos/cilindros
 - Possíveis **vantagens**?
 - Possíveis **desvantagens**?



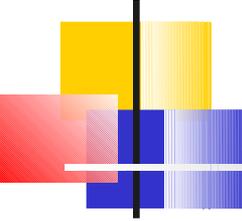
Discos

- Discos são *gargalos*
 - Discos são mais lentos que a CPU
 - Muitos processos são “disk-bound”, i.e., CPU tem que esperar pelos dados do disco



Técnicas para minimizar o problema

- **Multiprogramação:** CPU trabalha em outro processo enquanto aguarda o disco
- **RAID** (*Redundant Array of Inexpensive Disks*)
 - vs. SLED (*Single Large Expensive Disk*)
 - **Striping:** o arquivo é repartido entre vários *drives* (paralelismo), preferencialmente de forma transparente para o usuário/programa
 - Espelhamento: redundância de dados



Técnicas para minimizar o problema

- ***Disk cache***: blocos de memória RAM configurados para conter páginas de dados do disco
 - Ao ler dados de um arquivo, o cache é verificado primeiro; se a informação desejada não é encontrada, um acesso ao disco é realizado e o novo conteúdo é carregado no cache
- ***RAM Disk***: simula em memória o comportamento do disco mecânico
 - Carrega arquivos muito usados, dados descompactados, etc.

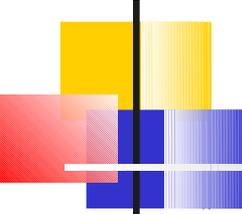
Fitas Magnéticas



- Introduzidas pela **IBM** na década de 50
 - Padronizou o tamanho do byte como 8 bits!
- Material plástico coberto por material magnetizável (óxido de ferro ou de cromo)

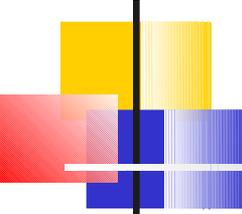
Fitas Magnéticas





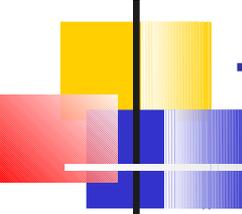
Fitas Magnéticas

- Leitor
 - Motor que rotaciona a fita
 - Cabeças de leitura que lêem a fita seqüencialmente
- Tecnologia similar a fitas cassetes
- Sofre mais **desgaste** que discos



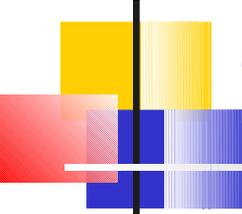
Fitas Magnéticas

- Fitas: permitem acesso seqüencial muito rápido, mas não permitem acesso direto/aleatório
- Compactas, resistentes, fáceis de transportar, mais baratas que disco
- Usadas como memória terciária (back-up, arquivo-morto) juntamente com os discos óticos



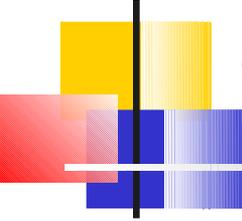
Organização dos dados na fita

- Posição de um registro é dada por um **deslocamento em bytes** (*offset*) relativo ao início do arquivo
- **Posição lógica** de um byte no arquivo corresponde diretamente à sua **posição física** relativa ao início do arquivo



Superfície da fita

- A superfície pode ser vista como um **conjunto de trilhas paralelas**, cada qual sendo uma seqüência de bits
- **9 trilhas paralelas** formam **1 frame**
 - Cada trilha tem 1 byte + paridade (em geral, paridade ímpar, i.e., o número de bits 1 é ímpar)

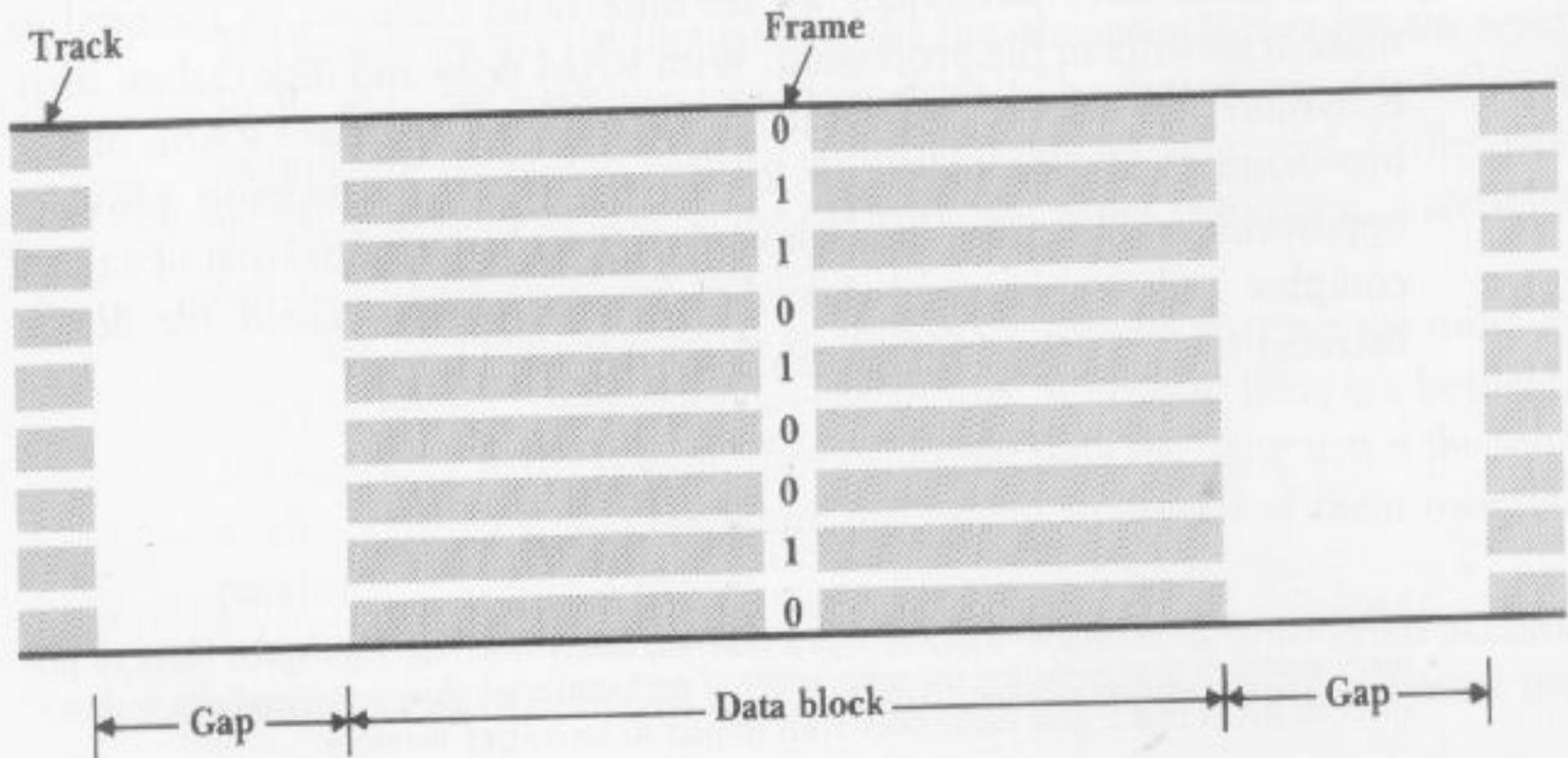


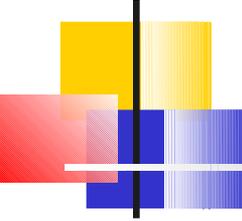
Superfície da fita

- Frames são agrupados em **blocos de dados** de tamanhos variados, os quais são separados por **intervalos (*interblock gaps*)** sem informações
- Intervalos são necessários para viabilizar parada/reinício

Superfície da fita

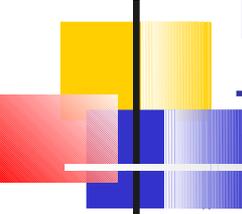
FIGURE 3.11 Nine-track tape.





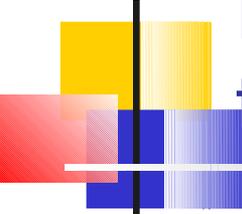
Medidas de comparação

- **Densidade:** bpi - *bytes per inch*
 - Ex: 6.250 bpi
- **Velocidade:** ips - *inches per second*
 - Ex: 200 ips
- **Tamanho do 'interblock gap':** *inches*
 - Ex: 0.3 *inches*
- 1 *inch* (polegada) ~ 2,5 cm



Exercício: estimativa do tamanho de fita necessário

- EX: armazenar em fita 1.000.000 de registros com 100 bytes cada. Suponha fita com 6.250 bpi, com intervalo entre blocos de 0.3 polegadas. Quanto de fita é necessário?
- **b** = comprimento físico do bloco de dados (pol.)
- **g** = comprimento físico do intervalo (pol.)
- **n** = número de blocos de dados
- **S** = comprimento de fita necessário (espaço físico) é dado por: $S = n * (b + g)$

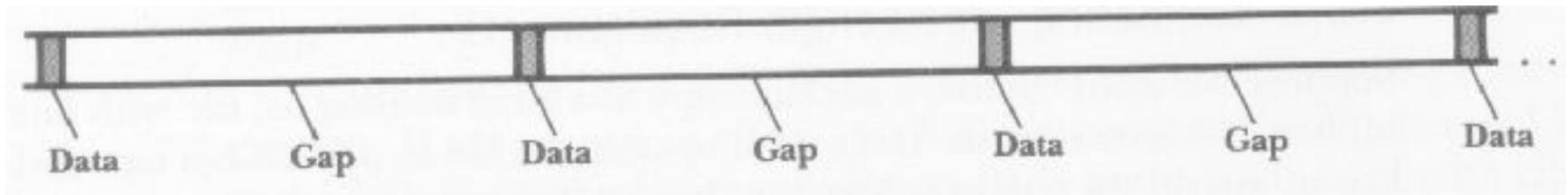


Resolução do exercício: estimativa do tamanho de fita necessário

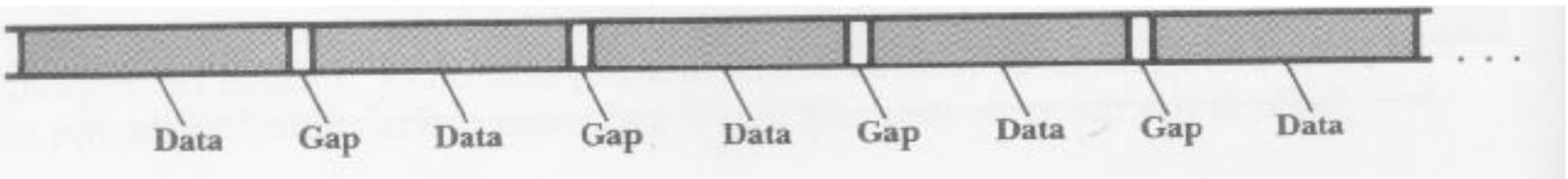
- Supondo **1 bloco = 1 registro**:
 $S = n \cdot (b + g)$
 $S = 1.000.000 \cdot (100/6.250 + 0.3)$
 $S = 316.000 \text{ pol} \sim 7.900 \text{ m}$
- Supondo **1 bloco = 50 registros**
 - $n = 1.000.000/50 = 20.000$ blocos
 - $b = 5000/6250 \sim 0.8 \text{ pol}$
 - $S = 20.000 \cdot (0.8 + 0.3) = 22.000 \text{ pol} \sim 492 \text{ m}$
- **Comprimentos típicos de fitas: 91 a 1.000 m**

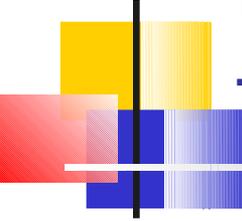
Estimativa do tamanho de fita necessário

- 1 registro por bloco



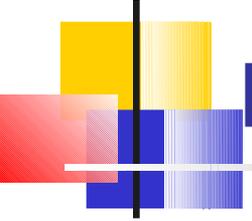
- 50 registros por bloco





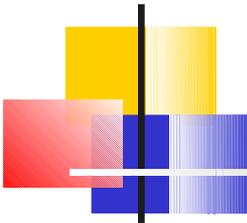
Estimativa de tempos de transmissão

- Taxa nominal de transmissão de dados = densidade (bpi) * velocidade (ips)
- Ex: Fita de 6.250 bpi e 200 ips
taxa transmissão = $6.250 * 200 = 1.250$ KB/s



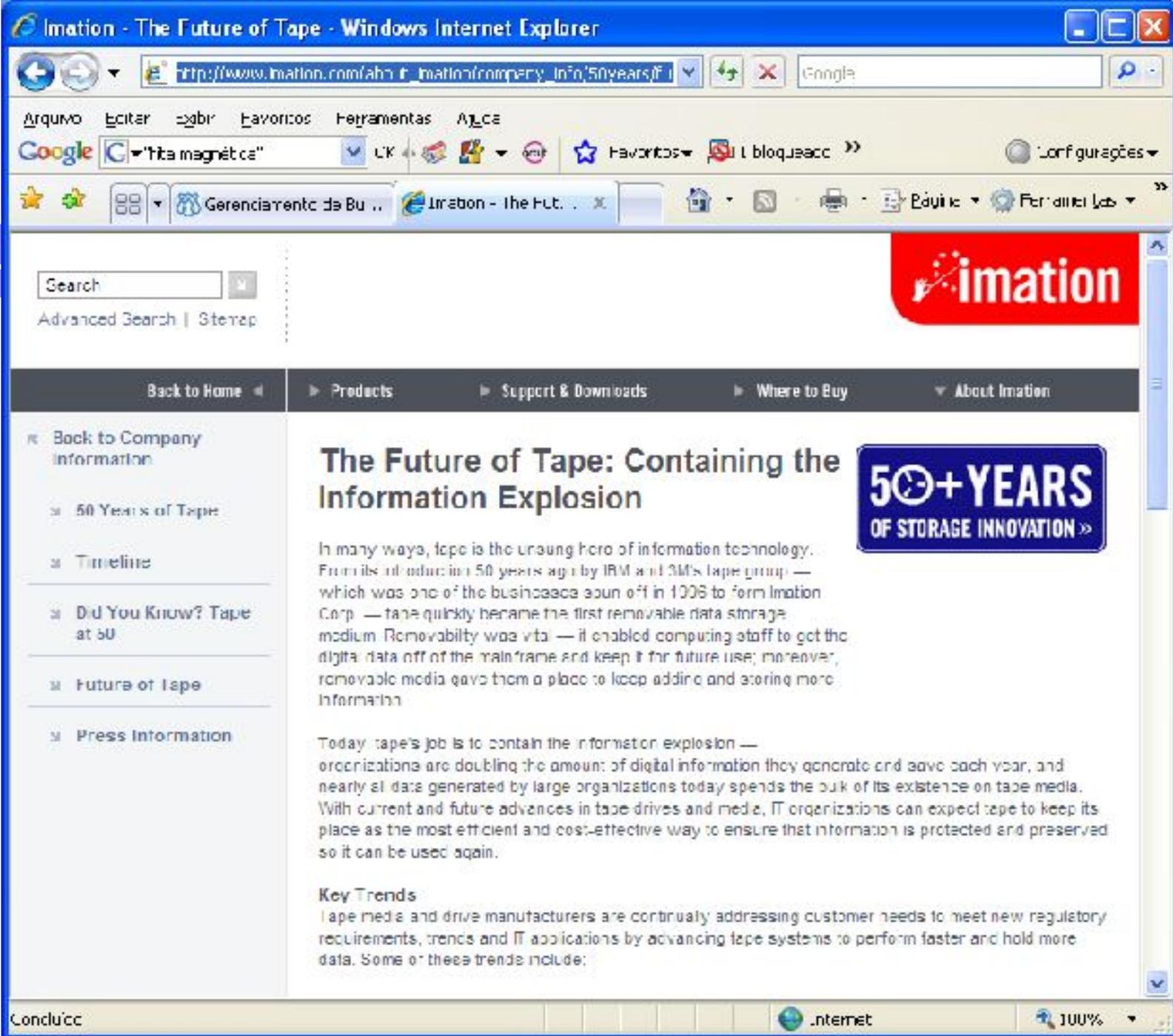
Quando usar fitas magnéticas

- Apropriadas para armazenamento seqüencial, quando não é necessário acesso direto/aleatório
- Quando não é necessária a atualização imediata (alterações periódicas são suficientes)
- Baixo custo e alta capacidade, adequada para armazenagem e transporte



Fitas magnéticas

- Tecnologia morta?



Search
Advanced Search | Stop



- Back to Company Information
- 50 Years of Tape
- Timeline
- Did You Know? Tape at 50
- Future of Tape
- Press Information

The Future of Tape: Containing the Information Explosion

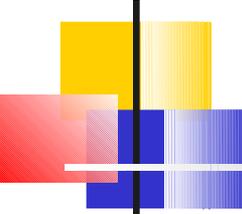


In many ways, tape is the unsung hero of information technology. From its introduction 50 years ago by IBM and 3M's tape group — which was one of the businesses spun off in 1998 to form Imaton Corp. — tape quickly became the first removable data storage medium. Removability was vital — it enabled computing staff to get the digital data off of the mainframe and keep it for future use; moreover, removable media gave them a place to keep adding and storing more information.

Today, tape's job is to contain the information explosion — organizations are doubling the amount of digital information they generate and save each year, and nearly all data generated by large organizations today spends the bulk of its existence on tape media. With current and future advances in tape drives and media, IT organizations can expect tape to keep its place as the most efficient and cost-effective way to ensure that information is protected and preserved so it can be used again.

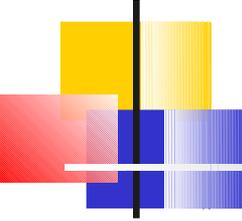
Key Trends

Tape media and drive manufacturers are continually addressing customer needs to meet new regulatory requirements, trends and IT applications by advancing tape systems to perform faster and hold more data. Some of these trends include:



Fitas magnéticas

- Imation (antiga 3M, parceira da IBM), 2004
 - *Recentemente, anunciamos um investimento de 49 milhões de dólares em uma moderníssima unidade de revestimento de fita, que manterá a Imation na linha de frente da tecnologia, além de desenvolvermos nossos cartuchos com capacidades de um terabyte e mais.*
 - *A fita tem sido -e ainda é- a maneira mais rentável para as empresas realizarem backups e recuperação de seus dados.*



Fitas magnéticas

- *Nova fita magnética é construída com nanotecnologia*
 - Redação do Site Inovação Tecnológica, 25/07/2005
 - *A empresa japonesa FujiFilm anunciou o lançamento do primeiro meio de armazenamento digital de dados baseado na nanotecnologia. Trata-se de uma fita magnética, voltada para backups em grandes centros de computação, que consegue armazenar até 300 GB de informações... a fita que agora foi lançada consegue garantir fidelidade dos dados gravados por 30 anos. A tecnologia Nanocubic consiste em uma camada ultra-fina de nanopartículas magnéticas, que são aplicadas por um processo que permite o controle preciso da espessura da camada de gravação de dados...*

Categorias

- Lista Completa
- Access Point
- Accessório
- Cabo
- Caixas de Som
- Calculadora
- Camera Digital
- Captura e Leitura
- Cartão de Memória
- Computador
- Cooler
- Disquete
- DVD Portátil
- DVD-RW
- Eletrônica
- Fax/Modem
- Filtro de Ar
- Firewall
- Fita Magnética
- Fita Magnética HP
- Fita Magnética Sony
- Fone de Ouvido
- Fonte
- Gabinete
- Imã
- GPS
- Hard Disk
- Hard Disk HP

Fita Magnética

Destaque em fita magnetica



Fita Magnética Sony SDX5-400C AIT-5 246 Metros 400GB-1040GB
24x R\$ 28,07

ou desce-lo no boleto à vista por R\$ 101,40

MARCA Selecione

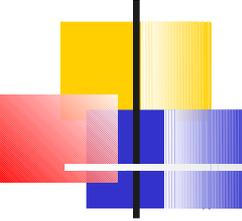
ORDENAR POR PREÇOS

Fita Magnética HP DAT 72 115R-DW026A 24-72GB 115R 2.0 8MB Int.	Fita Magnética HP DAT 72 115R-DW027A 24-72GB 115R 2.0 8MB Ext.	Fita Magnética HP DAT 160 SAS-Q1588A 80-160GB SAS 16MB Ext.	Fita Magnética HP DAT 160 115R-Q1581A 80-160GB 115R 16MB Ext.
R\$ 2.151,03 No boleto por R\$ 2.015,23	R\$ 2.323,69 No boleto por R\$ 2.174,90	R\$ 2.822,56 No boleto por R\$ 2.641,92	R\$ 2.776,99 No boleto por R\$ 2.599,26
+ Modelos de Fita Magnética	+ Modelos de Fita Magnética	+ Modelos de Fita Magnética	+ Modelos de Fita Magnética
Fita Magnética HP Ultrium 1760 F8070SK 1.6TB-5.760GB 110-4 SAS 3GB sec SAS 128MB	Fita Magnética HP Ultrium 448-170835A 200-400GB 3GB sec SAS 64MB Int.	Fita Magnética HP Ultrium 970-F8847A 400-800GB 3GB sec SAS 64MB Int.	Fita Magnética HP Ultrium 1760-F8070A 1.6TB-5.760GB 3GB sec SAS 128MB Int.
R\$ 8.985,85 No boleto por R\$ 8.036,35	R\$ 5.684,51 No boleto por R\$ 5.320,70	R\$ 7.246,77 No boleto por R\$ 6.702,90	R\$ 10.555,12 No boleto por R\$ 9.879,50
+ Modelos de Fita Magnética	+ Modelos de Fita Magnética	+ Modelos de Fita Magnética	+ Modelos de Fita Magnética



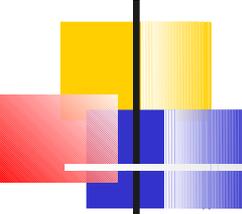
Produtos sob Encomenda
 Nós procuramos para você.





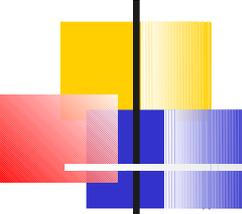
Memória Flash

- Um tipo de EEPROM (Electrically-Erasable Programmable Read-Only Memory)
 - um chip de armazenamento não-volátil
 - pode ser apagado eletronicamente e ser reprogramado
- Inventada em 1980 na Toshiba
- Se popularizou 20 anos depois com preços mais acessíveis



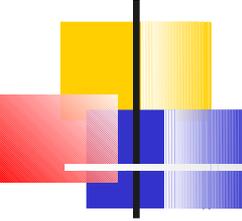
Memória Flash

- Tempo de acesso: 0,8 milissegundos
 - (HD ~10 ms - RAM ~0,0007 ms)
- Boa resistência:
 - choque, pressão, temperatura e umidade



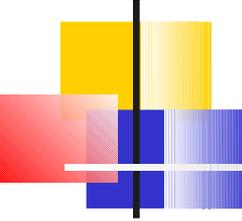
Memória Flash

- Flash NOR / NAND
 - NOR permitia acessar dados de maneira aleatória, mas com baixa velocidade
 - NAND faz acesso sequencial e acessa blocos de células (páginas), é o tipo mais usado atualmente pelo custo mais baixo



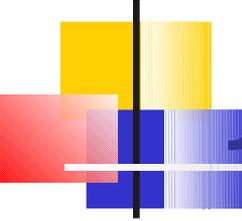
Memória Flash

- O processo de apagar um bloco modifica todos os bits para 1
- Qualquer local do bloco novo pode ser programada (modificada para 0)
- No entanto, quando um bit é modificado para 0, apenas apagando o bloco inteiro é possível modificá-lo para 1 novamente



Memória Flash

- Há um número **finito** de ciclos programar-apagar
 - A maioria dos produtos garantem aproximadamente 100 mil ciclos
 - Após isso o dispositivo pode começar a deteriorar

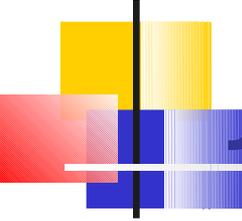


Jornada de um byte

- O que acontece quando 1 programa escreve um byte p/ um arquivo em disco?

?

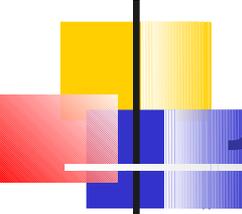
- Programa → disco



Jornada de um byte

```
char ch = 'P';  
fwrite(&ch, 1, 1, fp);
```

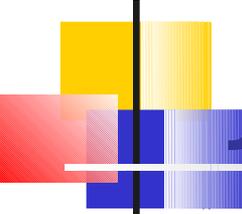
Todo o controle é passado ao sistema operacional que possui uma série de programas, compostos de camadas a cada nível com tratamento mais físico.



Jornada de um byte

- **Operações na memória**

- O comando ativa o SO (parte denominada *file manager*), que supervisiona a operação
 - Verifica se o arquivo existe, se tem permissão de escrita, etc.
 - Obtém a localização do arquivo físico (drive, cilindro, cluster ou extent) correspondente ao arquivo lógico
 - Determina em que setor escrever o byte
 - Verifica se esse setor já está no buffer de E/S; se não estiver, carrega-o, e escreve o byte no buffer
 - Antes de escrever no disco, aguarda possíveis bytes para o mesmo setor



Jornada de um byte

- Operações fora da memória
 - Processador de E/S
 - Aguarda a disponibilidade do recurso para poder efetivamente disparar a escrita no disco
 - Controlador de disco
 - Verifica se *drive de disco* está disponível para escrita
 - Instrui *drive* para mover cabeça de L/E para trilha/setor corretos
 - Disco rotaciona, o setor (e o novo byte) é escrito

Jornada de um byte

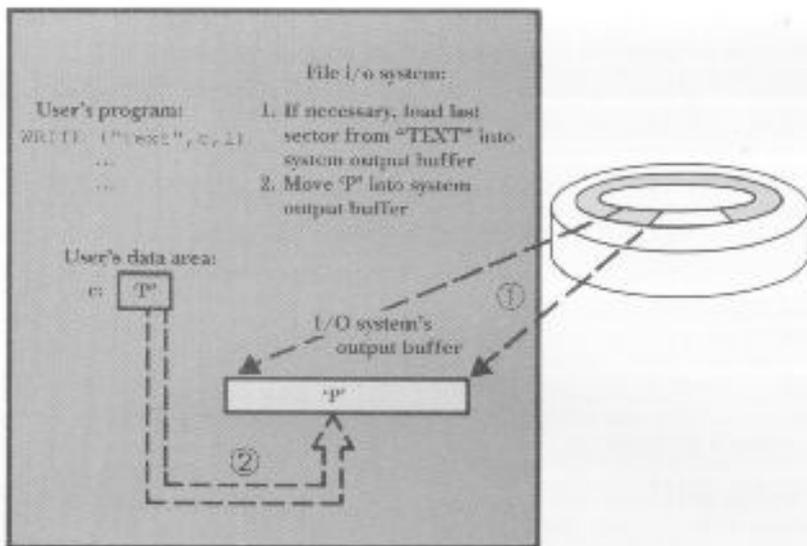


FIGURE 3.15 The file manager moves *P* from the program's data area to a system output buffer, where it may join other bytes headed for the same place on the disk. If necessary, the file manager may have to load the corresponding sector from the disk into the system output buffer.

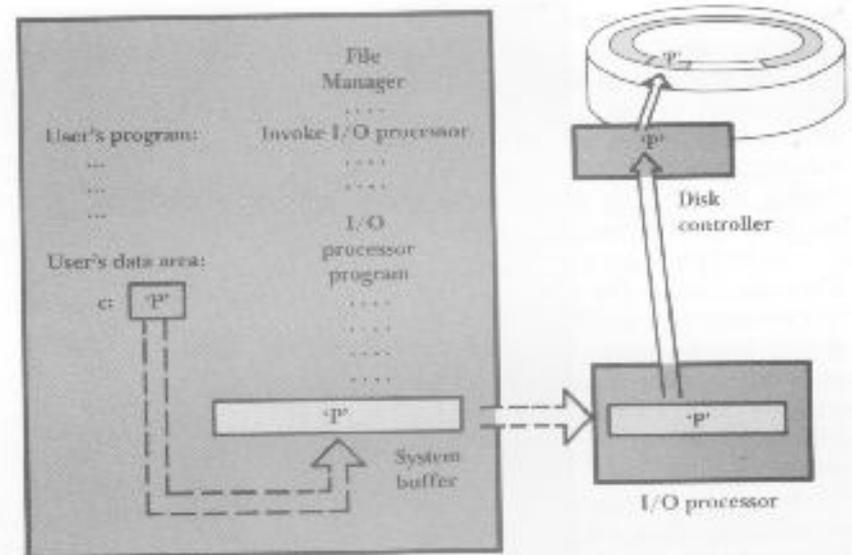
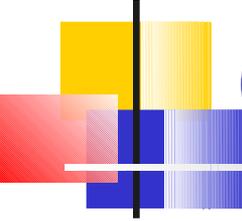
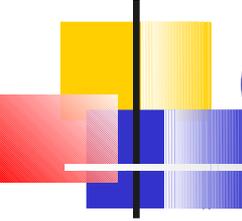


FIGURE 3.16 The file manager sends the I/O processor instructions in the form of an I/O processor program. The I/O processor gets the data from the system buffer, prepares it for storing on the disk, and then sends it to the disk controller, which deposits it on the surface of the disk.



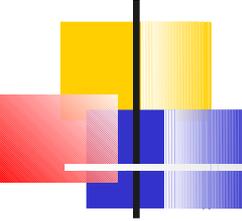
Gerenciamento de buffer

- *Buffering*
 - Permite trabalhar com grandes quantidades de RAM para armazenar informação sendo transferida, de modo a reduzir o número de acessos ao dispositivo de memória secundária



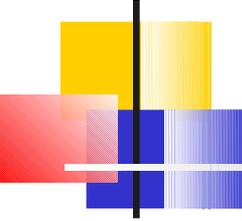
Gerenciamento de buffer

- *Buffering*
 - Um gerenciador de arquivos aloca *buffers* de E/S grandes o suficiente para conter dados, mas quantos são necessários?



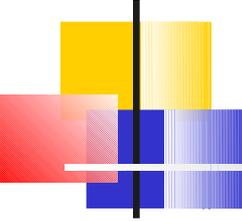
Buffer como gargalo

- Suponha um sistema que utilize um *único buffer*
 - Em um programa que realiza intercaladamente operações de leitura/escrita, o desempenho seria muito ruim
 - Por quê?
- Os sistemas precisam de, no mínimo, 2 *buffers*: **1 para entrada, 1 para saída**
 - Por exemplo, enquanto um buffer é transmitido para o disco, a CPU carrega dados em outro(s)



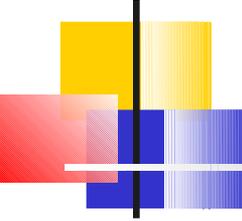
Buffer como gargalo

- Mesmo com 2 *buffers*, mover dados de e para o disco é muito lento, e os programas podem ficar “I/O bound”
- Para reduzir o problema
 - Algumas estratégias, que nem sempre são conhecidas pelo programador



Buffer como gargalo

- *Multiple buffering*
 - Double buffering é o mais comum
 - Vários buffers em um pool: escolhe-se um para usar (de preferência, o usado menos recentemente)
- *Compartilhamento de espaço entre programa e SO*
 - Área de dados do programa usada como buffer, ou buffer usado para área de dados



Buffer como gargalo

- *Divisão e junção de informações*
 - Vários buffers que guardam partes específicas das informações dos blocos (por exemplo, cabeçalho e dados), com escrita e leitura em paralelo