



Armazenamento Secundário

Estruturas de Dados II

Slides cedidos pelo Prof. Gustavo Batista (ICMC-USP)
[<http://www.icmc.usp.br/~gbatista/>]



Arquivos

- Mecanismo de organização da informação mantida em memória secundária:
 - HD;
 - Fitas;
 - CD;
 - DVD.
- Da mesma forma que variáveis e alocação dinâmica organizam o acesso do programa à memória principal.



Arquivos

- Existem dois principais motivos para a utilização de arquivos:
 - Acesso a memória não volátil;
 - Acesso a memória em grande quantidade e baixo custo.
- Entretanto, existe um grande custo associado:
 - Grande tempo de acesso.



Discos X Memória Principal

- Tempo de acesso:
 - HD: alguns milissegundos $\sim 10\text{ms}$;
 - RAM: alguns nanossegundos $\sim 10\text{ns} \dots 40\text{ns}$;
 - Ordem de grandeza da diferença entre os tempos de acesso é aproximadamente 250.000 (1/4 mais lento);
 - Supondo que o acesso à RAM equivale a buscar uma informação no índice de um livro que está em suas mãos, e que essa operação consome 20 segundos, o acesso a disco seria equivalente a mandar buscar a mesma informação em uma biblioteca que levaria quase 58 dias ($\sim 5.000.000$ segundos) para retornar a informação!!!
 - Porque? Basicamente porque discos são mecânicos e envolvem partes móveis.

* $1\text{ms} = 1.000.000\text{ns} \rightarrow 1.000.000/4$ (1/4) mais lento = 250.000



Discos X Memória Principal

	RAM	Discos (HD)
Custo	Alto	Baixo
Tempo de Acesso	Baixo	Alto
Capacidade	Baixa	Alto
Princípio	Elétrico	Magnético
Persistência	Volátil	Não-volátil
Acesso	Aleatório	Aleatório
Organização	Células	Trilhas/Setores



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

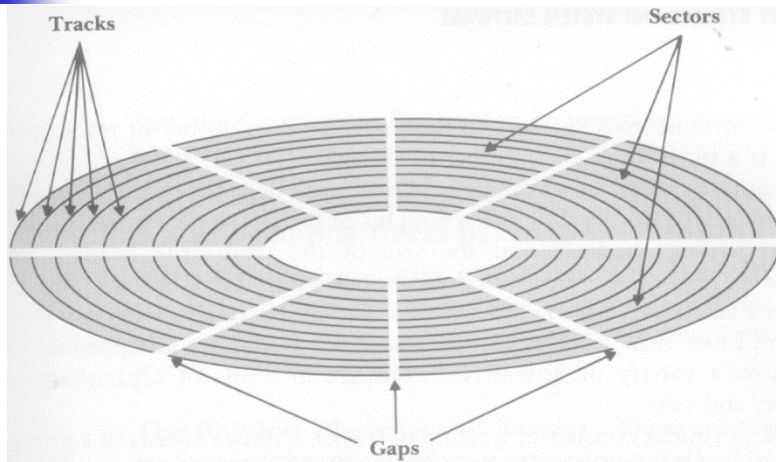


FIGURE 3.2 Surface of disk showing tracks and sectors.

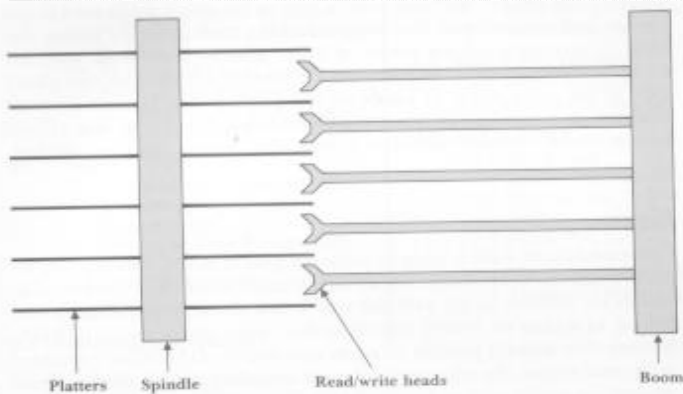
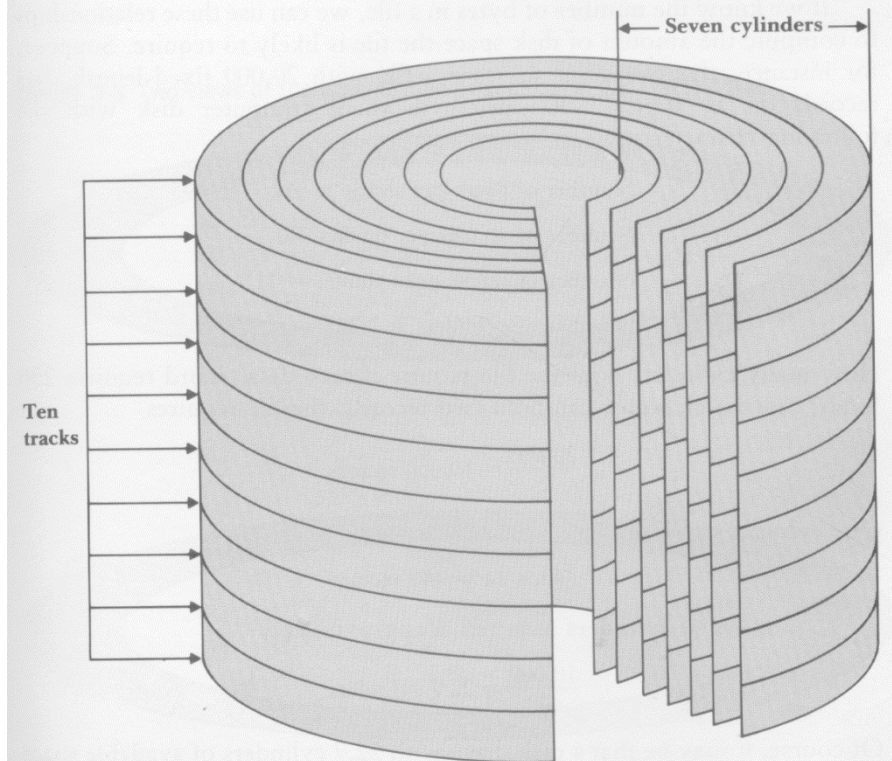


FIGURE 3.1 Schematic illustration of disk drive.

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:
 - `fread(&c,1,1,fd)`: 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.



Cluster

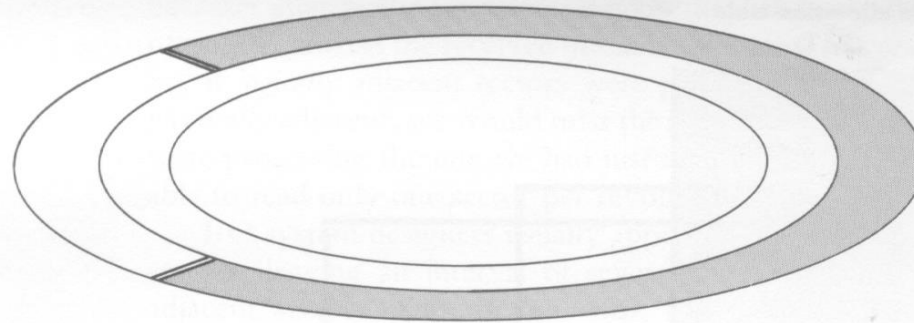
- Conjunto de setores logicamente contíguos no disco.
- Um arquivo é visto pelo S.O. como um grupo de clusters distribuídos no disco:
 - Arquivos são alocados em um ou mais clusters.



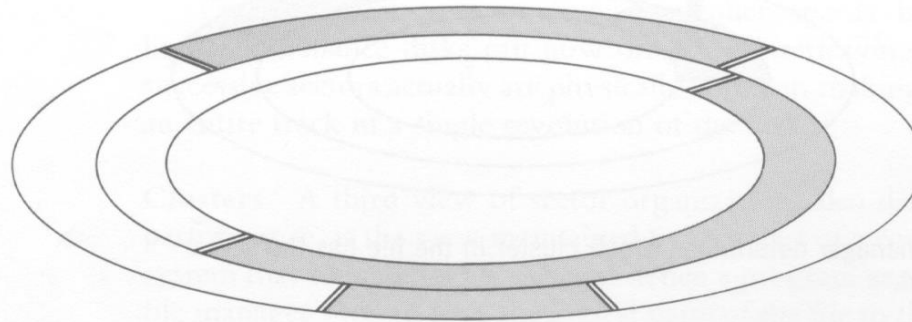
Extent

- Sequência de clusters consecutivos no disco, alocados para o mesmo arquivo.
- 1 seek 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



(a)



(b)

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

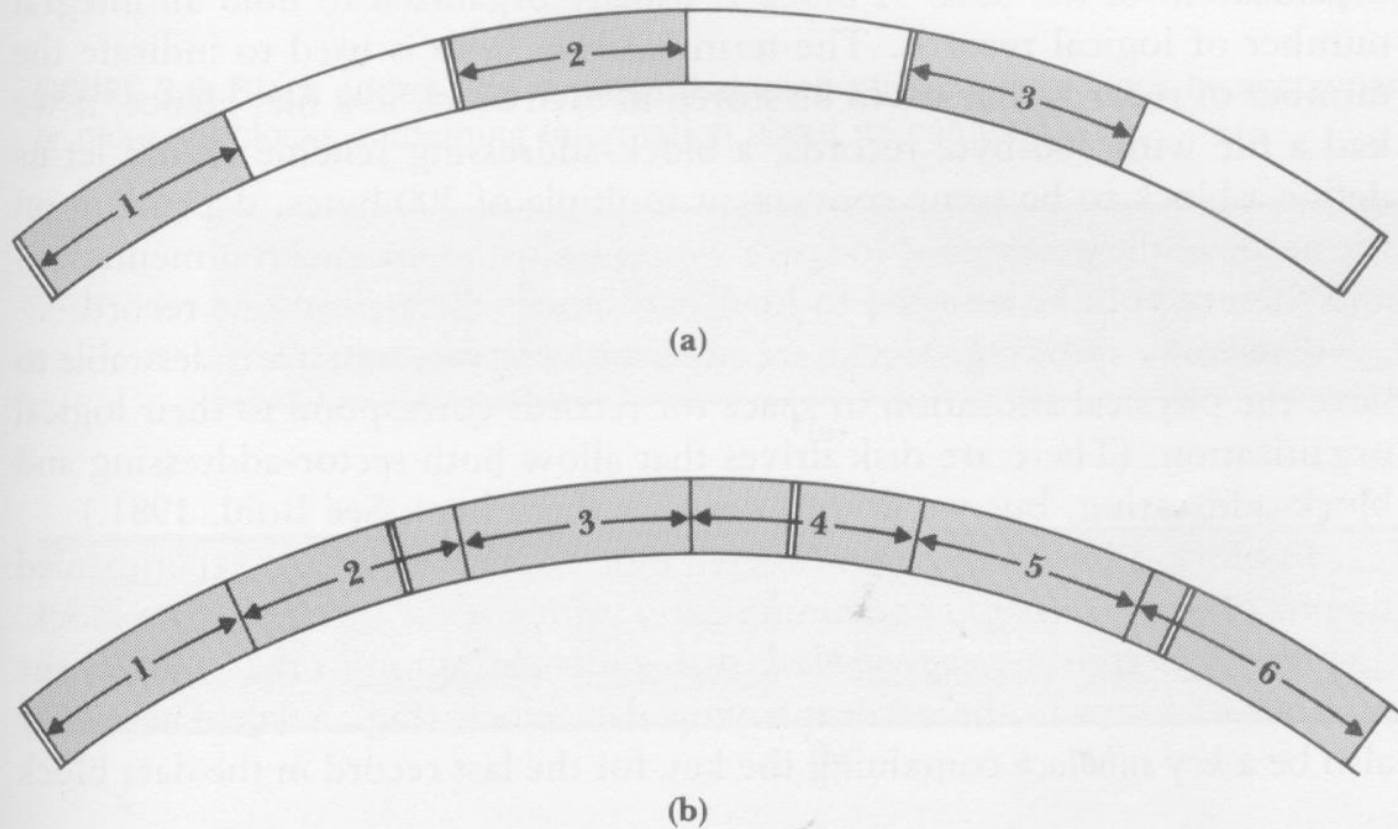


Fragmentação

- Perda de espaço útil decorrente da organização em setores de tamanho fixo.
 - Temos duas alternativas:
 - 1 registro por setor => fragmentação
 - Registros ocupando mais de 1 setor => acesso mais complexo
- Fragmentação também ocorre organizando os arquivos em clusters:
 - Ex: 1 cluster = 3 setores de 512 bytes, arquivo com 1 byte (quanto espaço se perdeu?)

Fragmentação

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





Capacidade do disco**

- 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.
- **Capacidade Nominal



Capacidade do disco**

- **Exemplo:**

- Queremos armazenar 20.000 registros de tamanho fixo num disco de 300 Mbytes que contém:
 - 512 bytes/setor
 - 40 setores/trilha
 - 11 trilhas/cilindro
 - 1331 cilindros
- **Quantos cilindros são necessários, se cada registro tem 256 bytes ?**
 - $2 \text{ registros/setor} = 10.000 \text{ setores}$. Se em cada cilindro há $40 \times 11 = 440$ setores, então o número de cilindros é aproximadamente $10.000/440 = 22.7$ cilindros.
 - É provável que um disco tenha essa quantidade de cilindros disponível, mas não de forma contígua, e o arquivo pode ser espalhando em dezenas, ou centenas, de cilindros.



Custo de acesso a disco

- É uma combinação de 3 fatores:
 - Tempo de busca (seek): tempo p/ posicionar o braço de acesso na posição (trilha/setor) desejada;
 - Delay de rotação: tempo p/ o disco rodar de forma que a cabeça de L/E esteja posicionada sobre a posição desejada;
 - Tempo de transferência: tempo p/ transferir os bytes.
- 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)



Tempo de Busca (Seek)

- Movimento de posicionar a cabeça de L/E sobre a trilha/setor desejado.
- É a parte mais expressiva do tempo de acesso.
 - Depende de quanto o braço precisa se movimentar.
 - Deve ser reduzido ao mínimo.
 - O conteúdo de todo um cilindro pode ser lido com 1 único seek.
- Para cálculos, se trabalha com o tempo de busca médio (tempo de busca para $1/3$ do número de cilindros).



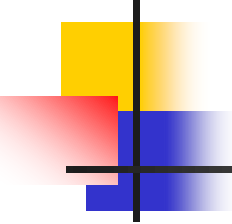
Delay de Rotação

- Por exemplo, para um HD de 5000 rpm, o delay de rotação é de 12ms.
- Na média, considera-se o delay de rotação de meia-volta.
- Na prática, esse delay é reduzido quando é possível ler/gravar o arquivo em setores da mesma trilha e trilhas do mesmo cilindro.



Tempo de Transferência

- Tempo transferência = $(n^{\circ} \text{ de bytes transferidos} / n^{\circ} \text{ de bytes por trilha}) * \text{tempo de rotação}.$
- Exemplo: disco de 10.000 rpm com 170 setores por trilha:
 - Para ler 1 setor: $1/170$ de rotação;
 - 10.000 rpm = 6ms por rotação;
 - Tempo de transf. para 1 setor = 0,035ms.



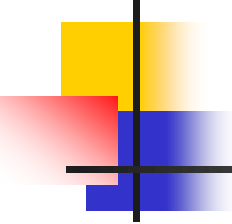
Exemplo de Cálculo de Tempo de Acesso

- Supor duas situações:
 - o arquivo está gravado sequencialmente em um disco;
 - o arquivo está espalhado pelo disco, cada cluster em uma trilha diferente.
- Objetivo:
 - Comparar os tempos de acesso e a influência do tempo de busca (seek).



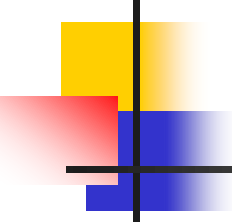
Exemplo de Cálculo de Tempo de Acesso

- Arquivo:
 - 8.704 kbytes = 8.704.000 bytes;
 - 34.000 registros de 256 bytes.
- HD/S.O.:
 - 1 setor = 512 bytes;
 - 1 cluster = 8 setores = 4096 bytes;
 - 1 trilha = 170 setores = 87.040 bytes (= 21 clusters).
- São necessários:
 - 2125 clusters ou
 - 100 trilhas.



Exemplo de Cálculo de Tempo de Acesso

- Primeira situação:
 - O arquivo está disposto em 100 trilhas.
- Supor os seguintes tempos (Seagate Cheetah):
 - Tempo de busca médio: 8 ms;
 - Delay de rotação: 3 ms;
 - Tempo de leitura de 1 trilha: 6 ms;
 - Total: 17 ms.
- Tempo total = $100 \times 17 \text{ ms} = 1.7 \text{ s}$.



Exemplo de Cálculo de Tempo de Acesso

- Segunda situação:
 - 2125 clusters alocados dispersos pelo disco.
- Supor os seguintes tempos (Seagate Cheetah):
 - Tempo de busca médio: 8 ms;
 - Delay de rotação: 3 ms;
 - Tempo de leitura de 1 cluster: 0.28 ms;
 - Total: 11.28 ms.
- Tempo total = $2125 \times 11.28 \text{ ms} = 23.9 \text{ s}$.



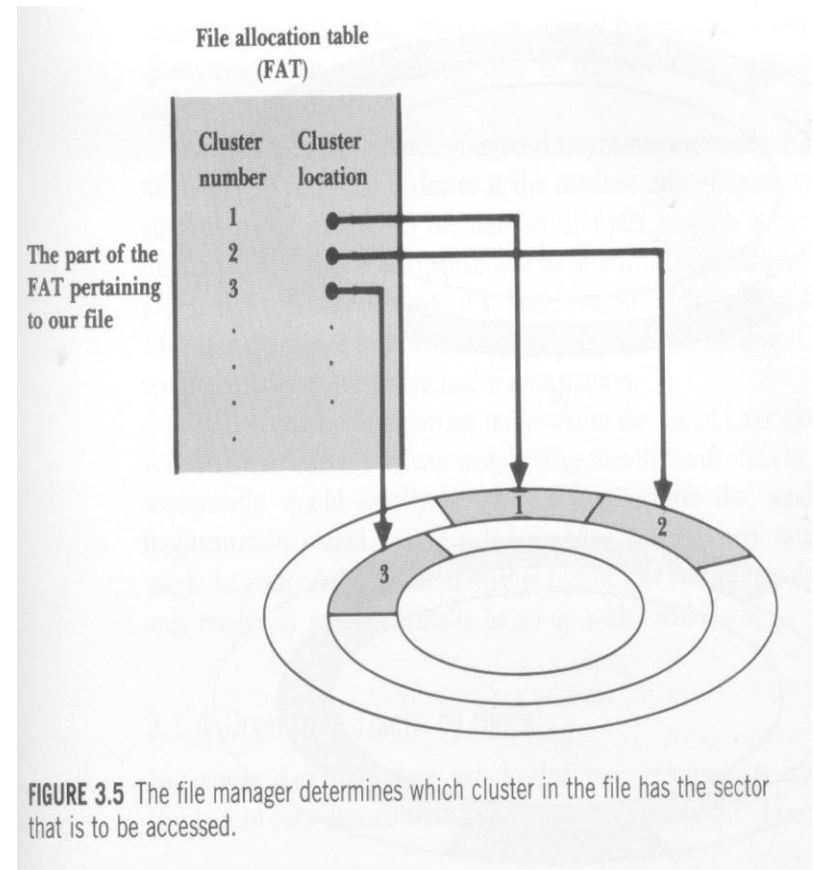
Sistema de Arquivos

- A organização do disco em setores/trilhas/cilindros é uma formatação física (já vem da fábrica).
- É necessária uma formatação lógica, que “instala” o sistema de arquivos no disco:
 - Subdivide o disco em regiões endereçáveis

Sistema de Arquivos

FAT – File Allocation Table

- Cada entrada na tabela dá a localização física do cluster associado a um certo arquivo lógico.
- 1 seek para localizar 1 cluster:
 - Todos os setores do cluster são lidos sem necessidade de seeks adicionais.



$$2^{16} = 65.536 \text{ posições} \times 512 \text{ bytes} = 33.554.432 \text{ bytes} = 32\text{MB}$$



Sistema de Arquivos

Tamanho do Cluster

- Definido pelo S.O. quando o disco é formatado.
- FAT (DOS/Windows): 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!



Sistema de Arquivos

- O sistema de arquivos FAT (DOS/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 S.O. determinar em qual cluster ele está (FAT).



Outros Sistemas de Arquivos

- FAT32 (Windows 95 e posteriores):
 - clusters de tamanho menor, endereça mais clusters, menos fragmentação.
- NTFS (New Technology File System):
 - Windows NT;
 - Mais eficiente: a menor unidade de alocação é o próprio setor de 512 bytes.
- ReFS (Resilient File System)
 - Construído sobre o NTFS



Materiais Visuais e de Apoio

- Funcionamento do Disco

- <https://www.youtube.com/watch?v=XgrT22tyYXA>
- <http://www.infowester.com/hd.php>

- SSD

- <http://www.infowester.com/ssd.php>



https://pt.wikipedia.org/wiki/Fita_magn%C3%A9tica

Fitas Magnéticas



Notícias

- <http://olhardigital.uol.com.br/noticia/ibm-e-fujifilm-batem-recorde-e-armazenam-220-tb-em-fita-cassete/48199>
- <https://www.dell.com/pt-br/work/shop/solu%C3%A7%C3%B5es-para-armazenamento-de-dados/sc/storage-products/data-protection>



Fitas Magnéticas

- Fitas: permitem acesso sequencial muito rápido, mas não permitem acesso direto
- Compactas, resistentes, fáceis de transportar, mais baratas que discos
- Usadas como memória terciária (back-up, arquivo morto, ...)



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

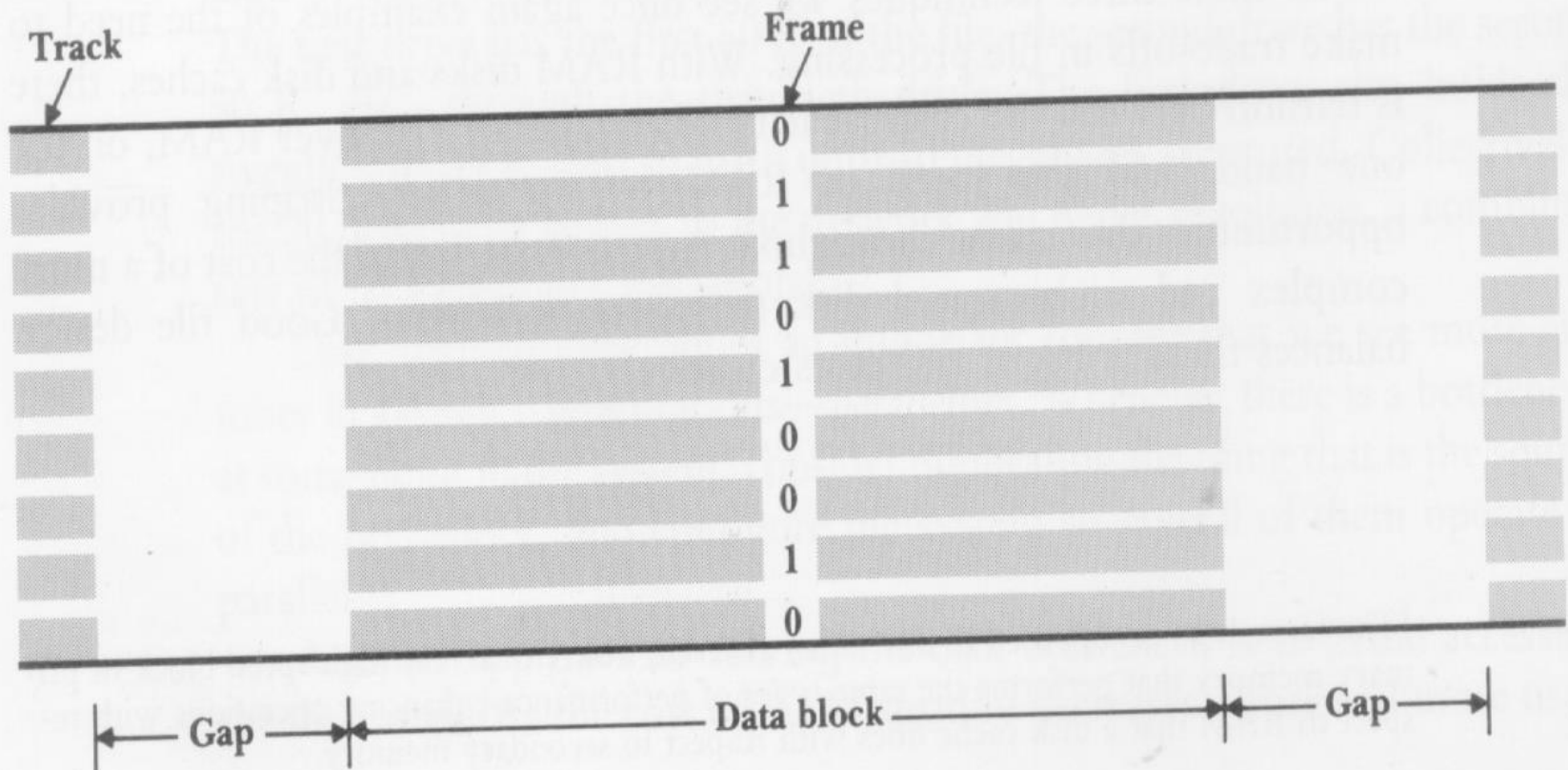


Superfície da fita

- A superfície pode ser “vista” como um conjunto de trilhas paralelas, cada qual sendo uma sequência de bits.
- 9 trilhas paralelas (1 frame): 1 byte + paridade (em geral, paridade ímpar, i.e., o número de bits = 1 é ímpar)
- 1 frame = 1 byte (8 bits em 8 trilhas) + paridade

Superfície da fita

FIGURE 3.11 Nine-track tape.





Superfície da fita

- Frames são agrupados em blocos de dados de tamanhos variados, os quais são separados por intervalos (interblock gaps) sem informações
 - Um grande número de frames com 0 é usado para preencher os intervalos
- Intervalos são necessários para viabilizar parada/reinício



Medidas de comparação para drivers de fita

- 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



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? Sejam:
- 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: $S=n*(b+g)$

$$S=n*(b+g)$$

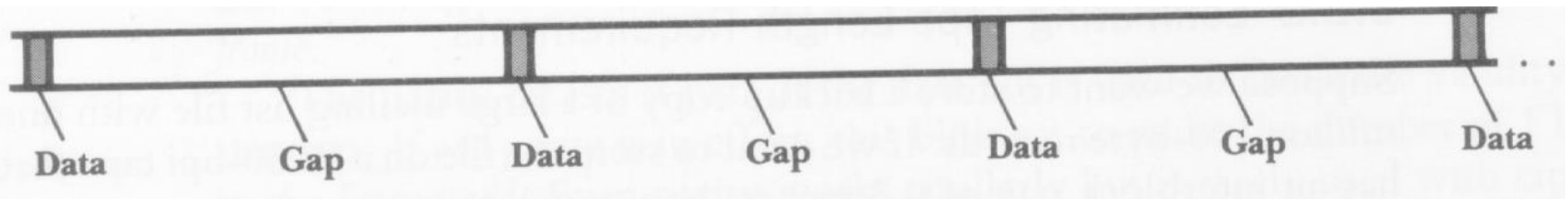


Estimativa do tamanho de fita necessário

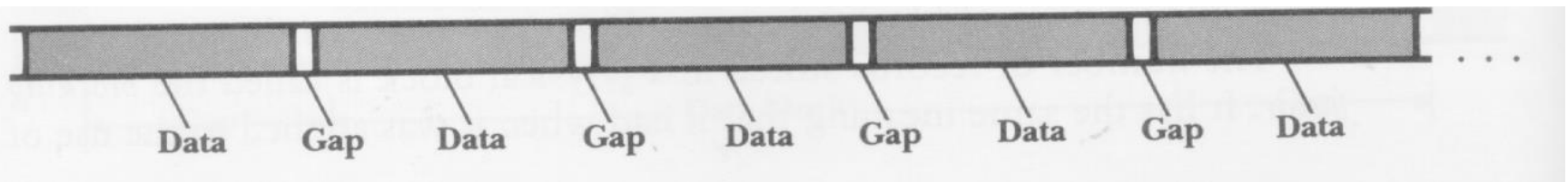
- Supondo 1 bloco = 1 registro:
 $S=1.000.000*(100/6.250+0.3)$
 $S=1.000.000*(0.016+0.3)$
 $S=316.000 \text{ pol} \sim 7.900 \text{ m}$
- Supondo 1 bloco=50 registros
 - $n=1.000.000/50=20.000$ blocos
 - $b=5000/6250 \sim 0.8 \text{ pol}$
 - $S=20.000*(0.8+0.3)=22.000 \text{ pol} \sim 550 \text{ m}$

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 = $6250 * 200 = 1.250 \text{ KB/s}$



Jornada de um Byte



Jornada de um byte

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

`write(arq,&c,1)`



Jornada de um byte

- **Operações na memória**
 - O comando ativa o S.O (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)
 - Instrui o processador de E/S (que libera a CPU de cuidar do processo de transferência) sobre a posição do byte na RAM, e onde ele deve ser colocado no disco



Jornada de um byte

- **Operações fora da memória**
 - **Processador de E/S:**
 - Formata o dado apropriadamente e decide o melhor momento de escrevê-lo no disco
 - Aguarda a disponibilidade do recurso p/ poder efetivamente disparar a escrita no disco
 - Envia o dado para o controlador de disco
 - **Controlador de disco:**
 - Instrui o drive p/ mover cabeça de L/E para trilha/setor corretos
 - Disco rotaciona até o setor e o novo byte é escrito



Gerenciamento de Buffer



Gerenciamento de buffer

- A transferência de dados entre a área de dados do programa e a memória secundária requer o uso de *buffers** de memória
- **Buffering**: permite trabalhar com grandes quantidades de dados em RAM, de modo a reduzir o nº de acessos que precisam ser feitos ao dispositivo de armazenamento secundário (buffers de E/S (buffers do sistema, não dos programas))

*memória temporária usada para armazenar e ler dados



Buffer como gargalo

- Suponha que um sistema utilize um único buffer. Em um programa que realiza, intercaladamente operações de leitura/escrita o desempenho seria muito ruim (Porque?).
- Os sistemas precisam de, no mínimo, 2 buffers: 1 p/ leitura, 1 p/ escrita
 - Mesmo no caso de transferência em uma única direção a existência de um único buffer retarda consideravelmente as operações, porque é demorado preencher um buffer com o conteúdo de todo um setor.

Buffer como gargalo

- Mesmo com 2 buffers, mover dados de e para o disco é muito lento
- Para reduzir o problema:
 - Multiple buffering
 - Double buffering

FIGURE 3.17 Double buffering: (a) The contents of system I/O buffer 1 are sent to disk while I/O buffer 2 is being filled; and (b) the contents of buffer 2 are sent to disk while I/O buffer 1 is being filled.

