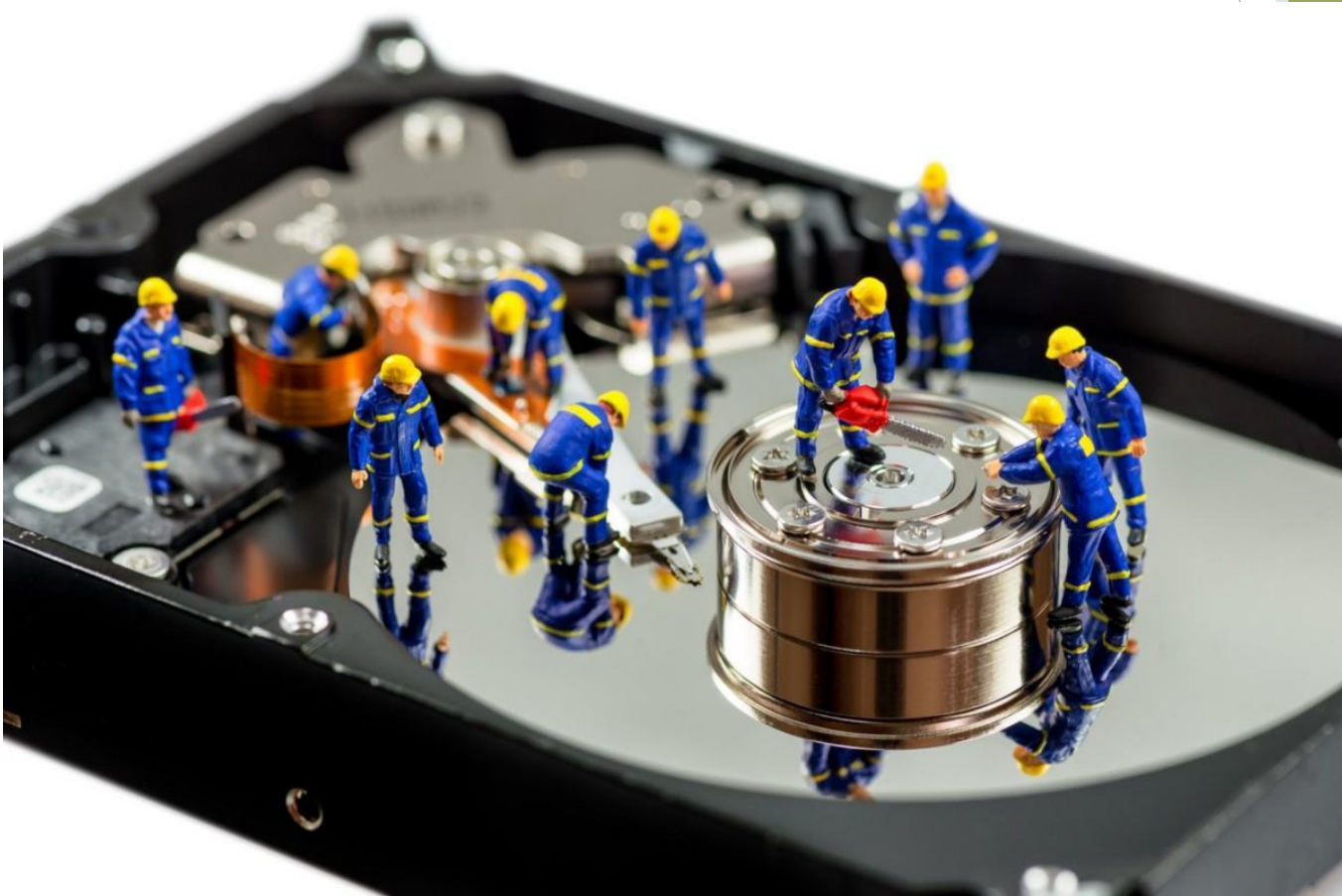


116327 -Organização de Arquivos



Organização de Arquivos

Disciplina: 116327

Prof. Oscar Fernando Gaidos Rosero

Universidade de Brasília - UnB
Instituto de Ciências Exatas - IE
Departamento de Ciência da Computação - CIC

Organização de Arquivos

Aula 4

M.Sc. Oscar Gaidos

Universidade de Brasília (UnB)

Sumário

Armazenagem Secundária

1. **Organização dos discos magnéticos**
2. Tempo de acesso ao disco
3. Fitas magnéticas (aplicações, tamanho do bloco, velocidade de transmissão)
4. Outros dispositivos
5. Transferência dos dados entre o programa e o dispositivo de armazenagem secundária
6. Conceito e técnicas de gerenciamento de buffers

Organização de dados

Existem duas formas básicas de organizar dados em um disco:

- ▶ Por setores
- ▶ Por blocos

Organização das trilhas em setores

- ▶ Como o disco vai diminuindo seu espaço na medida em que se aproxima do centro, logicamente os setores próximos ao centro são menores que os setores encontrados na parte externa do disco.
- ▶ Porém todos os setores tem um volume fixo de dados (512 bytes).

Organização em Setores

Cada setor arquiva 512 bytes de dados, mas o computador sabe onde estão guardados esses dados? E como ele consegue identificar esses endereços?

Organização em Setores

Cada setor arquiva 512 bytes de dados, mas o computador sabe onde estão guardados esses dados? E como ele consegue identificar esses endereços?

▶ Cabeçalho

- ▶ A primeira informação do cabeçalho são bytes de sincronia, conteúdo informação referente ao espaço entre um setor e outro.
- ▶ O segundo dado encontrado refere-se a identidade do setor, seu número, face e cilindro a que pertence (endereço).
- ▶ Depois, o disco passa a uma área onde estão registrados dados para conferência. Isto inclui um FLAG para indicar se o setor está danificado ou não, e bits de paridade sobre as informações do cabeçalho.

▶ Dados

- ▶ Somente se tudo estiver correto no cabeçalho, chegamos as informações gravadas, os 512 bytes

▶ ECC (error control code)

- ▶ O disco avança mais um pouco e chega a um campo específico onde estão registradas informações sobre total de bytes gravados, e bits de verificação de erro dos dados.

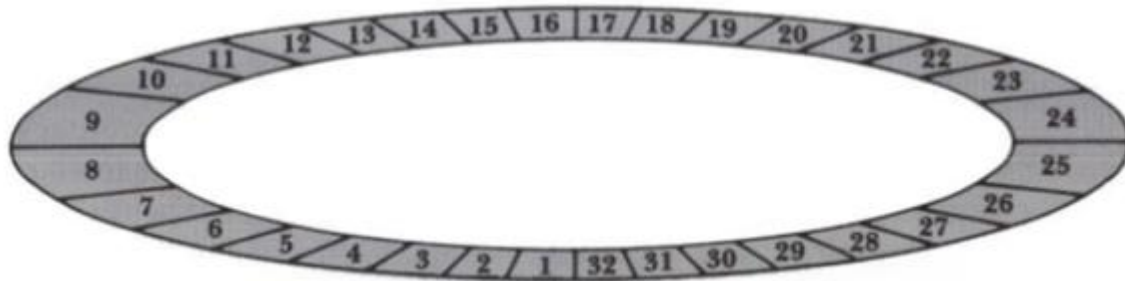
Organização em Setores

- ▶ Imagina-se que se o registro de dados ocupa mais que 512 bytes, então deve utilizar setores adjacentes.
- ▶ Essa é uma visão lógica adequada, mas fisicamente pode não ser uma boa forma de organização.
- ▶ O problema é que nem sempre é possível ler continuamente setores fisicamente adjacentes:
 - ▶ O controlador de disco, após ler os dados de um setor, pode precisar processar a informação recebida antes de estar pronto para aceitar mais informações.
 - ▶ Se dois setores logicamente adjacentes também estão fisicamente adjacentes, o controlador poderia perder a leitura do próximo setor (pois o disco continua em movimento de rotação), e teria que esperar uma outra rotação inteira do disco para acessar o setor perdido.

Organização em Setores: *Interleaving*

Interleaving

O interleaving é indicado para os casos em que a recuperação da informação é mais rápida que o tratamento da informação pela controladora.

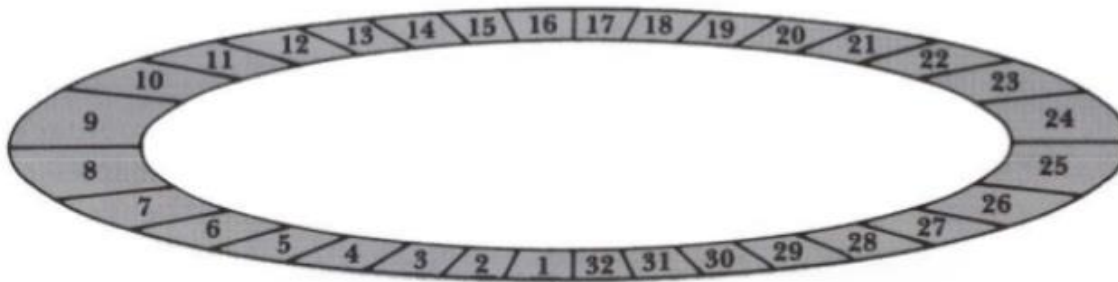


**interleaving factor: 1
(adjacentes)**

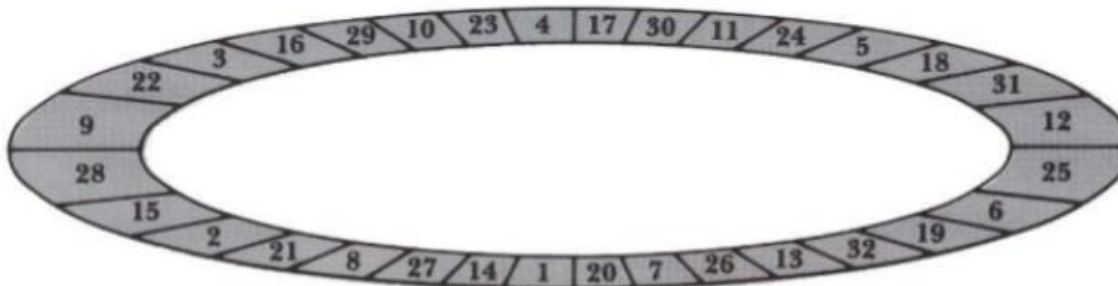
Organização em Setores: *Interleaving*

Interleaving

O interleaving é indicado para os casos em que a recuperação da informação é mais rápida que o tratamento da informação pela controladora.

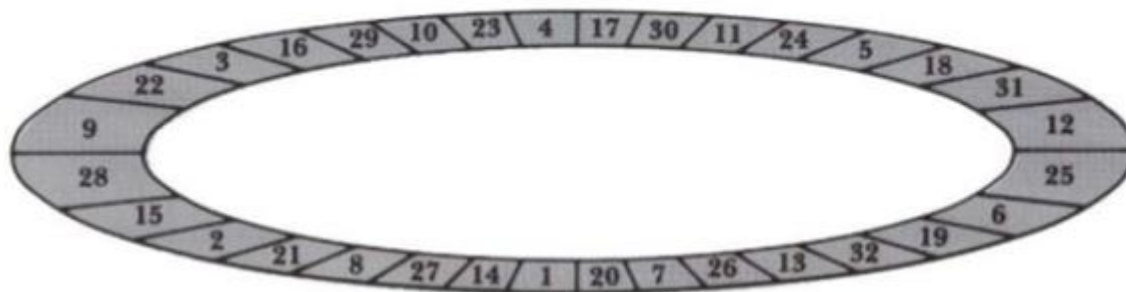


**interleaving factor: 1
(adjacentes)**



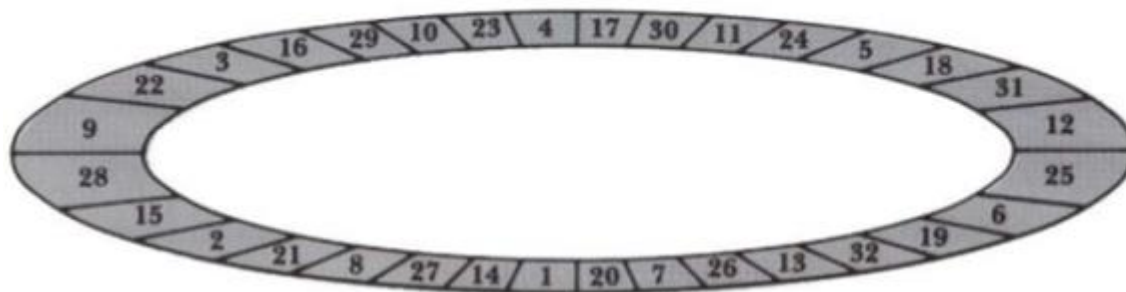
interleaving factor: 5

Quantas revoluções são necessárias para ler os 32 setores de uma trilha?



interleaving factor: 5

Quantas revoluções são necessárias para ler os 32 setores de uma trilha?



interleaving factor: 5

- Que melhora bastante, respeito de 32 revoluções num interleaving de fator 1.

Interleaving

- ▶ A tecnologia tem avançado de modo que discos de alto desempenho oferecem interleaving factors 1:1. Ou seja, a velocidade do controlador aumentou tanto que o interleaving não é mais necessário. Obs.: A interface ST-506/ST-412 (discos antigos, dos primeiros IBM PCs e Apple Macintoshes) necessitava de interleave. Essa realidade mudou, por volta dos anos 90 quando os padrões IDE/SCSI(ATA) surgiram.

"On modern disk drives, the interleave setting is always 1:1. Controller too slow? Ha! Today's controllers are so fast, much of the time they sit around waiting for the platters, tapping their virtual fingers.

How did this situation come to change so drastically in 15 years? Well, it's pretty simple. The spindle speed of a hard disk has increased from 3,600 RPM on the first hard disks, to today's standards of 5,400 to 15,000 RPM. An increase in speed of 50% to 132%.

The faster spindle speed means that much less time for the controller to be ready before the next physical sector comes under the head. However, look at what processing power has done in the same time frame: CPUs have gone from 4.77 MHz speeds to the environs of 1 GHz; an increase of over 20,000%! The speed of other chips in the PC and its peripherals have similarly gotten faster by many multiples."

Organização em setores

Organização de trilhas em setores

Além da organização física dos pratos em trilhas e setores, há uma organização lógica determinada pela formatação lógica e mantida pelo gerenciador de arquivos (file manager) dos sistemas operacionais, visando aumentar o desempenho, essa organização é denominada de: **Cluster** (um grupo de setores).

- ▶ Um cluster (também chamado de agrupamento) é a menor parte reconhecida pelo sistema operacional, e pode ser formado por vários setores.
- ▶ Um arquivo com um número de bytes maior que o tamanho do cluster, ao ser gravado no disco, é distribuído em vários clusters. Porém um cluster não pode pertencer a mais de um arquivo.

Exemplo: tamanho do setor: 512 bytes, tamanho do cluster: 4 setores

Organização em Clusters

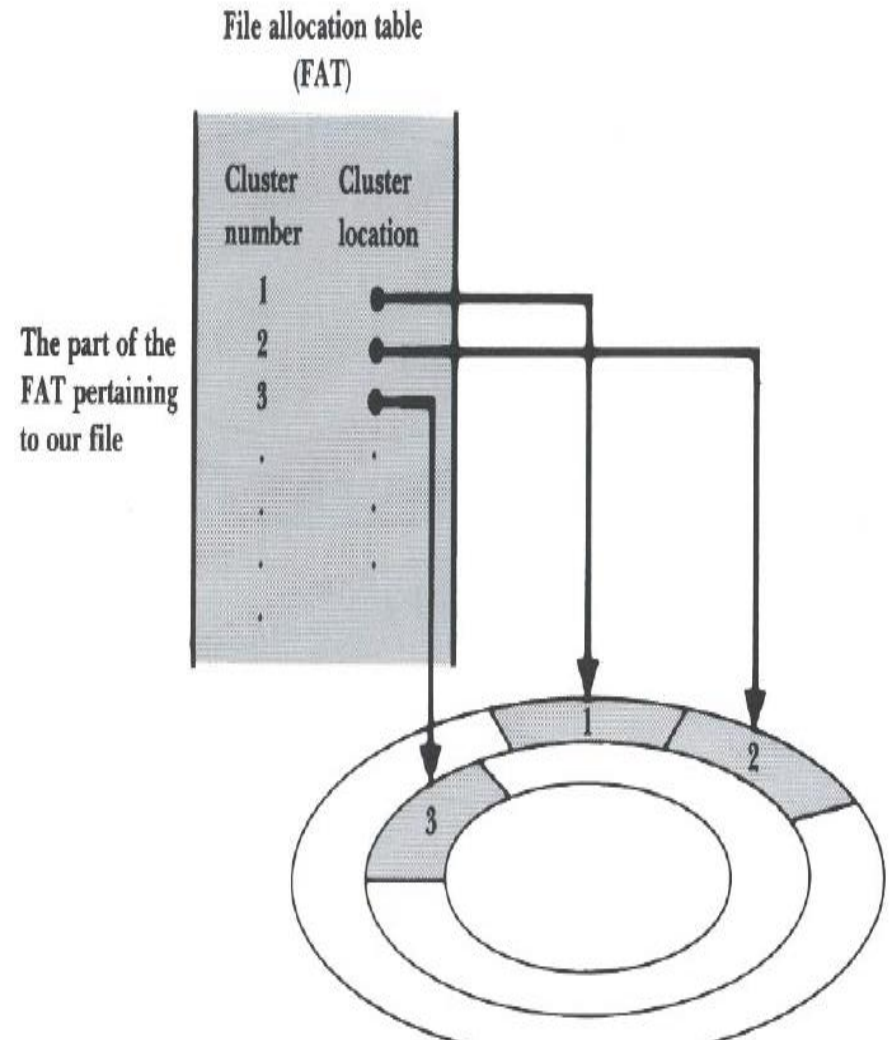
Gerenciador de arquivos

O gerenciador de arquivos mapeia as partes lógicas do arquivo e as partes físicas do mesmo. Quando um programa acessa um arquivo, o file manager (gerenciador de arquivos) deve associar o arquivo lógico às suas posições físicas.

- ▶ O gerenciador de arquivos (GA) acessa o arquivo como uma série de clusters de setores. Um cluster consiste de vários setores (logicamente) contíguos, (a contiguidade física depende do interleaving fator).
- ▶ O GA aloca um número inteiro de cluster para um arquivo.
- ▶ Cada cluster do disco é usado somente para um único arquivo, ou seja, num mesmo cluster, não pode haver informações sobre mais de um arquivo.

Organização de Clusters: Tabela FAT

- ▶ Considerando que um cluster associado a um arquivo foi encontrado, todos os seus setores podem ser acessados sem necessidade de um seeking adicional.
- ▶ Essa organização por setores/clusters é gerenciada através de um File Allocation Table (FAT). Nessa tabela, cada entrada dá a localização física do cluster associado a um certo arquivo lógico.



Organização em Clusters

Decidindo o tamanho de um Cluster

Em muitos sistemas, o administrador pode decidir o tamanho de setores/cluster. Ex.: No VAX (BSD Unix e OpenVMS) o número de setores default (3 setores (512 bytes)/cluster) pode ser mudado para qualquer número entre 1 e 65535.

- ▶ A vantagem de se ter clusters com um maior número de setores é a possibilidade de ler mais setores sem precisar de operações de seeking adicionais.
- ▶ Desvantagem ter muito espaço não utilizado no disco (fragmentação interna).

Organização em Clusters: Extents

Se houver bastante espaço disponível no disco, é possível fazer com que um arquivo seja formado apenas por clusters consecutivos. Denotamos que o arquivo consistiria de um extent (um único fragmento).

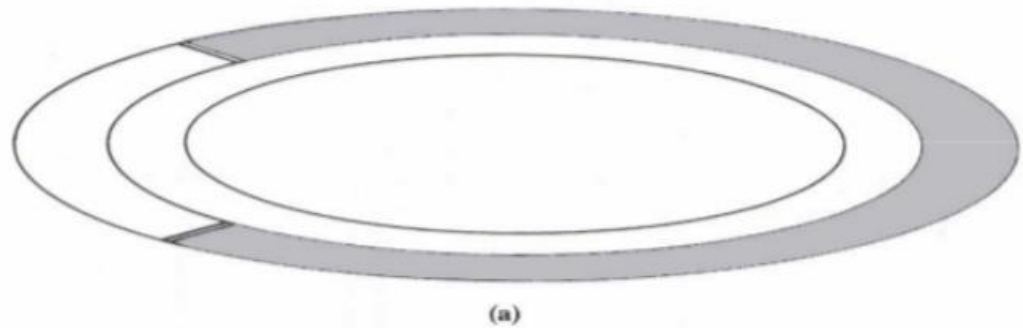
- ▶ O arquivo pode ser processado com o mínimo de tempo de busca, ou seja, com um único seeking.
- ▶ Ao se adicionar novos clusters, o GA tenta manter o arquivo ocupando um único extent, mas se não for possível, então o arquivo é dividido em mais extents.

Conforme o número de extents num arquivo aumenta, o arquivo fica mais espalhado no disco, e maior é a quantidade de seekings necessários para processá-lo.

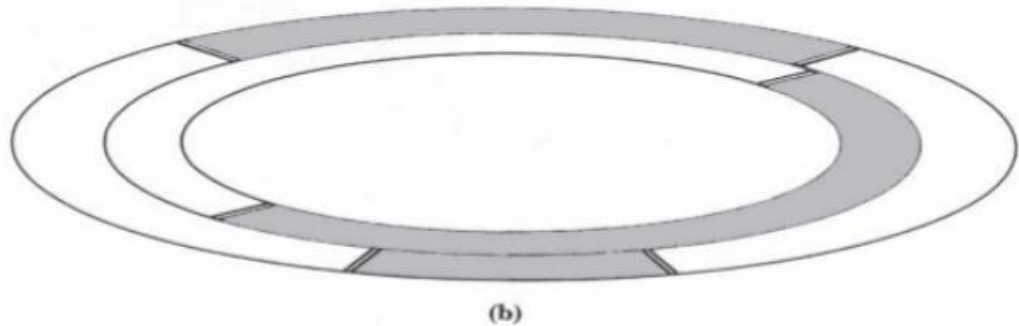
Organização de clusters: Extents

Figura: Distribuição do Arquivo em Extents

(a) Arquivo em um único extent.



(b) Arquivo dividido em vários extents.



Organização em Clusters: Fragmentação Interna

Fragmentação Interna em Clusters

Um cluster é a menor porção alocável para um arquivo.

Se o número de bytes do arquivo não é múltiplo do tamanho do cluster, então há uma perda: a perda ocorre na última extensão do arquivo. Por exemplo, se o arquivo tem apenas 500 bytes, com clusters de 3 setores de 512 bytes, então a perda é de 1036 bytes.

Quando usar clusters grandes?

- ▶ Quando o disco contém arquivos grandes a serem processados sequencialmente.

Quando usar clusters pequenos?

- ▶ Quando o disco tem pequenos arquivos e/ou arquivos a serem acessados aleatoriamente (acesso direto).

Organização de clusters: Fragmentação Interna

O tamanho do cluster pode diferir do tamanho dos registros de um arquivo.

Quando isso ocorre, existem duas alternativas possíveis:

- ▶ Armazenar um número inteiro de registros por cluster (unspanned records). Neste caso, a recuperação de um registro exige a busca de apenas um cluster, mas uma área não utilizada é mantida em cada cluster. Essa perda de espaço é denominada **fragmentação interna**.
- ▶ Divide-se os registros em vários clusters (spanned records). Neste caso, não se perde espaço com fragmentação interna, mas podem existir registros cuja recuperação exige a busca de 2 cluster.

Organização em Cluster: Fragmentação Interna

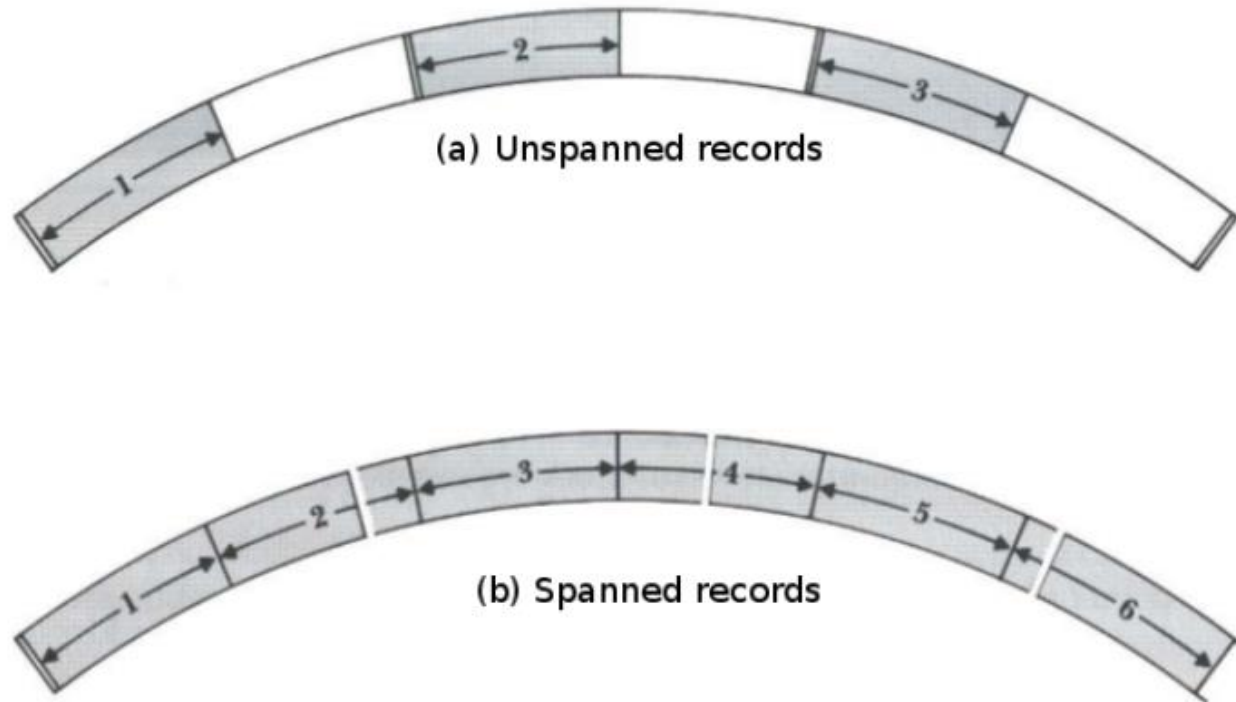


Figura: (a) unspanned records Vs (b) spanned records

Organização em Clusters: Tamanho dos Clusters

Tabela: Tamanho padrão dos clusters para discos sob determinados sistemas de arquivos do Windows XP profissional

Volume Size	FAT 16 Cluster Size	FAT 32 Cluster Size	NTFS Cluster Size
7MB-16 MB	2KB	Not Supported	512 bytes
17MB-32MB	512 bytes	Not Supported	512 bytes
33MB-64MB	1KB	512 bytes	512 bytes
65MB-128MB	2KB	1KB	512 bytes
129MB-256MB	4KB	2KB	512 bytes
257MB-512MB	8KB	4KB	512 bytes
513MB-1024MB	16KB	4KB	1KB
1025MB-2GB	32KB	4KB	2KB
2GB-4GB	64KB	4KB	4KB
4GB-8GB	Not Supported	4KB	4KB
8GB-16GB	Not Supported	8KB	4KB
16GB-32GB	Not Supported	16KB	4KB
32GB-2TB	Not Supported	Not Supported*	4KB

Nota

O FAT32 pode suportar até 2 Terabytes, porém o Windows 2000 tem a limitação de até 32 GB.

No NTFS, usando o tamanho padrão (4KB) de cluster pode-se criar um volume de até 16 terabytes.

Pode-se criar volumes até 256 Terabytes usando o máximo de cluster de 64 KB.

Ext3 vs Ext4 vs NTFS

Tabela: Comparativo entre os diversos Sistemas de Arquivos

	ext3	ext4	NTFS
Stand for	Third Extended File System	Fourth Extended File System	New Technology File System
Original OS	Linux	Linux	Microsoft
Max Volume Size	256TB	1 EB limited to 16GB	256 TB
Max File Size	16GB-2TB	16TB	16EB
Max Filename Length	254 Bytes	256 Bytes	255 UTF-16 code units
Journaling	Yes	Yes	Yes

Pesquisa

Pesquise mais detalhes sobre as diferenças entre FAT16, FAT32, NTFS, EXT2, EXT3 e EXT4.

Organização em Blocos

Também é possível organizar as trilhas em **Blocos**, ao invés de Clusters.

Blocos

- ▶ Tamanho variável
 - ▶ Definido pelo usuário
- ▶ A quantidade de dados transferidos em uma operação de I/O pode variar e é dada pelo tamanho do bloco.
 - ▶ Depende das necessidades do projetista de software (não de hardware)
- ▶ Bloco tem uma quantidade inteira de registros: é um registro físico.
 - ▶ A unidade de transferência entre memória e disco é o bloco
 - ▶ Blocking fator: Número de registros por bloco em um arquivo
 - ▶ Se tivermos registros de 300 bytes, podemos definir blocos de 300 bytes (ou um valor múltiplo de 300!).
 - ▶ Não existe mais o problema de fragmentação interna.
 - ▶ Não temos mais “quebra” de registros entre setores (uso de mais de um setor por registro). Isso porque os blocos podem variar em tamanho para se ajustar à organização lógica dos dados.

Setores vs Blocos

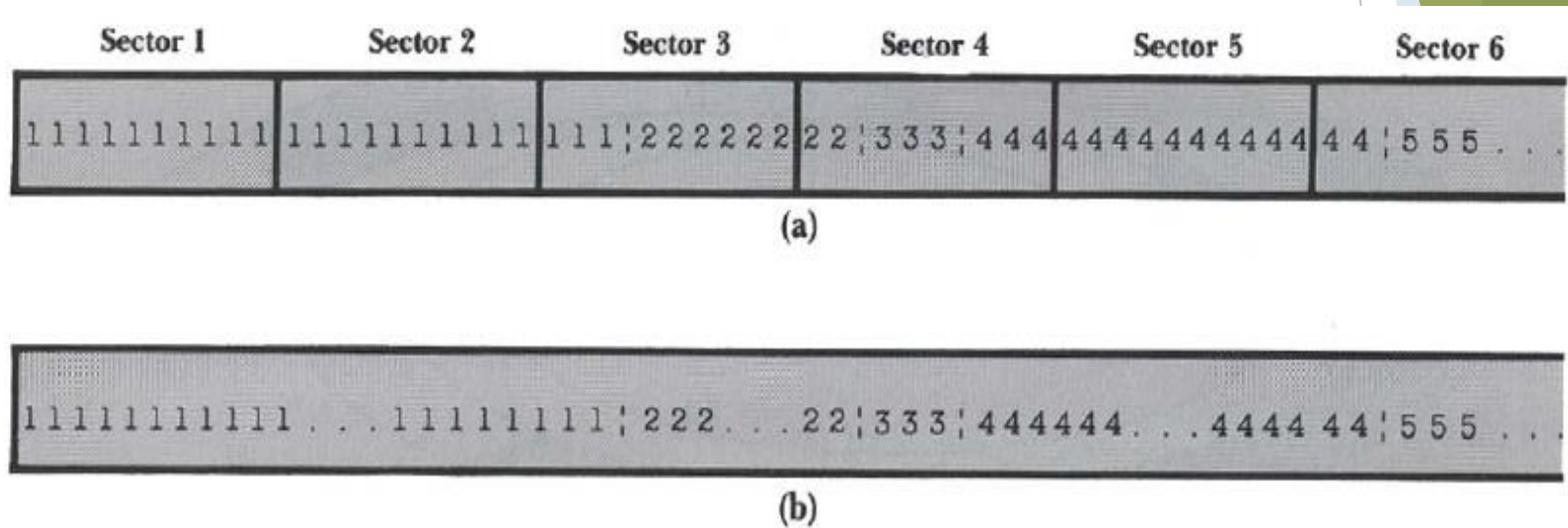


Figura: Organização em setores(a). Organização em Blocos(b).

Organização em blocos

- ▶ Em um mesmo arquivo pode-se ter blocos de tamanho variável.
- ▶ Cada bloco é precedido por um cabeçalho (sub-bloco) contendo:
 1. O tamanho do bloco
 2. A chave do último registro do bloco de registros.
- ▶ O controlador de disco pode assim, pesquisar uma trilha até encontrar o bloco requerido.
- ▶ Há controladores que permitem endereçamento por setor ou por bloco.

Organização em Blocos

- Esquema de endereçamento: cada bloco é normalmente dividido em sub-blocos, contendo informações extra sobre os dados do bloco.
 1. Count-subblock: Número de bytes no bloco
 2. Key-subblock: chave associada ao último registro. (A controladora de disco pode buscar por uma chave sem carregá-la para a memória principal).

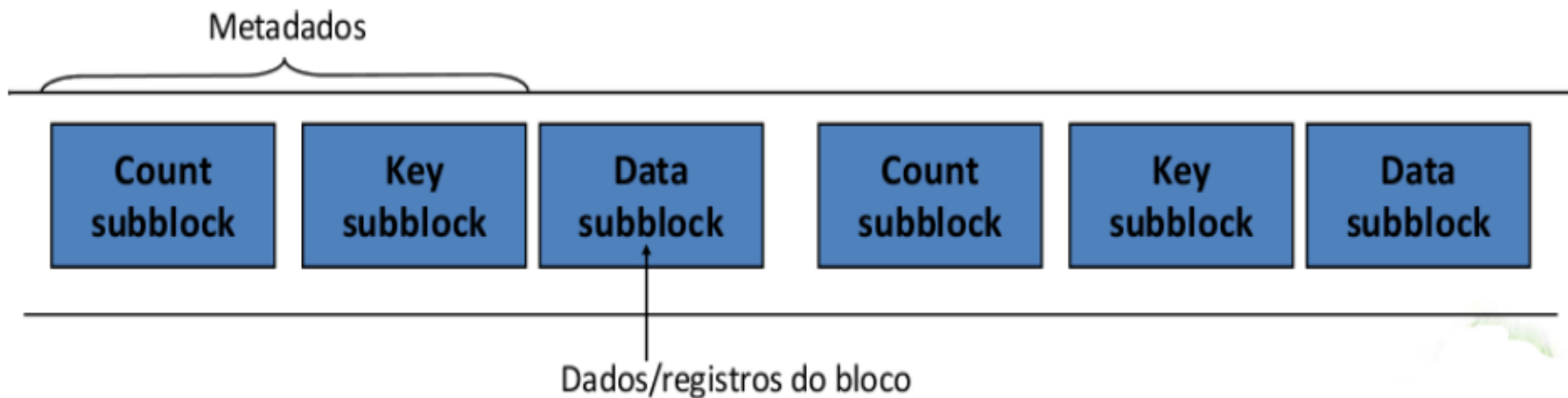


Figura: Organização de um bloco

Organização em blocos

- ▶ Quando o esquema key sub-bloco é utilizado, o controlador de disco pode buscar diretamente pela chave em operações I/O.
- ▶ O programa poderia requisitar ao controlador de disco para buscar um bloco com uma dada chave, operação feita sem necessidade de processamento em RAM, ao contrário do que ocorre com a organização por setores, na qual a busca por uma chave requer que o setor seja carregado na RAM.
- ▶ O resultado é uma busca mais eficiente.

Nondata Overhead: Setores

- ▶ Na organização de trilhas, usando blocos ou setores, há um gasto fixo de espaço em disco para armazenar informações de controle (extras), que chamamos de Nondata Overhead.
- ▶ Nos discos organizados por setor, temos que ter marcas de início/fim de setor, marcas de sincronização, se o setor é válido ou danificado, entre outras, são inseridas no processo de formatação. Esse overhead é invisível para o projetista.
- ▶ **Nondata Overhead = Espaço sem dados**

Nondata Overhead: Blocos

- ▶ Em discos organizados por blocos, também temos Nondata Overhead. Alguns dos quais o programador precisa conhecer, como **count sub-bloco** e **key sub-bloco**.
- ▶ Um número maior de informações extras (nondata overhead), que não é dado, pode ser colocada no disco, quando comparado ao esquema de organização por setores.

Vantagem

Permite ao programador escolher a organização dos dados, de forma a explorar melhor a capacidade do disco e otimizar o acesso.

Desvantagem

Exige do programador (e SO) um conhecimento da organização dos dados.

Nondata Overhead

Exemplo

Seja um disco organizado em blocos com 50000 bytes/trilha no qual o overhead/bloco é 300 bytes. Queremos armazenar um arquivo contendo registros de 150 bytes. Determinar o número de registros armazenados em uma trilha, para fatores de bloco iguais a 10 e 60.

- ▶ Para um fator de blocagem igual a 10: Se 10 registros de 150 bytes são armazenados por bloco, cada bloco terá 1500 bytes de dados, mas usará 1800 bytes no total, devido ao *overhead*.
- ▶ Então, o número de blocos que podem ser colocados em uma trilha é dado por:
$$\frac{50000}{1800} = 27.78, \text{ portanto, } 27 \text{ blocos, ou } 270 \text{ registros.}$$
- ▶ Para um fator de blocagem igual a 60: Se são 60 registros de 150 bytes, cada bloco armazena 9000 bytes de dados e 9300 no total (pelo overhead). Então, o número de blocos por trilha fica:
 $(50000/9300)=5.38$, por tanto, 5 blocos ou 300 registros

Nondata Overhead

- ▶ Blocos maiores, em geral, levam a um **uso mais eficiente do espaço**, pois há um número menor de bytes de **overhead**, em comparação com o número de bytes ocupados por dados.
- ▶ Mas nem sempre os blocos maiores garantem ganhos: a **fragmentação interna dentro da trilha** ocorre, pois as trilhas têm tamanho fixo e só podemos por um número inteiro de blocos dentro de uma trilha, e sempre sobra espaço no final.

Próxima Aula

- ▶ Memória Secundaria (continuação).