





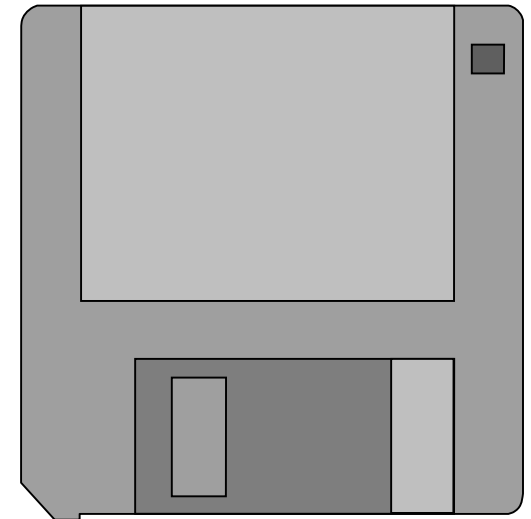
# Armazenamento de Informações

**I. O Nível Físico**

II. O Nível Lógico

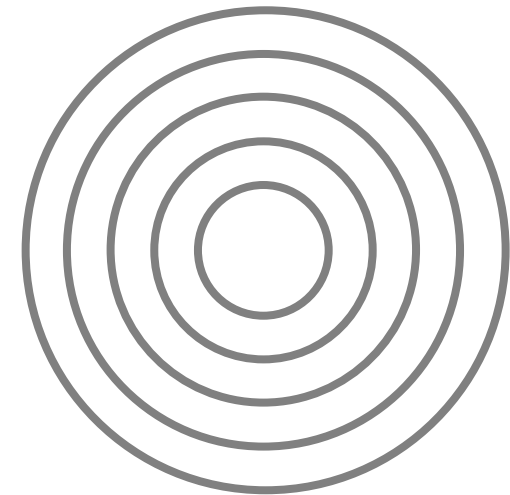
# Mídias Magnéticas

- Partículas magnetizadas de óxido de ferro
- *Discos* - superfícies em movimento circular contínuo



# Trilhas

- Circunferências Concêntricas
- Múltiplas trilhas por superfície
- Cabeças se movimentam entre trilhas



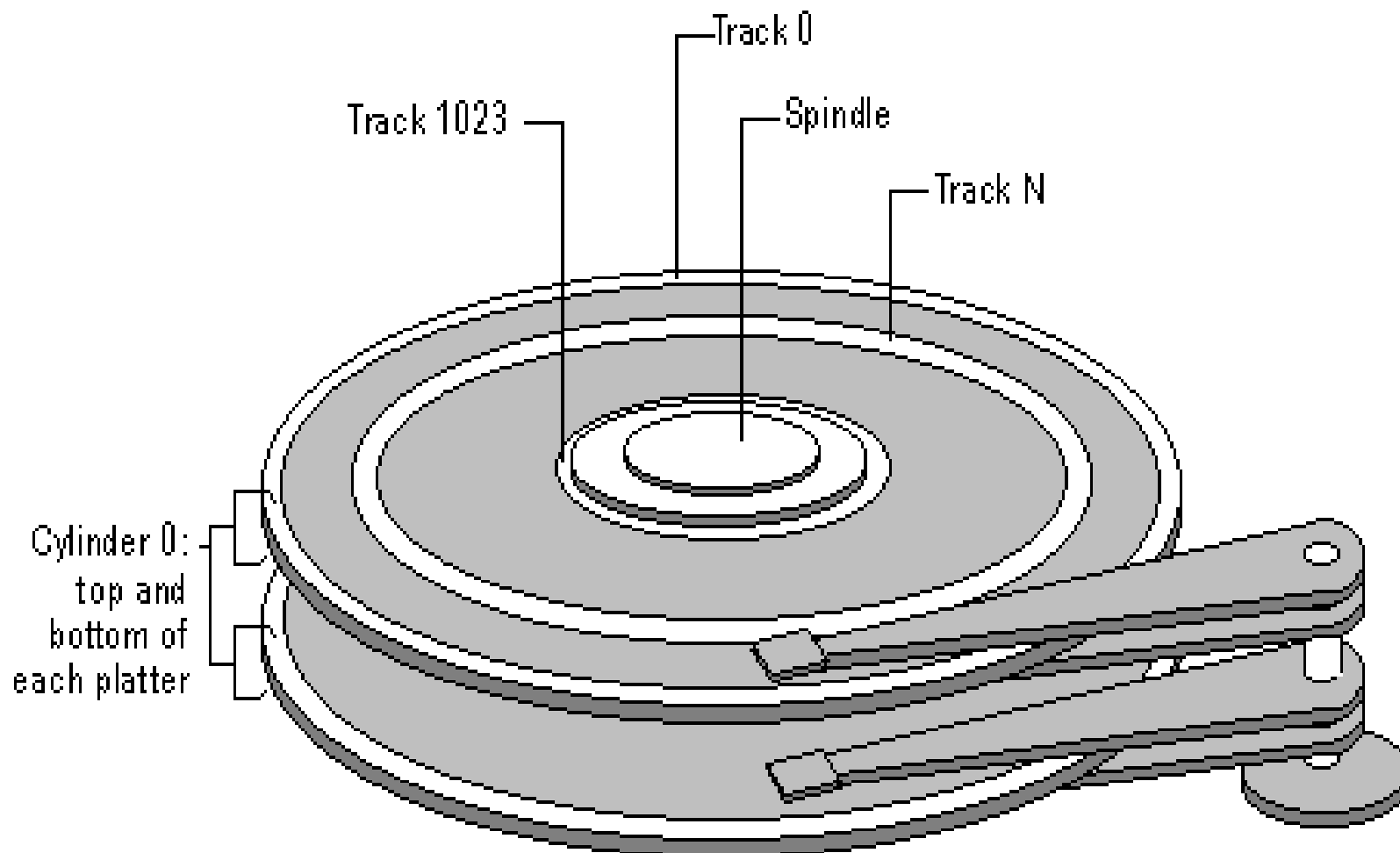
## Discos com Múltiplas Faces

- Como antigos discos de vinil
- Multiplicam a capacidade
- As cabeças perfazem pinças

# Cilindros

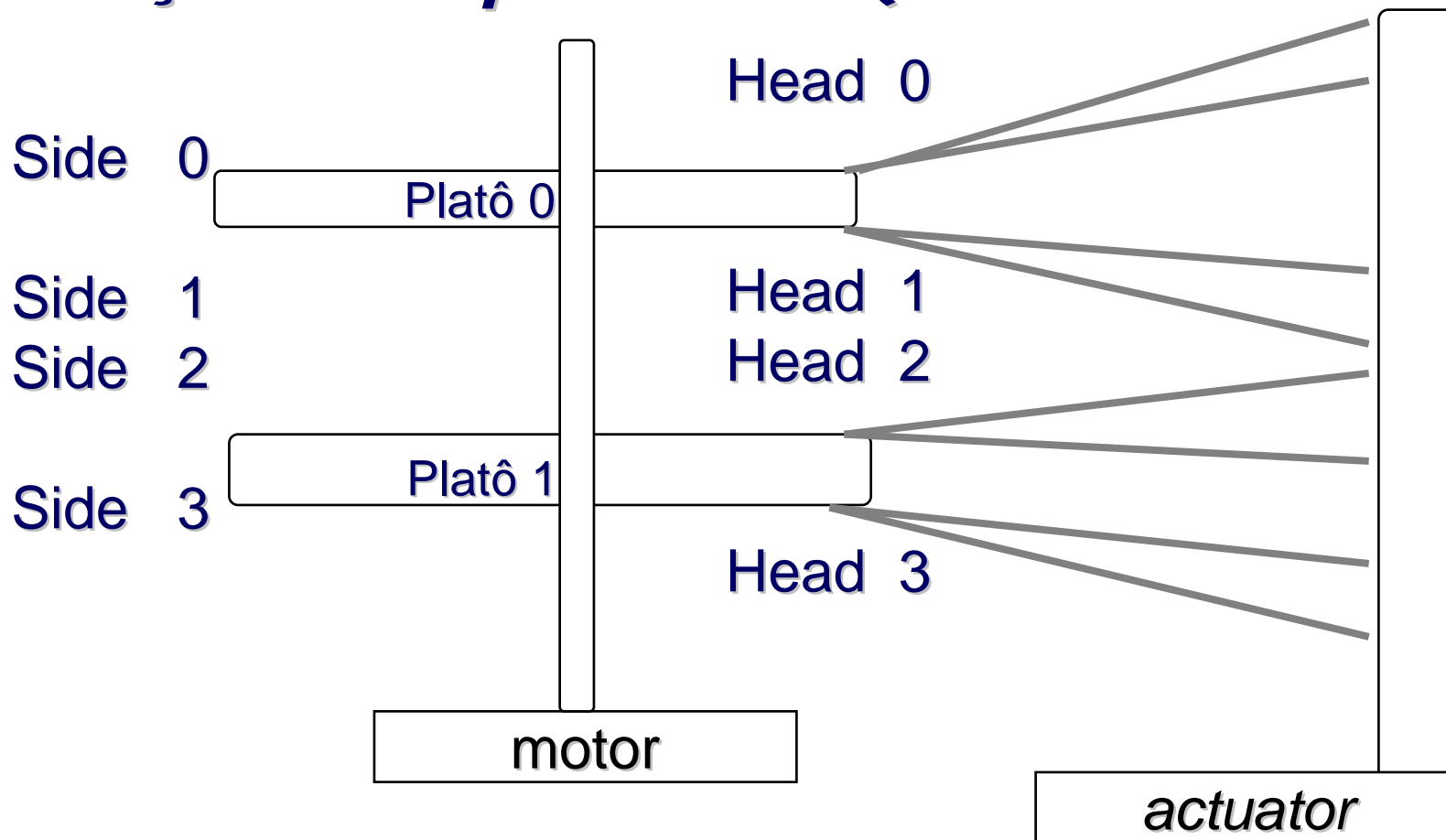
- Múltiplas faces otimizam as velocidades de leitura e gravação
- As trilhas alinhadas (acima e abaixo da mídia) formam CILINDROS
  - As mais externas formam o Cilindro 0
    - Superior: Trilha 0 , lado 0
    - Inferior : Trilha 0, lado 1

# Trilhas e Cilindros



# Discos Rígidos

## *Cabeças e Superfícies (heads & sides)*





## Disquetes - Aspectos Físicos

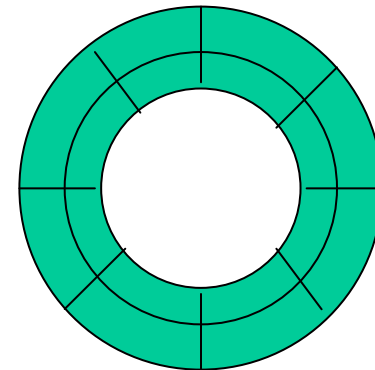
- Cabeças em contato com a superfície
- Vantagem
  - Sinal Maximizado
- Desvantagens
  - Baixas velocidades devido ao atrito
  - Desgaste

# Discos Rígidos - Aspectos Físicos

- Cabeças próximas a superfícies do disco
  - Velocidades altas
  - Preservação das mídias
- Camadas de ar de 10 micropolegadas
- Ausência de partículas de poeira, com montagem em “salas limpas”
- Múltiplas superfícies e cabeças

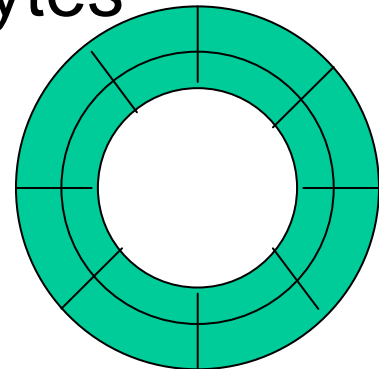
# Setores

- Unidades atômicas, blocos para Leitura Gravação
- Desde PC-DOS v 1.0 setores tem 512 Bytes



# Endereçamento por Setores

- Menor unidade endereçável
- Endereçamento escrito durante formatação física, sendo essencial para recuperação de dados
- Se endereço é corrompido ou inexistente ocorrem erros (“*sector not found*”, etc.)
- Desde PC-DOS v 1.0 setores tem 512 bytes



# Tecnologias de Discos Rígidos

- IDE (Integrated Drive Electronics)
- SCSI (Small Computer System Interface)
- SATA (Serial ATA)
- FORMATAÇÃO FÍSICA FEITA PELO FABRICANTE

# IDE -Integrated Device Eletronics

- Dispositivo com controladora integrada
- ATA - AT Attachment
- Conector de 40 pinos, subconjunto dos 98 pinos do barramento ISA -16 bits

# AT Attachment

- Limites ATA
  - Até 2002, 28 bits para endereçamento
  - limite de 268435455 setores = 128 Gb
- ATA-6
  - 44 bits
  - Limite de 144 Petabytes



# Evolução do IDE

Nome da Interface	ATA -2	Ultra-ATA ou ATA- 33 ou Ultra DMA	ATA-66 ou Ultra DMA 66
Taxas máximas de transferência (MB/seg)	16.6	33	66
Dispositivos por controladora	2	2	2





# SATA

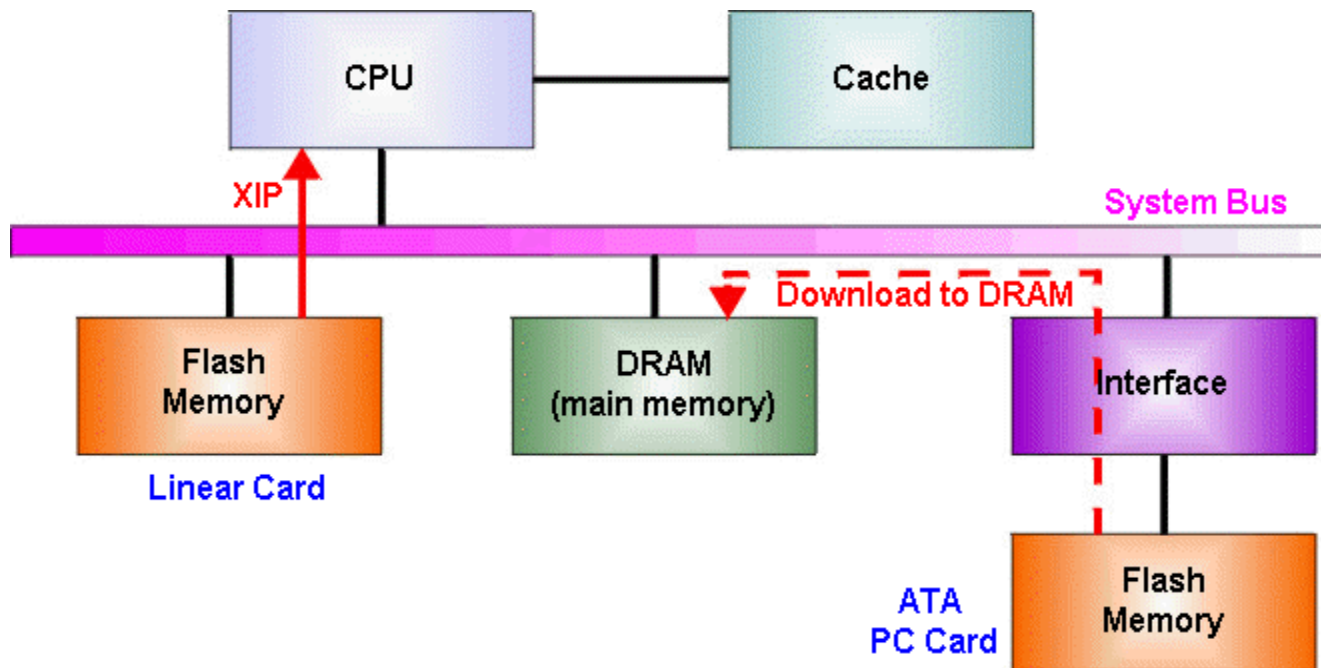
<b>Nome da Interface</b>	<b>SATA</b>	<b>SATA II</b>
<b>Taxas máximas de transferência (MB/seg)</b>	<b>150</b>	<b>300</b>
<b>Dispositivos por controladora</b>	<b>1</b>	<b>1</b>

# Outros Dispositivos – Memórias Flash

- Não voláteis, podendo ser apagadas por seções de uma só vez.
- *PC cards*, ou linear cards:  
Conectadas diretamente ao bus do PC, com acesso direto à CPU. Este modo é conhecido como '*execute in place*' (EIP) mode , oferecendo acesso para leitura randômica rápida (para dados dos sistemas, como código de boot, BIOS, fontes, etc.);
- *ATA flash PC cards*:  
Conectadas à CPU via interface ATA, como se faz com outros dispositivos de armazenamento como os discos rígidos. A CPU trata o cartão como equivalente aos HD, portanto este deve ser endereçável por meio de blocos.



# Outros Dispositivos – Memórias Flash



## Estrutura de Discos RAID

- RAID – múltiplos discos provêm
  - confiabilidade
  - disponibilidade
  - por meio de redundância
  
- RAID – seis tipos



## RAID (cont)

- Múltiplos discos em um mesmo trabalho cooperativo.
- *Disk striping* – grupo de discos como uma única unidade de armazenamento.
  - Espelhamento (*mirroring*) – discos em duplicata .
  - Paridade de blocos – recuperação de falhas com menor redundância.

## RAID (cont)

- Nível 0 : blocos espalhados entre múltiplos discos (*data striping*), podendo ser acessados em paralelo. Não há redundância de dados.
- Nível 1 : discos espelhados (*disk mirroring*). Dados são gravados simultaneamente, tornando o sistema tolerante a falhas em uma mídia.
- Nível 3 : como no nível 0, com o acréscimo de um disco dedicado para armazenar dados para correção de erros
- Level 5 : Provê espalhamento de dados no nível de bytes, assim com o espalhamento de informações para correção de erros, entre as mídias existentes. Oferece a melhor performance juntamente com a possibilidade de recuperar falhas em mídias.

# Níveis de RAID



(a) RAID 0: non-redundant striping



(b) RAID 1: mirrored disks



(c) RAID 2: memory-style error-correcting codes



(d) RAID 3: bit-interleaved Parity



(e) RAID 4: block-interleaved parity

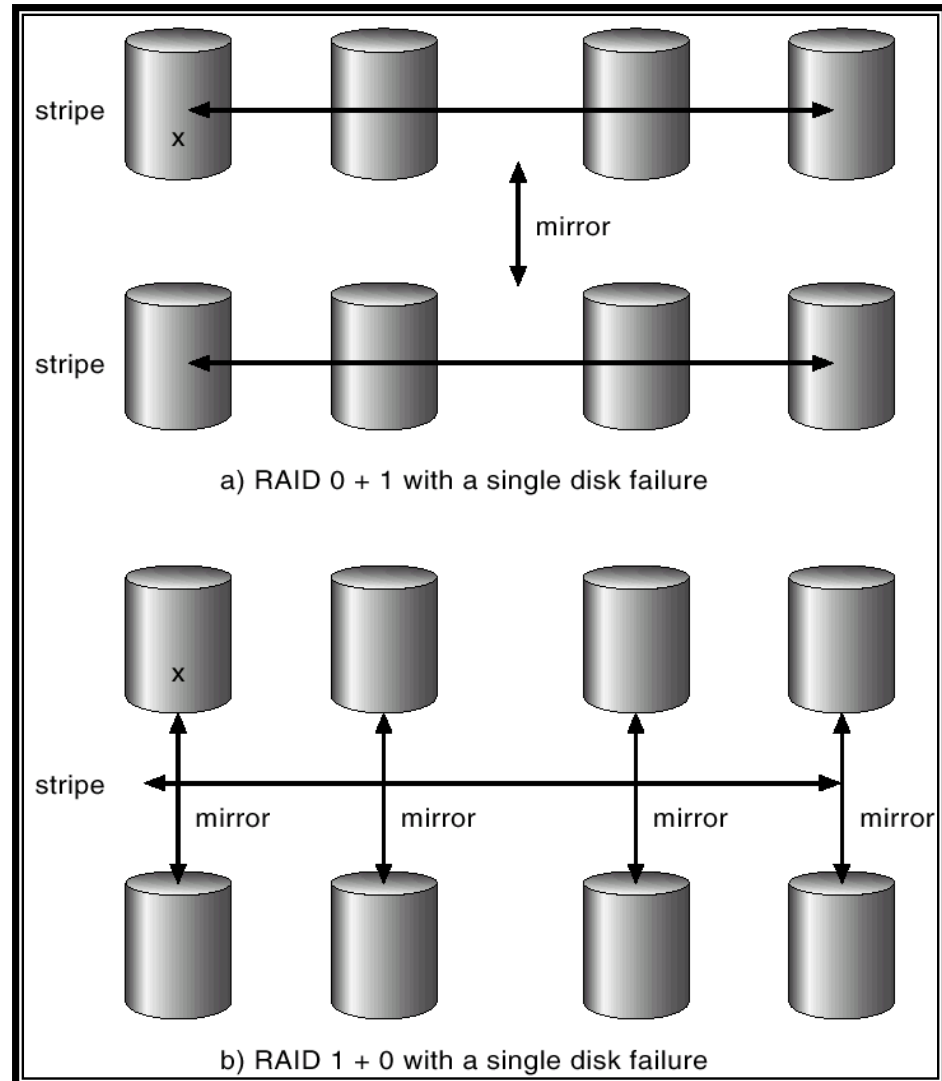


(f) RAID 5: block-Interleaved distributed parity



(g) RAID 6: P + Q redundancy

# RAID (0 + 1) and (1 + 0)







# Armazenamento de Informações

I. O Nível Físico

**II. O Nível Lógico**

## Estrutura de Discos

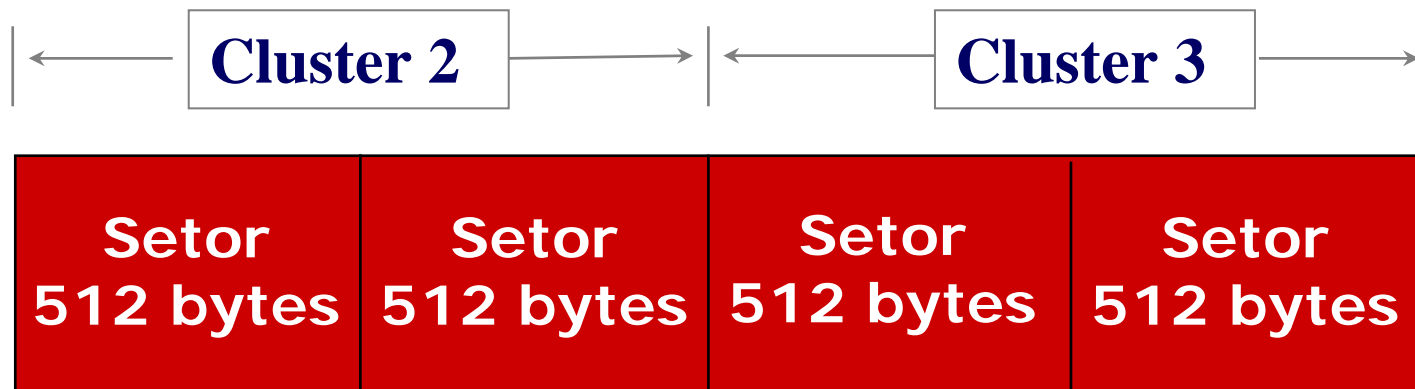
- Discos são endereçados como vetores de *blocos lógicos*, cada bloco sendo a menor unidade de transferência.
- Um vetor de blocos representa um mapeamento seqüencial do espaço em disco
  - Setor 0 é o primeiro setor da primeira trilha no cilindro mais externo.
  - Mapeamento prossegue pela trilha, por outras trilhas do cilindro, e depois pelos demais cilindros em direção ao centro.

# Gerência de Discos

- Formatação física = dividir o disco em setores, de tal forma que uma controladora possa ler e gravar.
- Para começar a guardar arquivos, o s.o. necessita criar estruturas de dados para controle do disco:
  - Particionar o disco em grupos de cilindros.
  - Formatação lógica – preparar o sistema de arquivos.
- Setor de Boot inicializa o sistema.
  - A seqüência de boot reside em ROM.
  - Neste setor, o código encontra e carrega o s.o.

# Formatação em Clusters

- Conjunto de setores contíguos
- A menor unidade de alocação de espaço em disco
- O sistema de arquivos enumera todos os clusters de uma partição
- Correspondem a entradas da FAT



## Estudo de Caso – DOS/FAT

- DOS – *Disk Operating System*
- FAT – *File Allocation Table* – tabela de alocação de arquivos

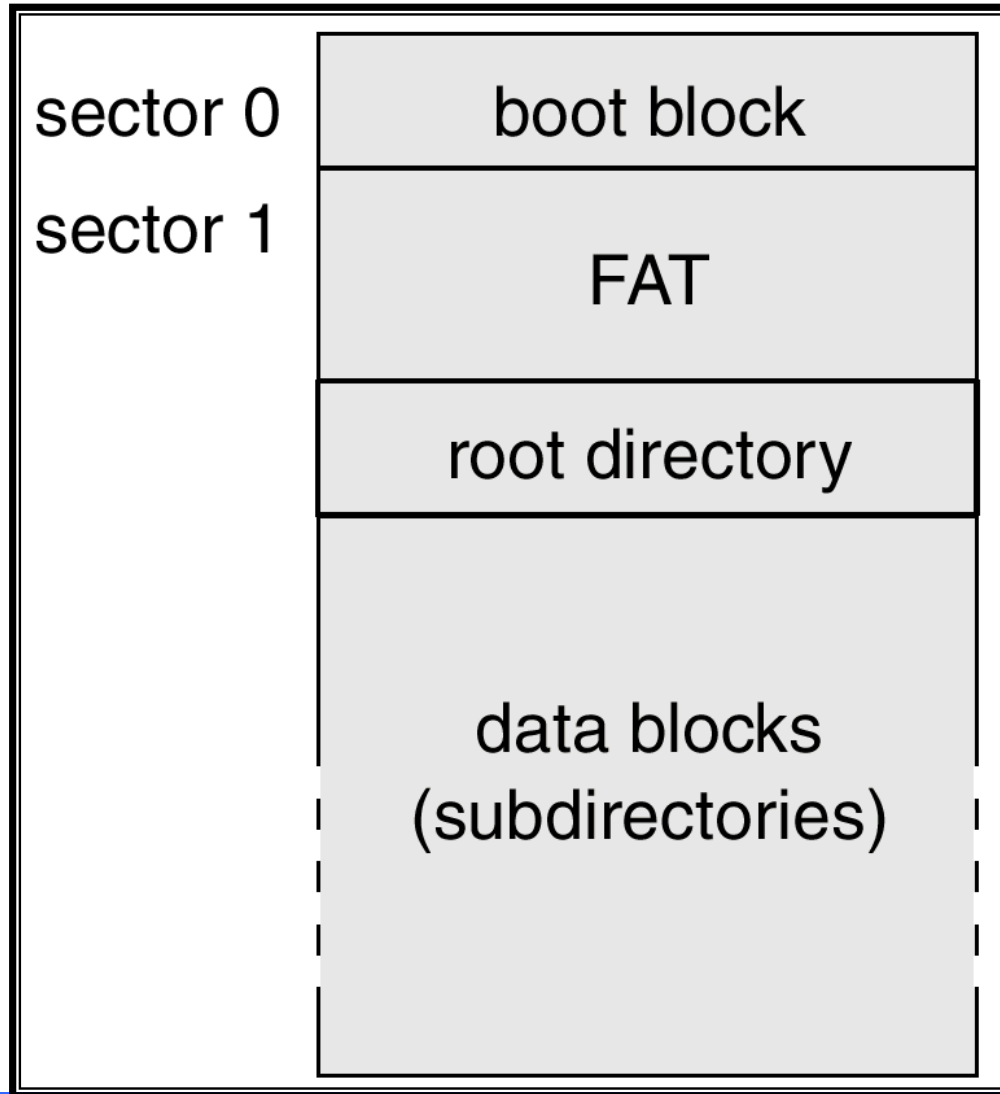
## DOS - Histórico

- DOS 1 – Não possuía diretórios
- DOS 2 - Introduziu diretórios hierárquicos e suporte para armazenagem de massa, iniciando o uso de discos rígidos externos. O sistema operava os discos utilizando device drivers carregados depois do boot, feito somente através de disquetes
- 1983 - IBM XT (eXTended) com discos de 10 MB acionados por controladoras inseridas nas interfaces do bus interno.

# DOS - Estruturas Lógicas

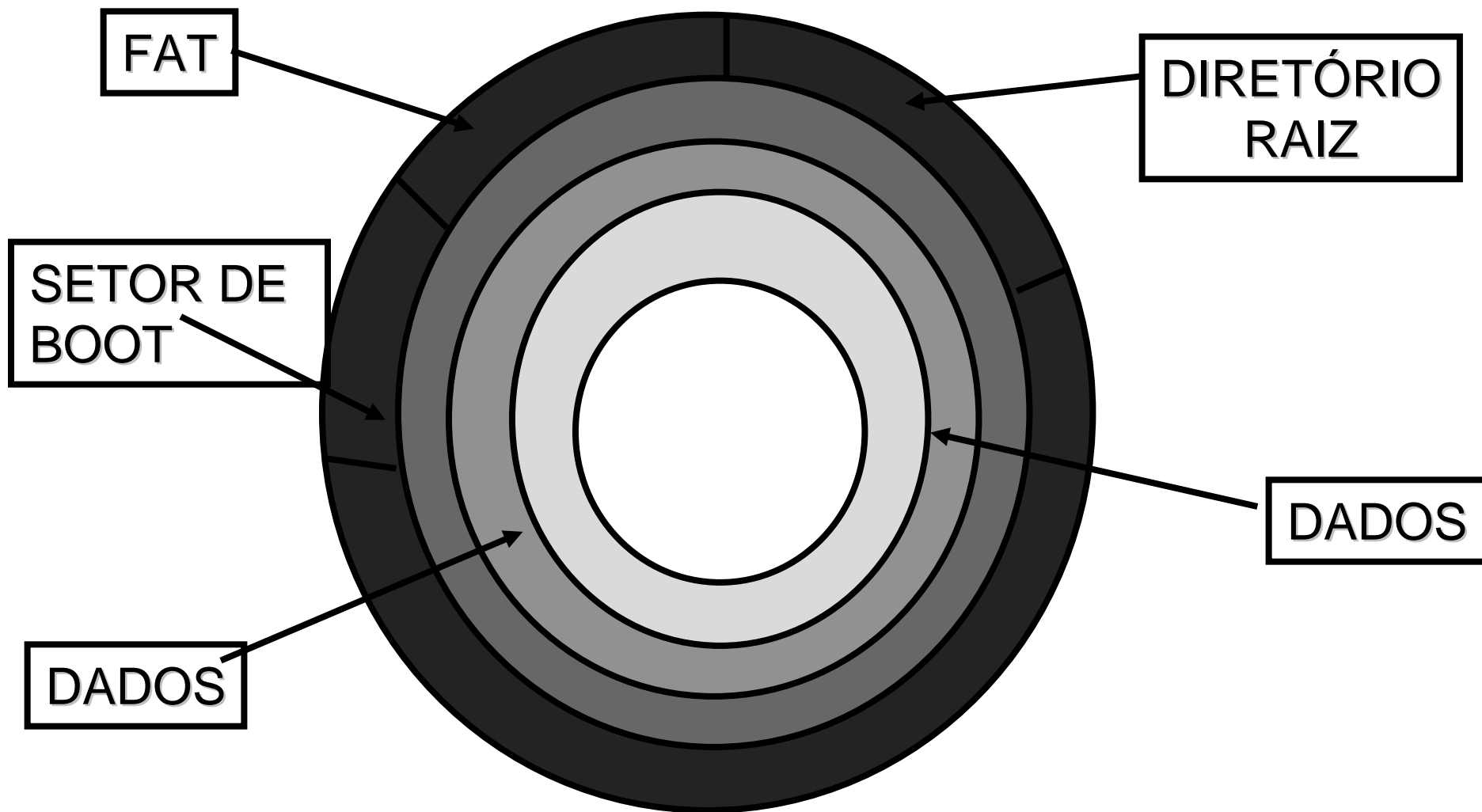
- 4 Áreas Principais
  - Setor de Boot
  - File Allocation Table (FAT)
  - Diretório Raiz
  - Área de Dados

# Layout DOS





# Áreas de Discos DOS



# Formatação DOS

- Apenas uma Partição de Boot permitida
- Se o DOS não reconhece alguma entrada na tabela de partições, o disco permanece inacessível
- Em discos rígidos, a formatação não sobrepõe dados de arquivos, **MAS APENAS METADADOS**

# *File Allocation Table - FAT*

- Sistema de Arquivos FAT
  - Diretórios
  - FAT – *File Allocation Table* – tabela de alocação de arquivos

## Diretório

- Área em disco que guarda informações sobre arquivos, como nome, tamanho, data e hora de modificação, localização no disco, etc
- Volumes DOS (disquetes, HDs) tem ao menos um diretório, o diretório raiz
- Todo arquivo possui uma entrada em diretório



## \ > Diretório Raiz

- Última parte da área de sistema
- Tamanho varia com tipo de disco
- Em disquetes de 1.44 MB , FORMAT cria um diretório raiz com 224 entradas
- Na maioria dos discos rígidos, são criados com 512 entradas
- Uma entrada se refere, basicamente a um arquivo ou a outro sub-diretório



# Entrada de Diretório

<b>DESCRIÇÃO</b>	<b>TAMANHO (BYTES)</b>	<b>FORMATO</b>
Nome do Arquivo	8	CARACTERES ASCII
EXTENSÃO	3	CARACTERES ASCII
ATRIBUTOS	1	ATRIBUTOS EM BITS
RESERVADO	10	NÃO USADO
HORA	2	CODIFICADO (2 BYTES)
DATA	2	CODIFICADO (2 BYTES)
PRIMEIRA ENT. FAT	2	2 BYTES
TAMANHO	4	INTEIRO LONGO (31 BITS)
<b>TOTAL</b>	<b>32</b>	



# Tipos de Entradas de Diretório

- Arquivo
- Subdiretório
- Exceto no diretório raiz, sempre haverá dois tipos de entradas especiais logo no início:
  - . indica o próprio diretório
  - .. indica o diretório superior na hierarquia

Disk Editor

Object Edit Link View Info Tools Help More>

Name	.Ext	ID	Size	Date	Time	Cluster	76	A	R	S	H	D	V
Cluster 191, Sector 1,115													
.		Dir	0	1-10-96	4:30 pm	191	-	-	-	-	D	-	
.		Dir	0	1-10-96	4:30 pm	0	-	-	-	-	D	-	
ANSI	SYS	File	9065	3-10-93	6:00 am	192	A	-	-	-	-	-	
APPEND	EXE	File	10774	3-10-93	6:00 am	197	A	-	-	-	-	-	
ATTRIB	EXE	File	11165	3-10-93	6:00 am	203	A	-	-	-	-	-	
BACKUP	EXE	File	36092	4-09-91	5:00 am	209	A	-	-	-	-	-	
CHKDSK	EXE	File	12907	3-10-93	6:00 am	227	A	-	-	-	-	-	
CHKLIST	CPS	File	2457	3-07-95	10:02 am	234	A	-	-	-	-	-	
CHKSTATE	SYS	File	41600	3-10-93	6:00 am	236	A	-	-	-	-	-	
CHOICE	COM	File	1754	3-10-93	6:00 am	257	A	-	-	-	-	-	
COMMAND	COM	File	52925	3-10-93	6:00 am	258	A	-	-	-	-	-	
COUNTRY	SYS	File	17066	3-10-93	6:00 am	284	A	-	-	-	-	-	
DBLSPACE	NOT	File	274484	3-10-93	6:00 am	293	A	-	-	-	-	-	
DEBUG	EXE	File	15715	3-10-93	6:00 am	428	A	-	-	-	-	-	
DEFRAG	EXE	File	75033	3-10-93	6:00 am	436	A	-	-	-	-	-	
DEFRAG	HLP	File	9227	3-10-93	6:00 am	473	A	-	-	-	-	-	
Cluster 191, Sector 1,116													
DELTREE	EXE	File	11113	3-10-93	6:00 am	478	A	-	-	-	-	-	
DISKCOMP	COM	File	10620	3-10-93	6:00 am	484	A	-	-	-	-	-	

Sub-Directory

C:\DOS

Cluster 191

Offset 0, hex 0



Disk Editor

Object Edit Link View Info Tools Help

Cluster 191, Sector 1,115

00000000:	2E 20 20 20 20 20 20 20	-	20 20 20 10 00 00 00 00	.	.....
00000010:	00 00 00 00 00 00 DC 83	-	2A 20 BF 00 00 00 00 00	.....	â* .....
00000020:	2E 2E 20 20 20 20 20 20	-	20 20 20 10 00 00 00 00	..	.....
00000030:	00 00 00 00 00 00 DC 83	-	2A 20 00 00 00 00 00 00	.....	â* .....
00000040:	41 4E 53 49 20 20 20 20	-	53 59 53 20 00 00 00 00	ANSI	SYS ....
00000050:	00 00 00 00 00 00 00 30	-	6A 1A C0 00 69 23 00 00	.....	0j→.i#..
00000060:	41 50 50 45 4E 44 20 20	-	45 58 45 20 00 00 00 00	APPEND	EXE ....
00000070:	00 00 00 00 00 00 00 30	-	6A 1A C5 00 16 2A 00 00	.....	0j→...*..
00000080:	41 54 54 52 49 42 20 20	-	45 58 45 20 00 00 00 00	ATTRIB	EXE ....
00000090:	00 00 00 00 00 00 00 30	-	6A 1A CB 00 9D 2B 00 00	.....	0j→.¥+..
000000A0:	42 41 43 4B 55 50 20 20	-	45 58 45 20 00 00 00 00	BACKUP	EXE ....
000000B0:	00 00 00 00 00 00 00 28	-	89 16 D1 00 FC 8C 00 00	.....	(ë... "î..
000000C0:	43 48 4B 44 53 4B 20 20	-	45 58 45 20 00 00 00 00	CHKDSK	EXE ....
000000D0:	00 00 00 00 00 00 00 30	-	6A 1A E3 00 6B 32 00 00	.....	0j→π.kZ..
000000E0:	43 48 4B 4C 49 53 54 20	-	43 50 53 20 00 00 00 00	CHKLIST	CPS ....
000000F0:	00 00 00 00 00 00 5C 50	-	67 1E EA 00 99 09 00 00	.....	\Pg▲Ω.ő...
00000100:	43 48 4B 53 54 41 54 45	-	53 59 53 20 00 00 00 00	CHKSTATSYS	....
00000110:	00 00 00 00 00 00 00 30	-	6A 1A EC 00 80 A2 00 00	.....	0j→∞.Ćó..
00000120:	43 48 4F 49 43 45 20 20	-	43 4F 4D 20 00 00 00 00	CHOICE	COM ....
00000130:	00 00 00 00 00 00 00 30	-	6A 1A 01 01 DA 06 00 00	.....	0j→.....

Sub-Directory C:\DOS

Cluster 191 Offset 95, hex 5F

# Alocação de Espaço em Disco

- Tabela controla os clusters livres e ocupados
- Alocação Encadeada = FAT
- Alocação Indexada = NTFS, Unix

# FILE ALLOCATION TABLE

- Tabela com estado de todos os clusters
- Cada unidade de alocação possui uma entrada na FAT
- O menor espaço ocupado por um arquivo é um cluster
- Lista Encadeada

# TIPOS DE FAT

- Números guardados em cada entrada da FAT tem 12, 16 ou 32 bits
- 12 bit = 1 1/2 byte
- 16 bit = 2 bytes
- 32 bit = 4 bytes
- Discos Rígidos:
  - menos que 16 MB - 12-bit FAT
  - mais que 16 MB - 16-bit FAT

# FAT - 16 bit

Tamanho  
da Partição

Tamanho  
do Cluster

- 2 Kb
- 4 Kb
- 8 Kb
- 16 Kb
- 32 Kb

128 Mb

256 Mb

512 Mb

1 Gb

**2 Gb**

*máximo*

# FAT - 32 bit

Tamanho  
da Partição

Tamanho  
do Cluster

- 2 Kb
- 4 Kb
- 8 Kb
- 16 Kb
- 32 Kb

*máximo*

2 Gb  
8 Gb  
16 Gb  
32 Gb  
**2 Tb**

# Valores de Entradas na FAT

<i><b>Cluster</b></i>	<i><b>Valor</b></i>
<b>0</b>	<b>livre</b>
<b>1</b>	<b>ruim</b>
<b>2</b>	<b>EOF</b>

## Disk Editor

Object Edit Link View Info Tools Help

Sector 1

		3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	<EOF>	43	44	45	46	47	48
49	50	<BAD>	0	0	0	55	56
57	58	<BAD>	60	61	62	63	64
65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	<EOF>
81	82	83	84	85	86	87	88
89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128
129	130	131	132	133	<EOF>	135	136
137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152
153	154	155	156	157	158	159	160

FAT (1st Copy)

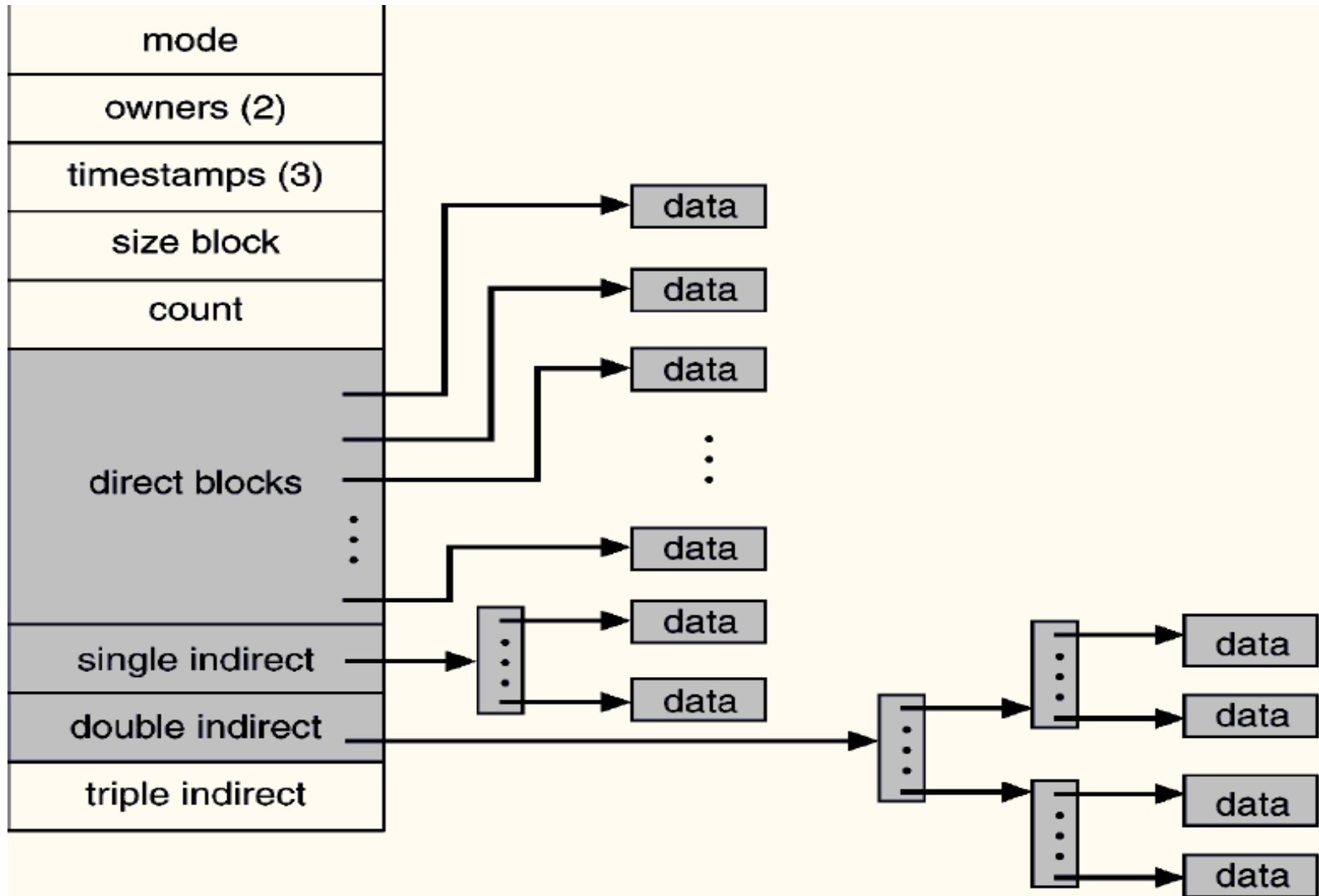
Sector 1

Drive A:

Cluster 2, hex 2



# I-node Linux



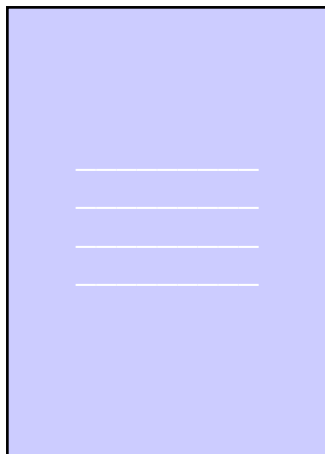


# Exercício – Alocação de Espaço DOS

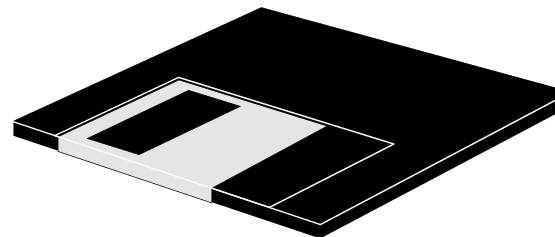
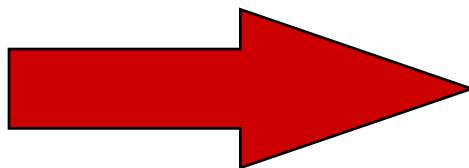
Exemplo de alocação de  
espaço em disco para um  
arquivo, sob o sistema  
DOS

# Criando Arquivo - DOS

*Arquivo.txt*



**Tamanho:**  
*2.300 bytes*



## Disquete de 3 1/2

512 Bytes por Setores

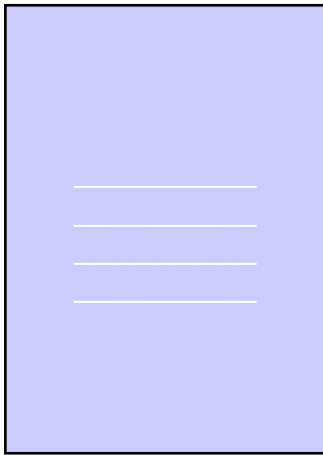
2 Setores por Cluster

1.024 Bytes por Cluster

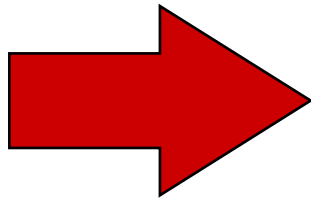
3 Clusters para *Arquivo.txt*

# Nova Entrada em Diretório - DOS

*Arquivo.txt*



**Tamanho:**  
*2.300 bytes*

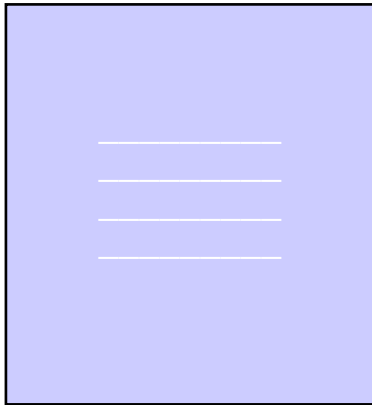


## Entrada de Diretório

- ✓ nome do arquivo
- ✓ hora de criação
- ✓ data de criação
- ✓ cluster inicial
- ✓ tamanho em bytes

# Alocando Espaço - DOS

*Arquivo.txt*



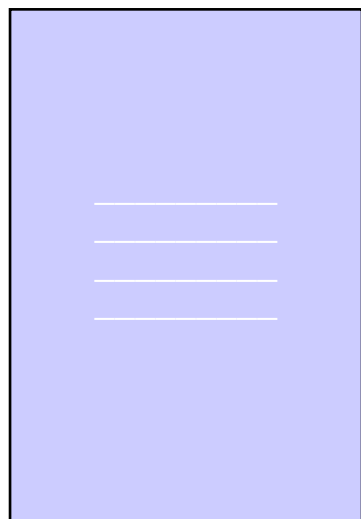
**Tamanho:**  
*2.300 bytes*

**FAT**

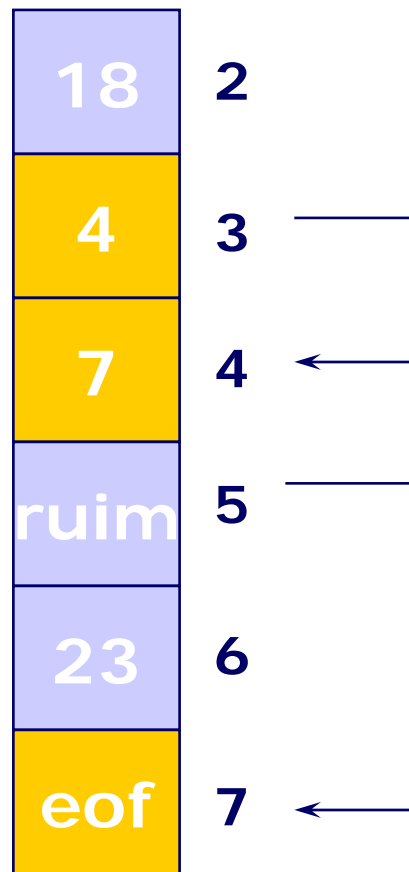
18	2
livre	3
livre	4
ruim	5
23	6
livre	7

# Ocupando Espaço - DOS

*Arquivo.txt*



Começando em 3





# Sistema de Arquivos NTFS

- A estrutura fundamental do sistema de arquivos NTFS é um volume:
  - O utilitário de administração de discos é usado para criá-los
  - É baseado em uma partição lógica de disco
  - Pode ocupar uma porção, um disco inteiro ou estender-se por múltiplos discos.
- Todos os meta dados, como informações sobre o volume, são guardados em um arquivo regular.
- NTFS usa clusters como unidade básica de alocação de espaço em volumes:
  - Um cluster =  $2n$  setores



## NTFS - Detalhes

- NTFS endereça espaço em disco por meio de números de clusters lógicos (*logical cluster numbers* - LCNs).
- Um arquivo do NTFS, além de ser uma seqüência de bytes como no MS-DOS ou UNIX, é um objeto estruturado consistido de atributos.
- Arquivos são descritos por registros, compondo um vetor armazenado no arquivo especial *Master File Table* (MFT).
- Cada arquivo em um volume possui um único identificador chamado *file reference*.
  - 48-bit *file number* + 16-bit *sequence number*.
- O espaço de nomes do NTFS é organizado em uma hierarquia de diretórios; a raiz de indexação (*index root*) contém o nível superior de uma árvore B+.



# NTFS - Segurança

- A segurança em volumes NTFS é derivada do modelo de objetos do W2K.
- Cada objeto arquivo tem seu descritor de segurança armazenado em um registro na MFT.
- Este descritor contém:
  - *Token* de acesso do criador do arquivo
  - A C L – *Access Control List* – Lista de Controle de Acesso

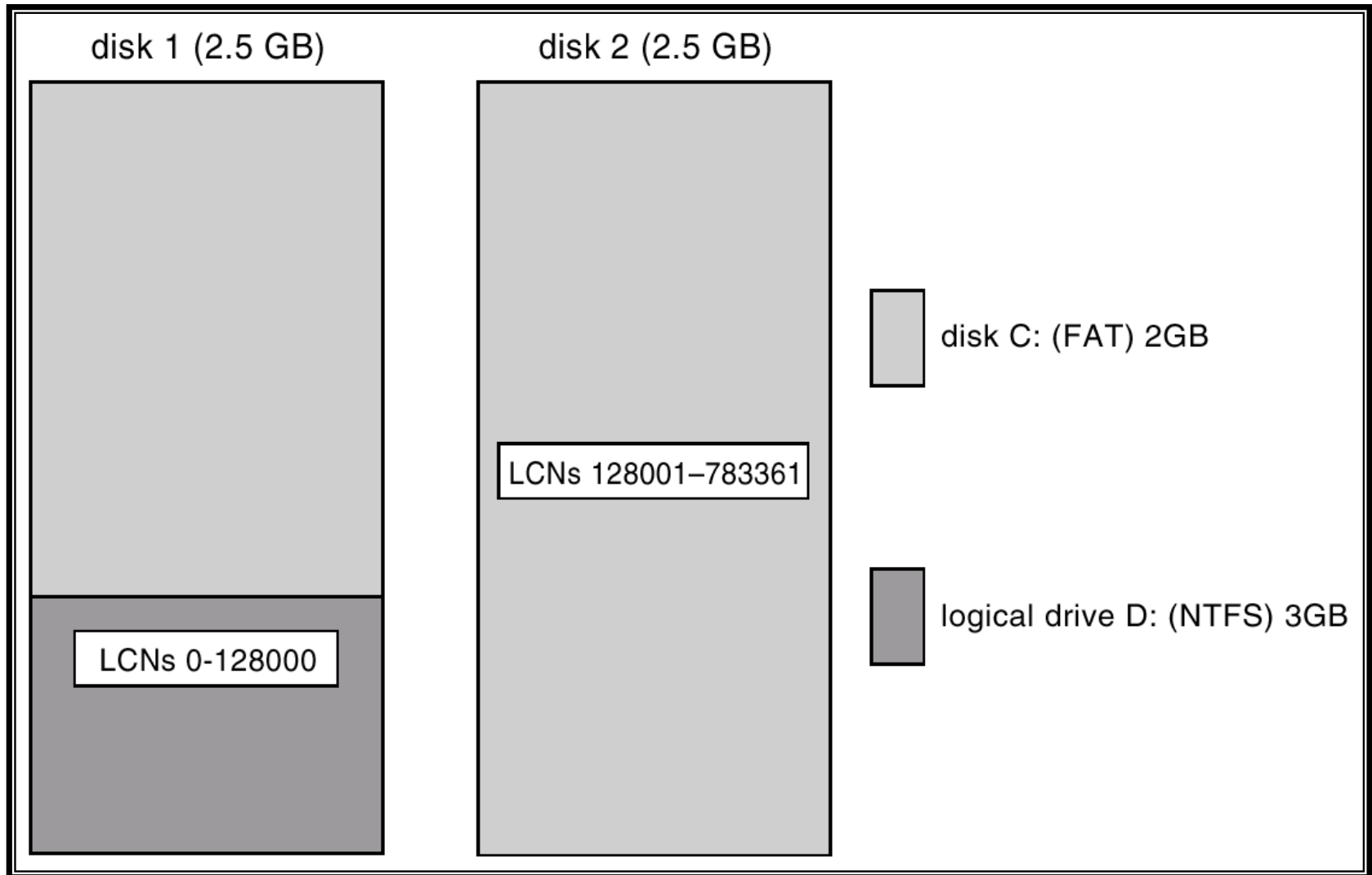


# Gerência de Volumes e Tolerância a Falhas

- *FtDisk* - *fault tolerant disk driver*, pode combinar múltiplos discos em um volume lógico
- Múltiplos discos podem ser concatenados para formar um *volume set*
- *Stripe set* = RAID 0 – múltiplas partições conjugadas
- Espelhamento de discos (*mirroring*), ou RAID 1 – duas partições iguais em dois discos
- Re-mapeamento de clusters para corrigir erros



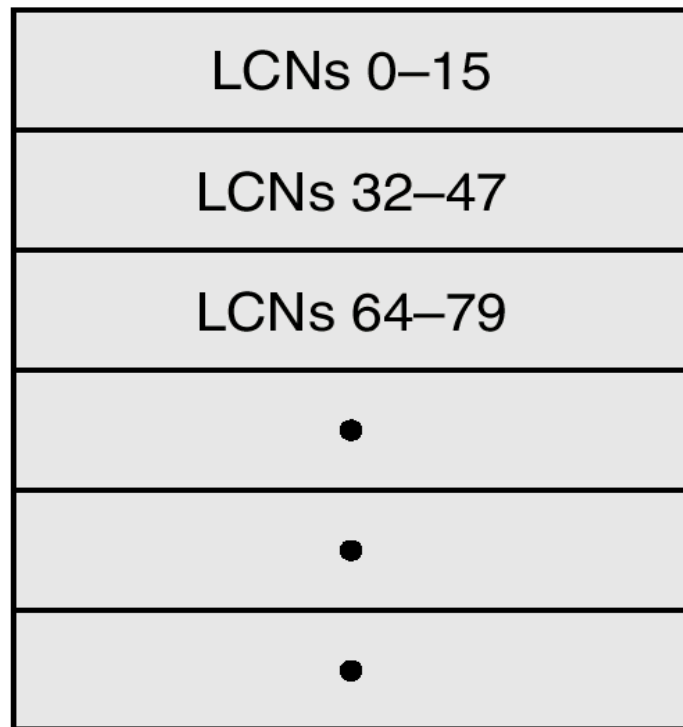
# Volumes em Dois Discos



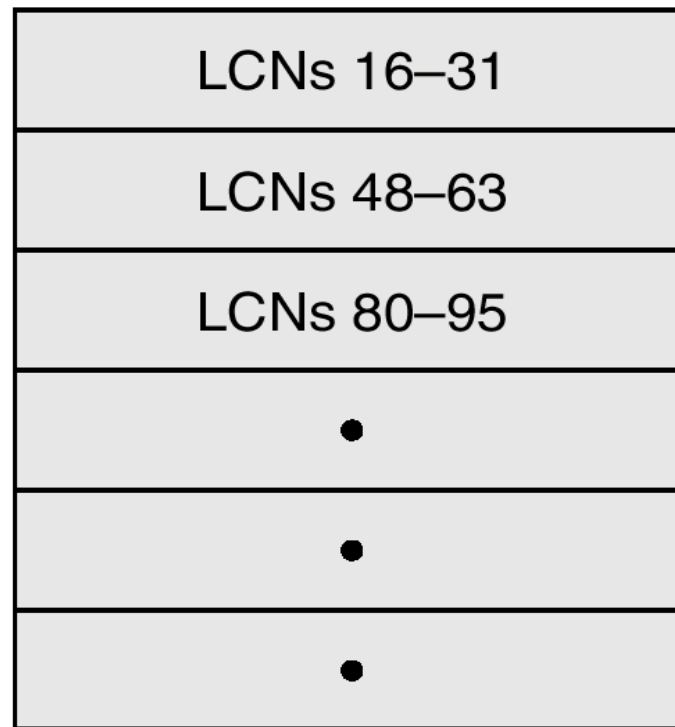


# Stripe Set em Dois Discos

disk 1 (2 GB)



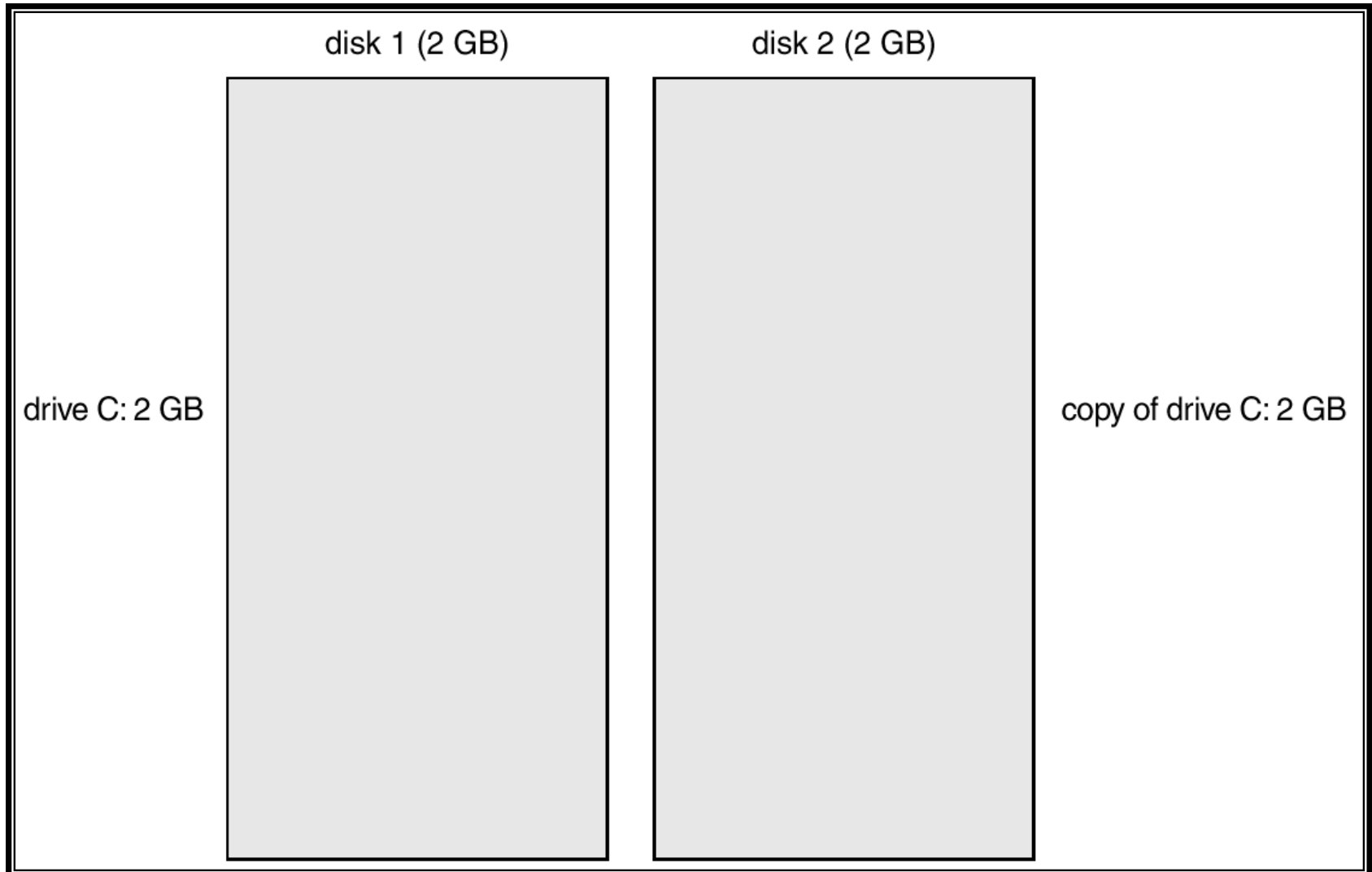
disk 2 (2 GB)



logical drive C: 4 GB



# Espelhamento em Dois Discos



## ***MFT - Master File Table***

- Entradas descrevem *todos* os arquivos e diretórios
- Entrada (#0) descreve a própria MFT , com dados como seu tamanho
- Diretório raiz (#5)
- Mapa de alocação de clusters
- *Security Descriptors*
- *Journaling*

## Entradas da MFT (1)

- cada entrada tem seu número (similares aos *inodes* do linux)
- arquivos e diretórios de usuários começam com #25
- as entradas contêm listas de atributos
- Exemplos:
  - *standard information*, contendo:
    - datas e horas (*MAC times*)
    - nome do arquivo
    - dados (\$Data)
    - "*index alloc*" e "*index root*", para diretórios

## Entradas da MFT (2)

- cada atributo tem um "id", o que significa que mais de um de cada tipo pode existir para cada arquivo
- atributos \$Data com mais de um "id" podem ser usados para esconder dados (*alternate streams*)
- atributos residentes tem cabeçalho e conteúdo guardados na \$MFT (se couberem)
- caso contrário, conteúdo é guardado em clusters
- excesso de atributos são distribuídos em mais de uma entrada na \$MFT



# NTFS – Arquivos (1)

- Atributos Típicos
- \$STANDARD\_INFORMATION (#16)
  - MAC
  - *security ID*
  - ID dos proprietários
  - permissões DOS
  - dados para controle de quotas
- \$FILE\_NAME (#48)
  - nome em unicode
  - MAC adicionais
  - entrada da \$MFT do diretório "pai"

## NTFS – Arquivos (2)

- \$OBJECT\_ID (#64)
  - identificadores originais sobre "Object ID", "Volume ID", "Domain ID"
- \$DATA (#128)
- Quando o arquivo é excluído, a flag "IN\_USE" da entrada na \$MFT é limpa, mas outros atributos permanecem inalterados



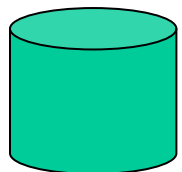
# Informações Armazenadas em Meios Eletrônicos



# **Material Adicional**

**Informações Armazenadas  
em  
Meios Eletrônicos**

# Exercício – FAT - DOS



Disco c/  
10.000 MBytes

Tabela c/  
100 células

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										

Cada célula corresponderá a \_\_\_\_\_ Bytes

# Exercício – Novos Arquivos - DOS

0 1 2 3 4 5 6 7 8 9

Nome	1o Cluster	Tam
a1	[ ]	200 MB
a2	[ ]	300 MB
a3	[ ]	474 MB
a4	[ ]	400 KB
a5	[ ]	138 MB

0									
1									
2									
3									
4									
5									
6									
7									
8									

*5 arquivos novos*



# Exercício – Inserindo Arquivo - DOS

0 1 2 3 4 5 6 7 8 9

Nome	1o Cluster	Tam
a1	[ ]	200 MB
a2	[ ]	300 MB
a3	[ ]	474 MB
	[ ]	400 KB
a5	[ ]	138 MB
	[ ]	

0									
1									
2									
3									
4									
5									
6									
7									
8									

Inserir **a6** com **589 MB**

iniciando no 1o. Cluster de **a4** em nova entrada de diretório



# NTFS – Diretórios (1)

- Diretórios são indexados no armazenamento
- B-Tree ordenadas em ordem alfabética
- Atributos típicos
  - \$STANDARD\_INFORMATION (#16)
  - \$FILE\_NAME (#48)
  - \$OBJECT\_ID (#64)
  - \$INDEX\_ROOT (#144)
    - raiz da B-Tree
    - Estruturas \$INDEX\_ROOT e uma ou mais "Index Entry" descrevem arquivos ou sub-diretórios
    - a estrutura "Index Entry" contém cópia de atributos "\$FILE\_NAME" para arquivos ou sub-diretórios
  - \$INDEX\_ALLOCATION (#160)

## NTFS – Diretórios (2)

- sub-nós da B-Tree - pequenos diretórios tem seus dados guardados na própria estrutura do \$INDEX\_ROOT
- O conteúdo dos \$INDEX\_ALLOCATION é composto de "Index Buffers"
- Cada "Index Buffer" contém uma ou mais estruturas "Index Entry", iguais às encontradas no \$INDEX\_ROOT
- \$BITMAP (#176) - aponta quais estruturas da B-Tree estão em uso