Sistemas Operacionais

Sistema de Arquivos

Edeyson Andrade Gomes

www.edeyson.com.br

Sistema de Arquivos

- Mecanismo que provê armazenamento e acesso a dados e programas do Sistema Operacional e do usuário;
- Aspecto mais visível do Sistema Operacional;
- Formado por duas partes distintas:
  - Coleção de Arquivos: Armazenagem de dados;
  - Estrutura de Diretórios: Responsável pela organização e informações sobre os arquivos do sistema;
- Interface homogênea e transparente para a manipulação de dados em memória secundária.

Sistema de Arquivos

- Arquivo:
  - Coleção de informações logicamente relacionadas (*bits*, *bytes*, linhas e registros), que representam programas ou dados;
  - Referido pelos usuários através de um nome;
  - Conjunto de registros definidos pelo sistema de arquivos;
  - É composto por uma série de atributos, que podem variar de acordo com o Sistema Operacional. Atributos básicos são:

Atributos dos Arquivos

- Nome: Distinção entre caracteres, extensão máxima, partes (nome e extensão);
- Localização: Formado pelo ponteiro para o dispositivo e o local onde se encontra o arquivo neste dispositivo;
- Tamanho: Tamanho atual do arquivo. Pode conter também o tamanho máximo permitido;
- Informações para proteção de acesso: privilégios de acesso dos usuários;
- Data e hora: da criação do arquivo, da última modificação feita no arquivo e do último acesso feito ao arquivo;
- Identificação do usuário: que criou o arquivo.

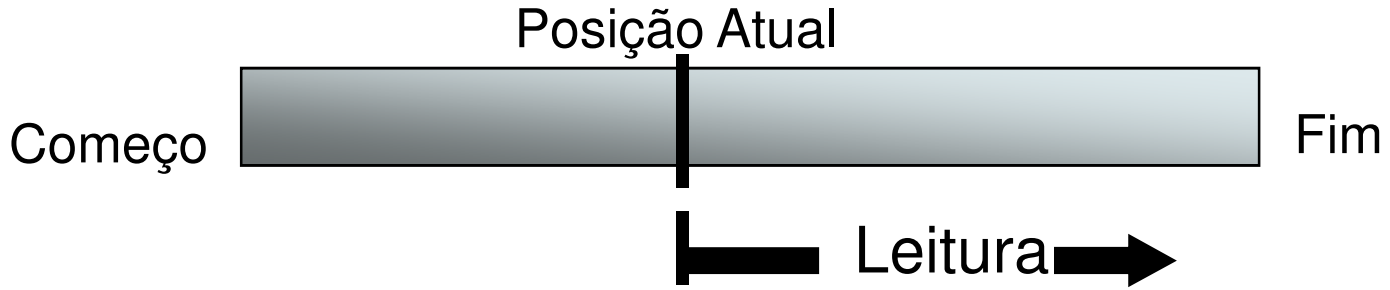
Métodos de Acesso ao Arquivo

Métodos de Acesso ao Arquivo

- Forma como a informação é acessada no arquivo;
- A depender de como o arquivo está organizado o sistema pode recuperá-lo de diferentes maneiras;
- Tipos de acesso:
  - Acesso seqüencial;
  - Acesso direto;
    - Acesso indexado;
- Sistemas Operacionais podem dar suporte a mais de um método de acesso.

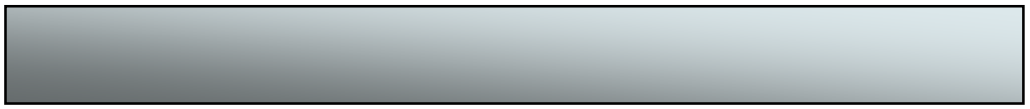
Métodos de Acesso ao Arquivo

- Acesso seqüencial:
  - Método de acesso mais comum. Usado em fitas;
  - Informações no arquivo são processadas na ordem que foram gravadas;
  - Gravação só é possível no final do arquivo.



Métodos de Acesso ao Arquivo

- Acesso seqüencial:
  - Alocação Contígua



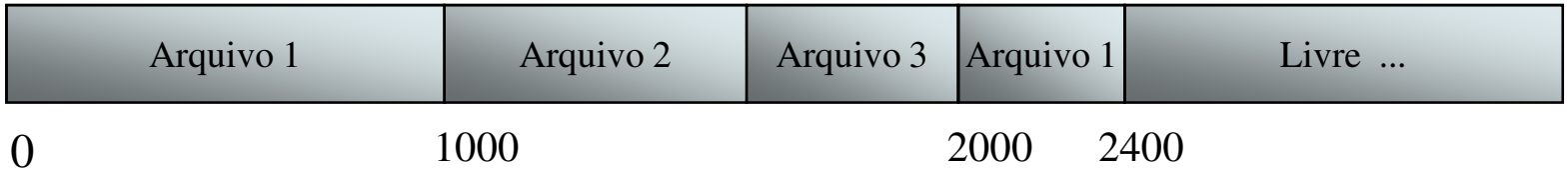
  - Informações do Arquivo:
    - Início e Tamanho
  - Problemas
    - Sistema trabalha com pré-alocação
      - Usuário precisa definir antecipadamente o tamanho do arquivo

Métodos de Acesso ao Arquivo

- Acesso seqüencial:
  - Problemas
    - Alocação de arquivo maior que necessário
      - Desperdício
    - Alocação de arquivo menor que necessário

Métodos de Acesso ao Arquivo

- Alocação Contígua
  - Como o arquivo1 pode crescer?

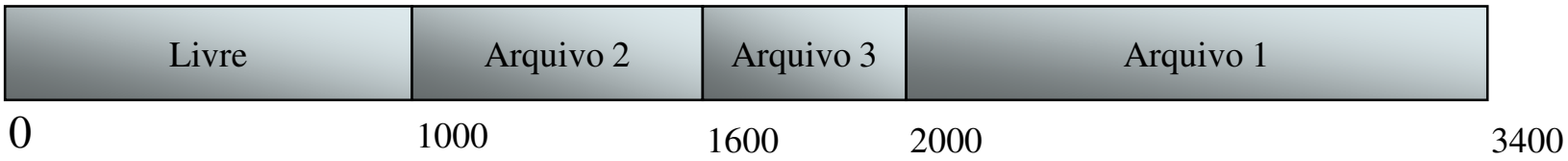


  - Informação do Arquivo
    - Início, Tamanho, número de segmentos
      - Segmento 0 ➔ 0, 1000
      - Segmento 1 ➔ 2000, 400

Métodos de Acesso ao Arquivo

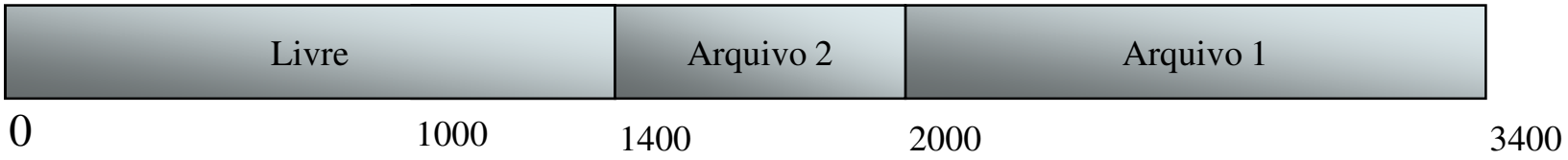
- Alocação Contígua
  - Muitos segmentos
    - Perde desempenho no acesso seqüencial
      - Desfragmentação
      - Arquivo com um único segmento

Métodos de Acesso ao Arquivo



Apagando o Arquivo 3

Como alocar um arquivo com 1200 bytes?



Métodos de Acesso ao Arquivo

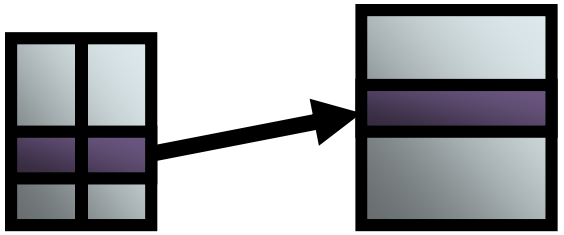
- Alocação Contígua
  - Perguntas:
    - Se apagarmos o Arquivo 3, como reusar o espaço?
    - Como determinar as áreas livres do disco?
    - Como alocar tais áreas?
  - Respostas:
    - Listas encadeadas ou mapas de bits
    - Best Fit, First Fit ou Worst Fit
      - Parece com o que?

Métodos de Acesso ao Arquivo

- Acesso direto:
  - Arquivo é dividido em blocos que podem ser acessados em qualquer ordem;
  - Ideal para grande quantidade de informação, como as bases de dados;
  - Leitura baseada na especificação do número do bloco;
  - Possível combinar o acesso direto com o acesso seqüencial.

Métodos de Acesso ao Arquivo

- Acesso indexado:
  - Sofisticação do acesso direto;
  - Chamado de acesso indexado ou por chave;
  - Arquivo deve possuir uma área de índice onde existam ponteiros para os diversos registros;



Operações sobre Arquivos

- System Calls permitem às aplicações realizar operações de E/S como tradução de nomes em endereços, leitura e gravação de dados, criação e eliminação de arquivos;
- Operações sobre arquivos mais comuns são:

  - Criar;
  - Abrir;
  - Ler;
  - Gravar;
  - Fechar;
  - Renomear;
  - Posicionar;
  - Destruir.

Sistema de Arquivos

- Discos podem ser divididos em diversas partições ou volumes;
- Cada disco contém pelo menos uma partição onde estão localizados os arquivos e os diretórios;
- Diretório:
  - Também chamado de diretório de dispositivo ou Tabela de conteúdo do volume;
  - Estrutura de dados que mantém informações sobre a coleção dos arquivos contidos no disco;
    - Nome, tamanho, localização, tipo, etc;
  - Pode ser visto como um conjunto de tabelas que associam nomes a arquivos.

Sistema de Arquivos

- Um diretório também é um arquivo, com vários atributos, mas tratado de forma diferenciada pelo SO (Bit define se é arquivo – bit = 0 ou diretório – bit = 1)
- Operações sobre diretórios mais comuns são:
  - Procurar, apagar, renomear e copiar arquivos;
  - Mostrar uma lista com o conteúdo do diretório;
- Estruturas mais comuns são:
  - Nível Único (Single-Level Directory);
  - Dois Níveis (Two-Level Directory);
  - Árvore;
  - Grafo Acíclico;
  - Grafo.

Implementação do Sistema de Arquivos

- Forma de armazenamento de arquivos na memória secundária demanda controle dos espaços livres e dos espaços alocados aos arquivos e diretórios;
- Espaço livre gerenciado através de:
  - Mapa de bits: Cada entrada na tabela é associada a um bit que indica se o bloco está livre (bit=0) ou ocupado (bit=1);
  - Ligação encadeada: Cada bloco livre guarda o endereço do próximo bloco livre;
  - Tabela com endereço do 1° bloco livre de cada segmento e o número de blocos livres contíguos;

Alocação de Disco

- Alocação Contígua:
  - Armazenagem do arquivo no disco em blocos seqüenciais;
  - Informações necessárias: Nome, endereço de início e tamanho;
  - Problemas:
    - Definição do tamanho definitivo do arquivo no instante da sua criação;
    - Pré-alocação de espaço "extra" pode gerar ociosidade do espaço alocado por muito tempo;
    - Alocação para novos arquivos depende que existam *n* blocos dispostos em seqüência no disco.

Alocação de Disco

- Alocação Contígua:
  - Três estratégias para alocação do espaço para o arquivo:
    - First-fit: Primeiro segmento livre com tamanho suficiente para o arquivo;
    - Best-fit: Menor segmento livre disponível com tamanho suficiente para o arquivo;
    - Worst-fit: Maior segmento livre disponível com tamanho suficiente para o arquivo;
  - Todas as três estratégias geram fragmentação de espaços livres.

Alocação de Disco

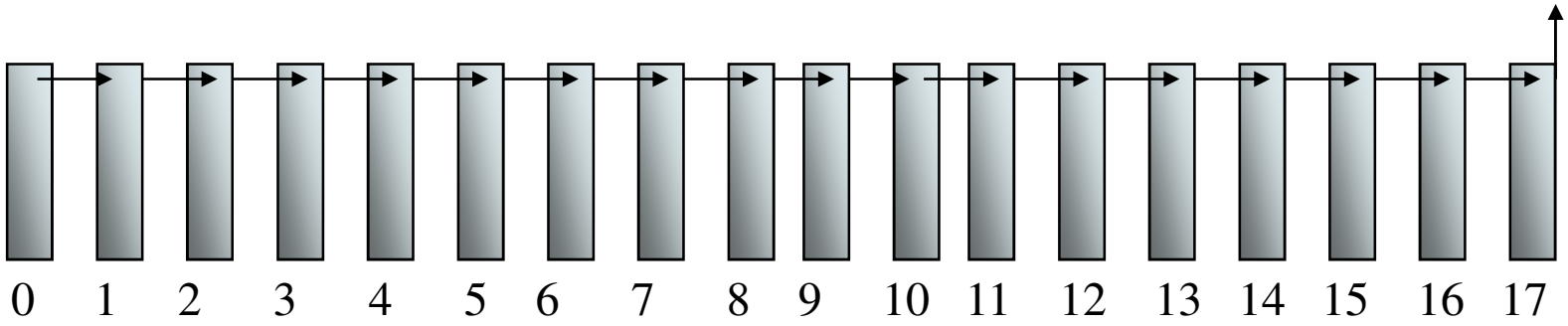
- Situação crítica: Quando existem espaços livres mas nenhum deles suporta um novo arquivo;
- Solução é a desfragmentação do disco, que deve ser periódica, pois seu efeito é temporário.

Alocação de Disco

- Uso de Blocos
  - Menor unidade de alocação lógica
  - Diminui a fragmentação a um único bloco por arquivo
    - Apenas o último bloco
  - Perda média = 50% do tamanho do bloco por arquivo

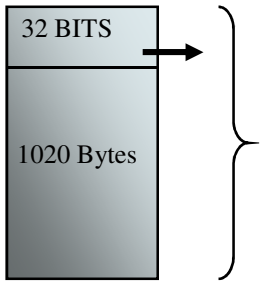
Alocação de Disco

Ligação encadeada



LBL – Lista de Blocos Livres

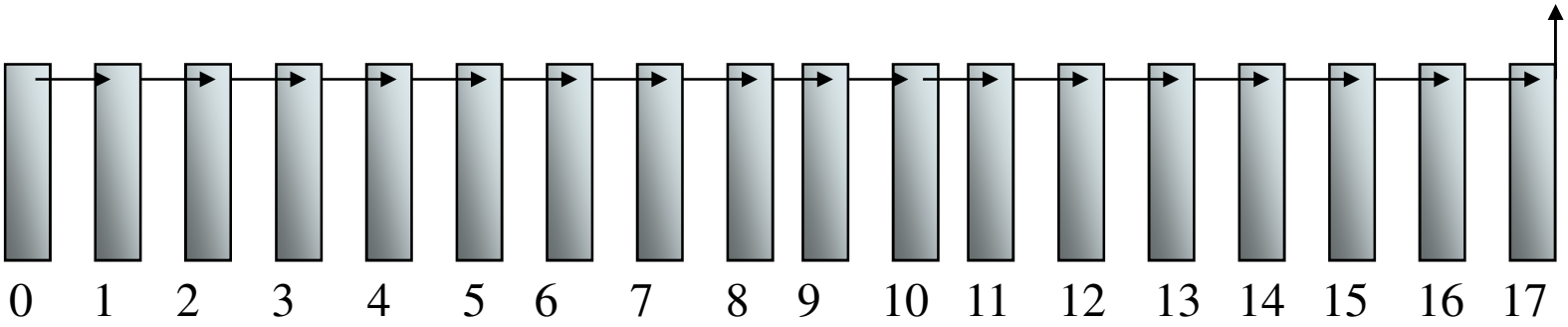
LBL → Bloco 0, 18KB



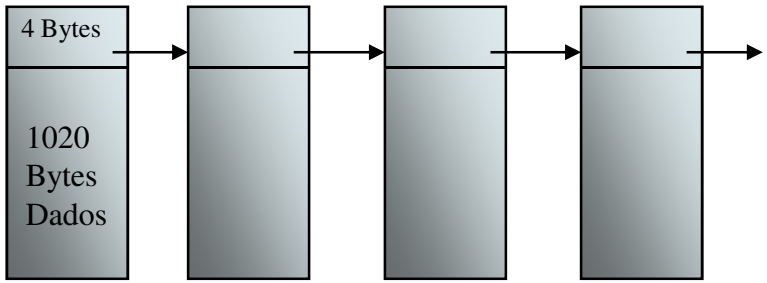
Bloco com 1KB = 1024 BYTES = 4 + 1020

Alocação de Disco

Ligação encadeada



LBL – Lista de Blocos Livres
LBL → Bloco 0, 18KB

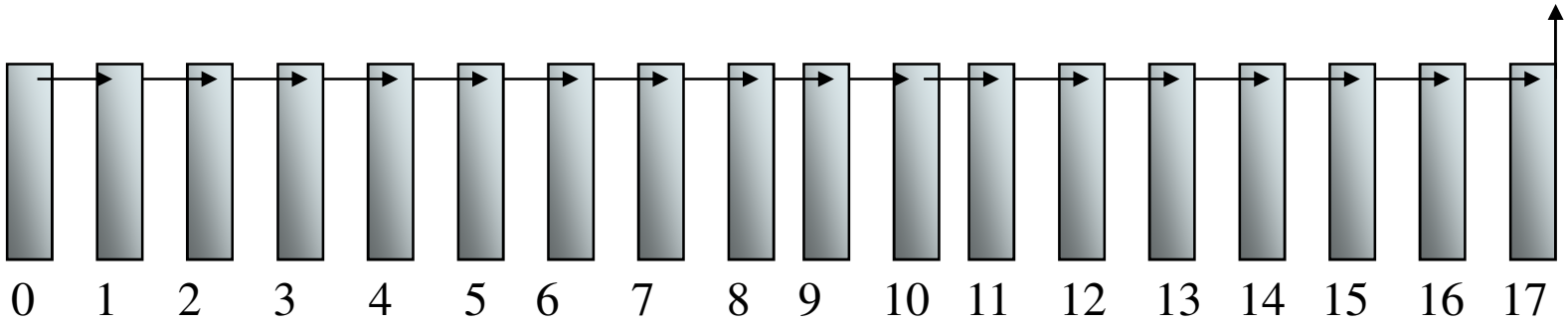


Para alocar um arquivo
com 3 KB – 3072 bytes – necessitamos
de 4 blocos devido ao espaço dos ponteiros.

Alocação de Disco

Ligação encadeada

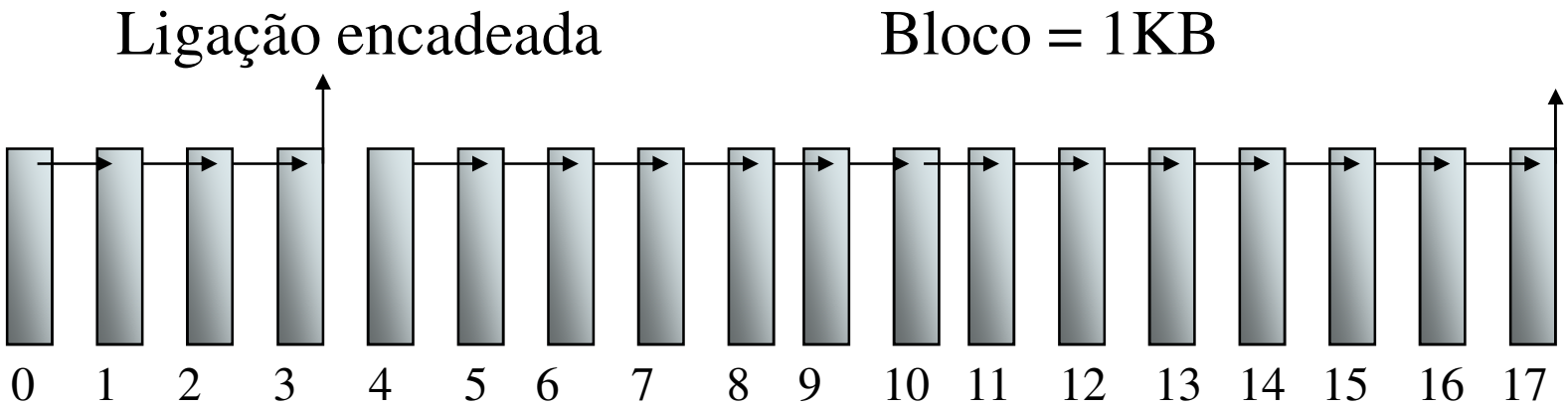
Bloco = 1KB



NOME	Bloco Inicial	Tamanho
LBL	0	18K

Criar Arquivo 1 com 3072 Bytes

Alocação de Disco



NOME	Bloco Inicial	Tamanho
LBL	4	14K
Arquivo 1	0	3072

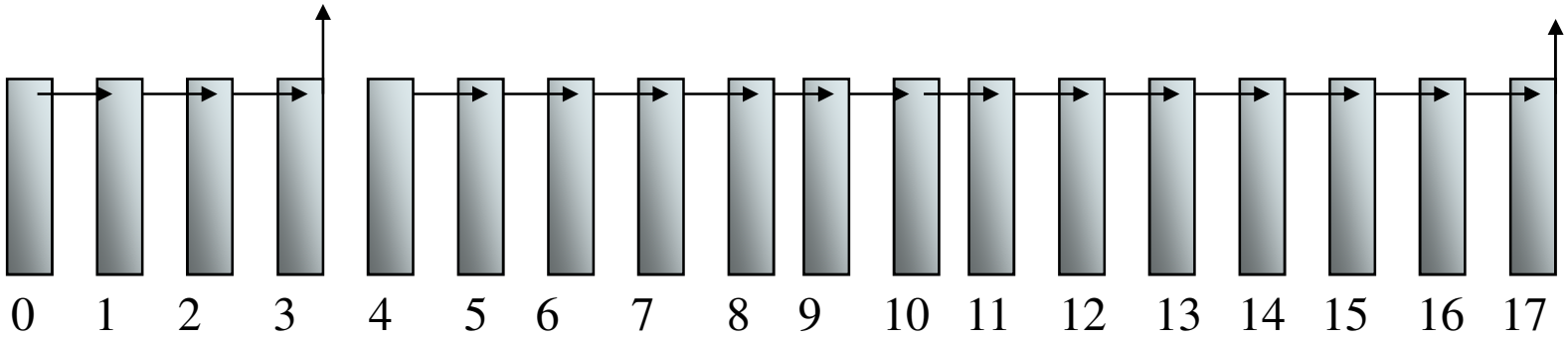
Criar Arquivo 1 com 3072 Bytes

3072 / 1020 = 3 com resto 12

3 blocos cheios e 1 com 12 bytes.

Alocação de Disco

Ligação encadeada

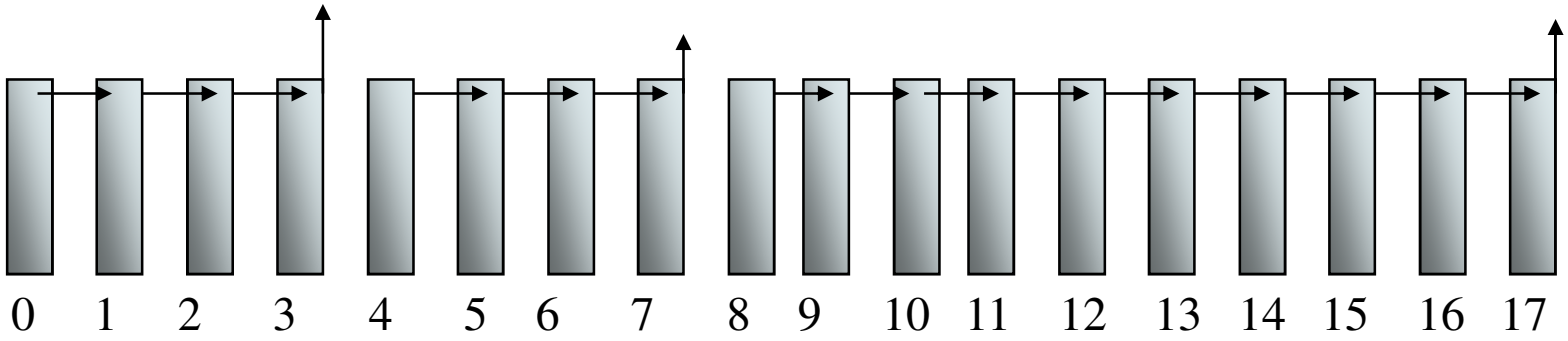


NOME	Bloco Inicial	Tamanho
LBL	4	14K
Arquivo 1	0	3072

Criar Arquivo 2 com 4000 Bytes

Alocação de Disco

Ligação encadeada

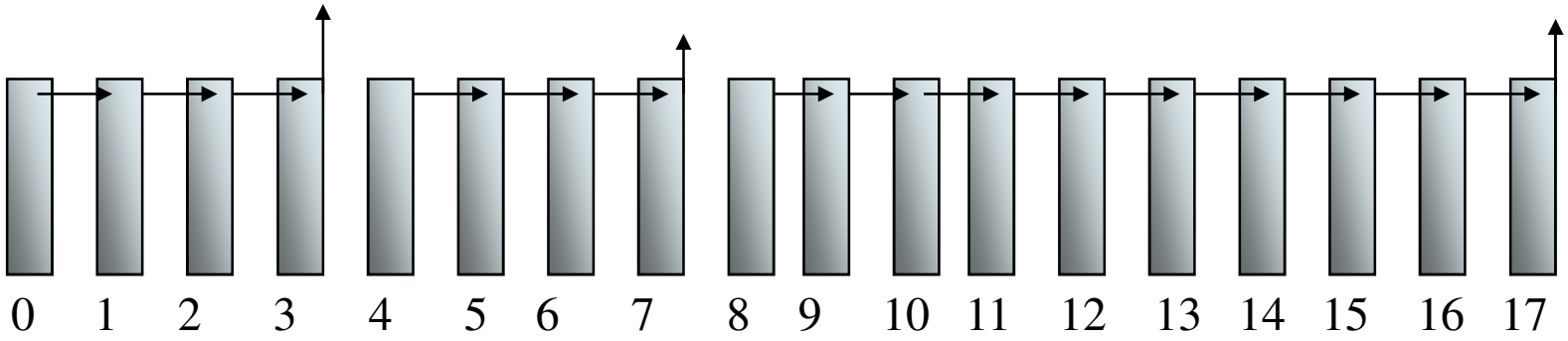


NOME	Bloco Inicial	Tamanho
LBL	8	10K
Arquivo 1	0	3072
Arquivo 2	4	4000

Criar Arquivo 2 com 4000 Bytes

Alocação de Disco

Ligação encadeada

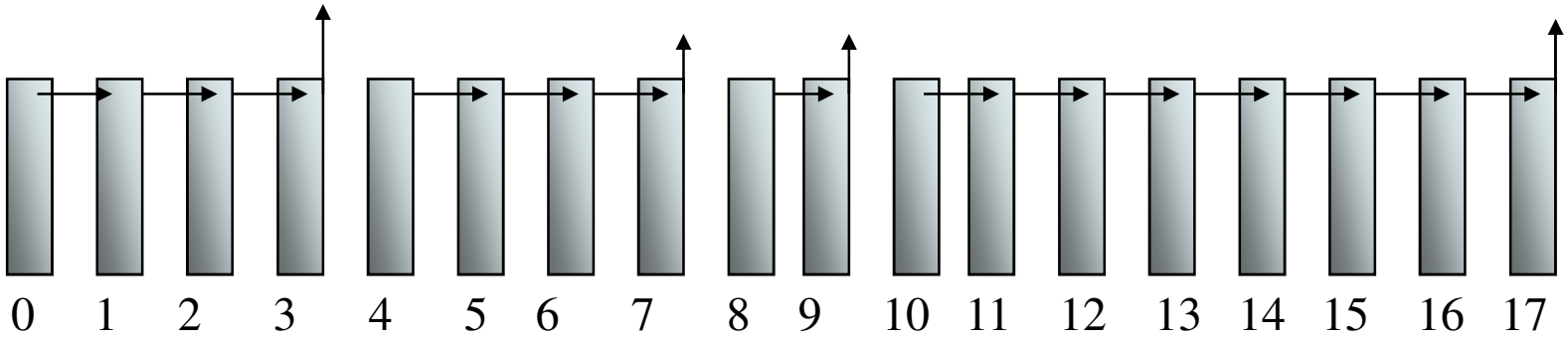


NOME	Bloco Inicial	Tamanho
LBL	8	10K
Arquivo 1	0	3072
Arquivo 2	4	4000

Criar Arquivo 3 com 2000 Bytes

Alocação de Disco

Ligação encadeada

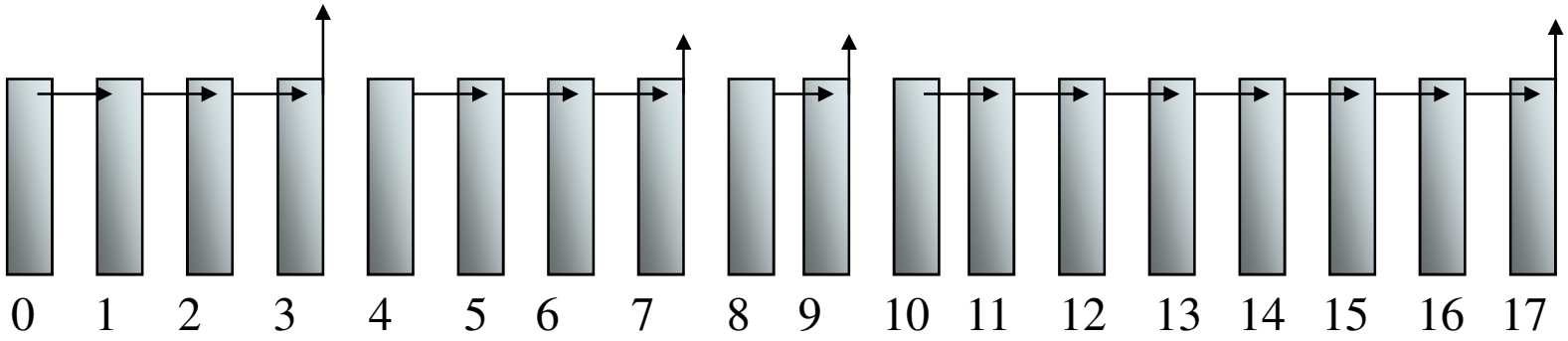


NOME	Bloco Inicial	Tamanho
LBL	10	8K
Arquivo 1	0	3072
Arquivo 2	4	4000
Arquivo 3	8	2000

Criar Arquivo 3 com 2000 Bytes

Alocação de Disco

Ligação encadeada

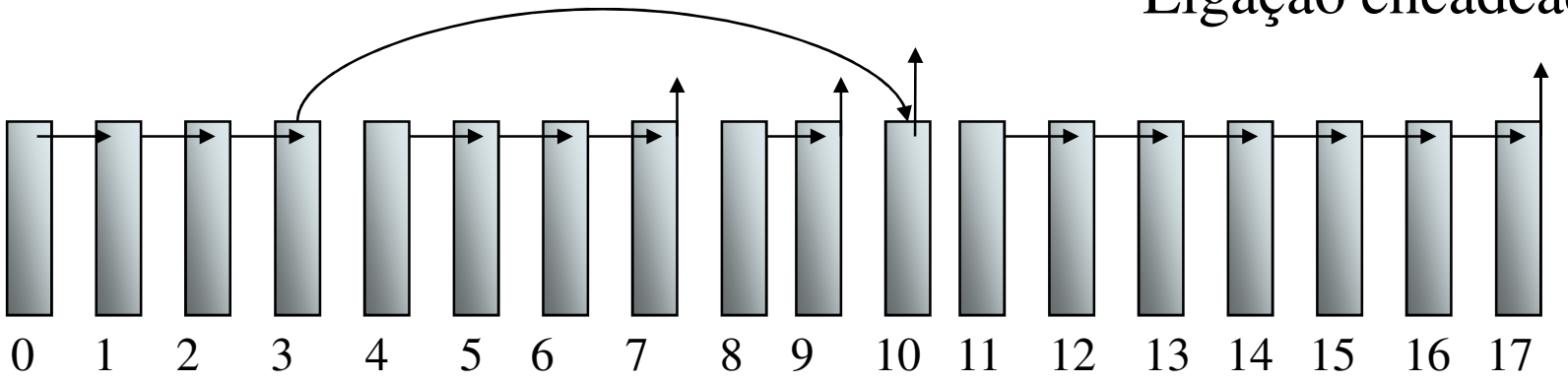


NOME	Bloco Inicial	Tamanho
LBL	10	8K
Arquivo 1	0	3072
Arquivo 2	4	4000
Arquivo 3	8	2000

Arquivo 1 cresce de 3072 bytes para 4200 bytes

Alocação de Disco

Ligação encadeada



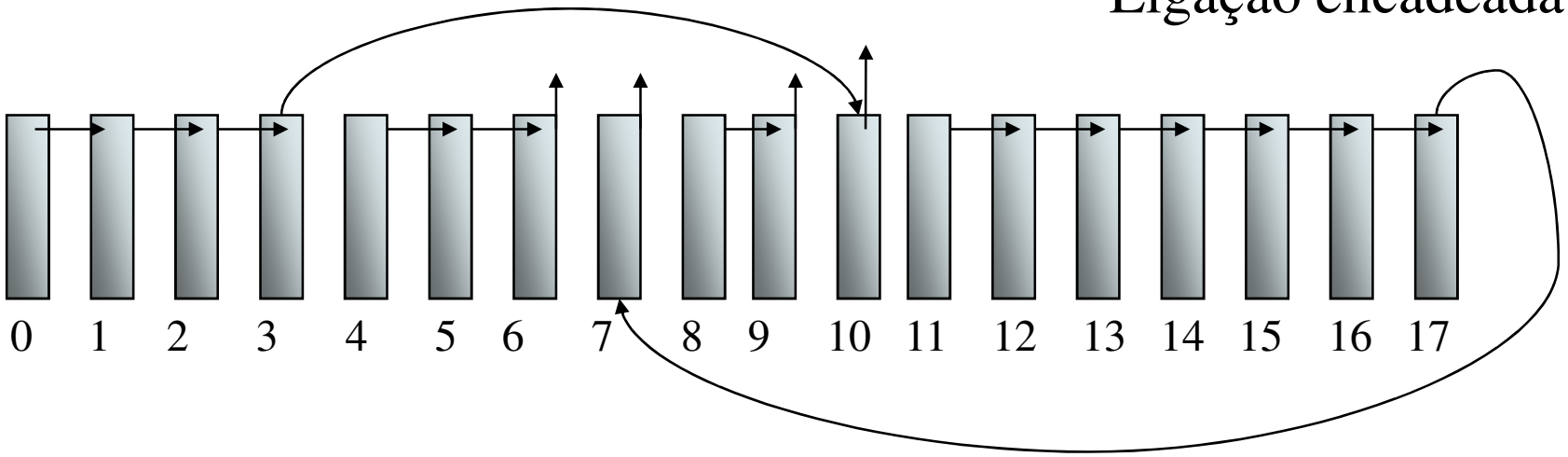
NOME	Bloco Inicial	Tamanho
LBL	11	7K
Arquivo 1	0	4200
Arquivo 2	4	4000
Arquivo 3	8	2000

Arquivo 1 cresce de 3072 bytes para 4200 bytes

Arquivo 2 diminui de 4000 bytes para 3000 bytes

Alocação de Disco

Ligação encadeada

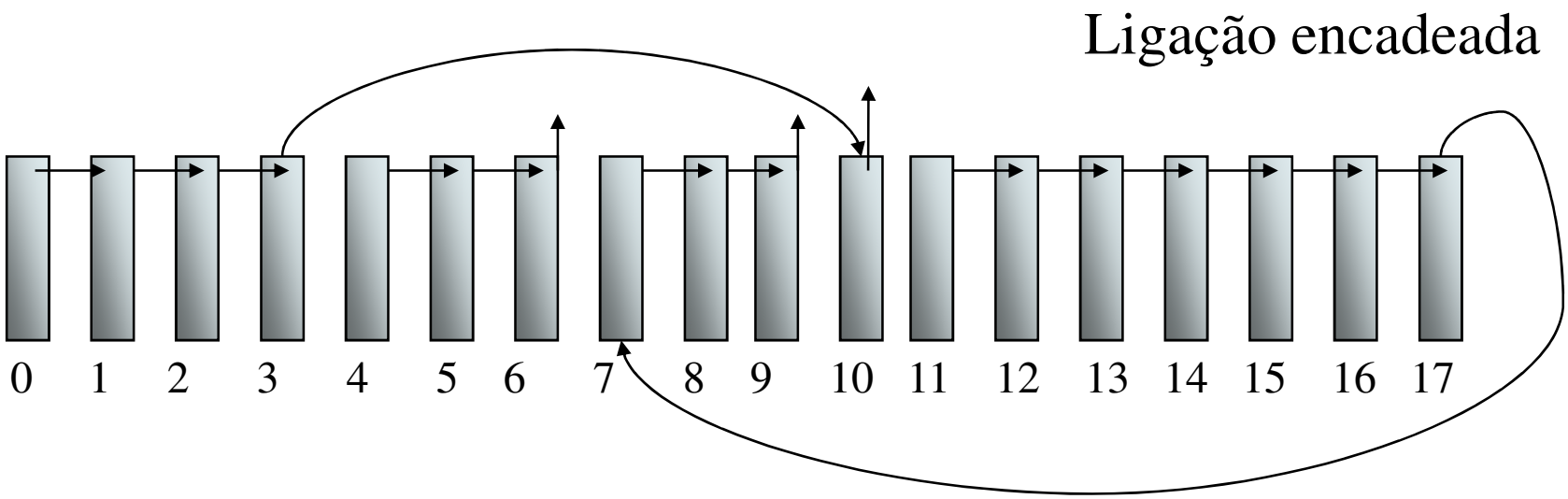


NOME	Bloco Inicial	Tamanho
LBL	11	8K
Arquivo 1	0	4200
Arquivo 2	4	3000
Arquivo 3	8	2000

Arquivo 2 diminui de 4000 bytes para 3000 bytes

Apagando o Arquivo 3

Alocação de Disco

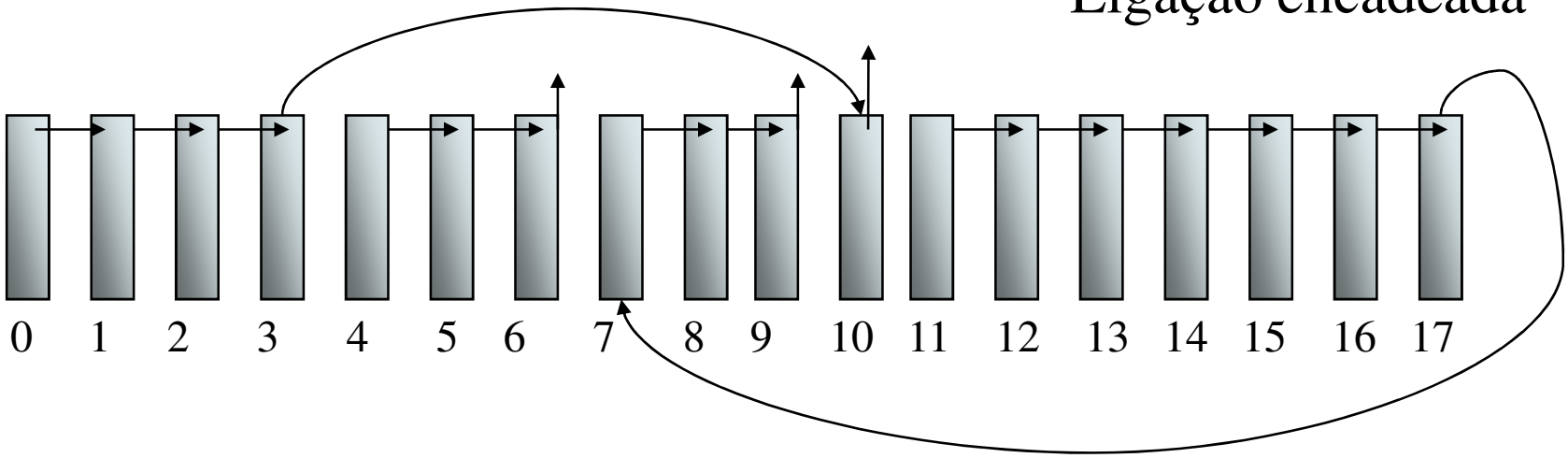


NOME	Bloco Inicial	Tamanho
LBL	11	10K
Arquivo 1	0	4200
Arquivo 2	4	3000
?rquivo 3	8	2000

Apagando o Arquivo 3

Alocação de Disco

Ligação encadeada



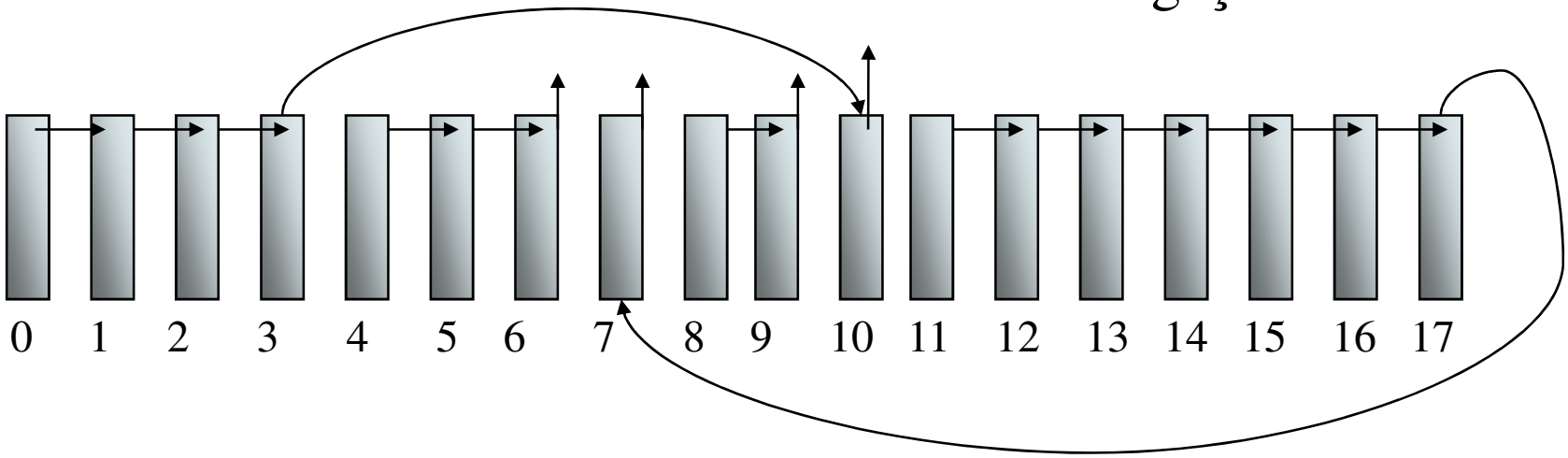
NOME	Bloco Inicial	Tamanho
LBL	11	10K
Arquivo 1	0	4200
Arquivo 2	4	3000
?Arquivo 3	8	2000

Restaurando o Arquivo 3

Como é possível?
O ? Indica exclusão lógica.

Alocação de Disco

Ligação encadeada



NOME	Bloco Inicial	Tamanho
LBL	11	8K
Arquivo 1	0	4200
Arquivo 2	4	3000
Arquivo 3	8	2000

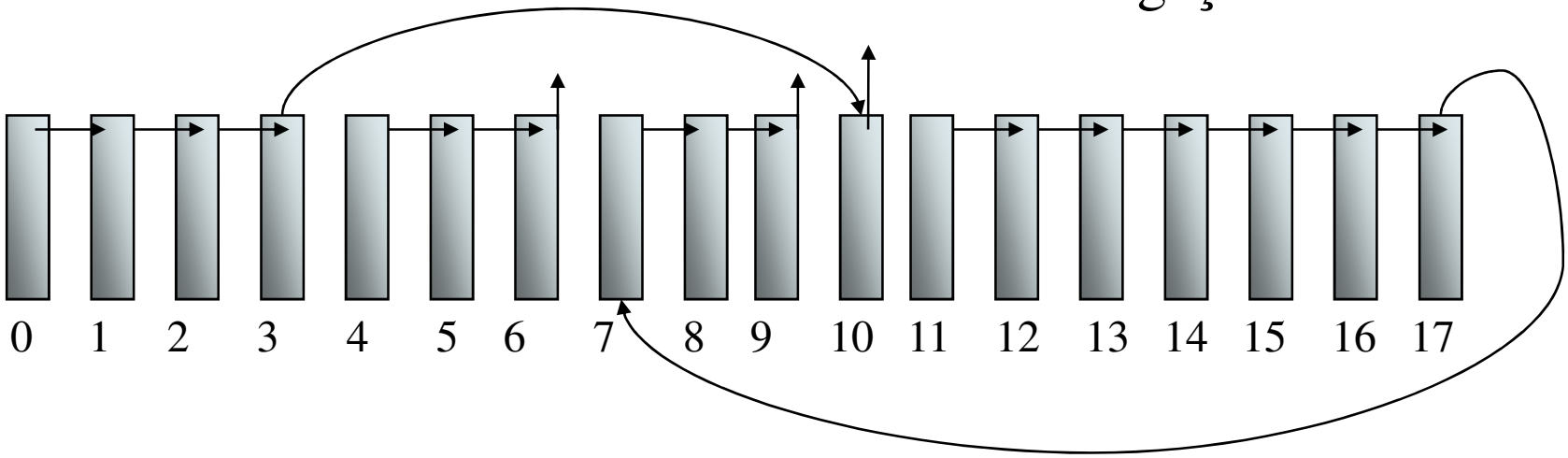
Restaurando o Arquivo 3

Como é possível?
O ? Indica exclusão lógica.

Apagando os Arquivos 3 e 1.

Alocação de Disco

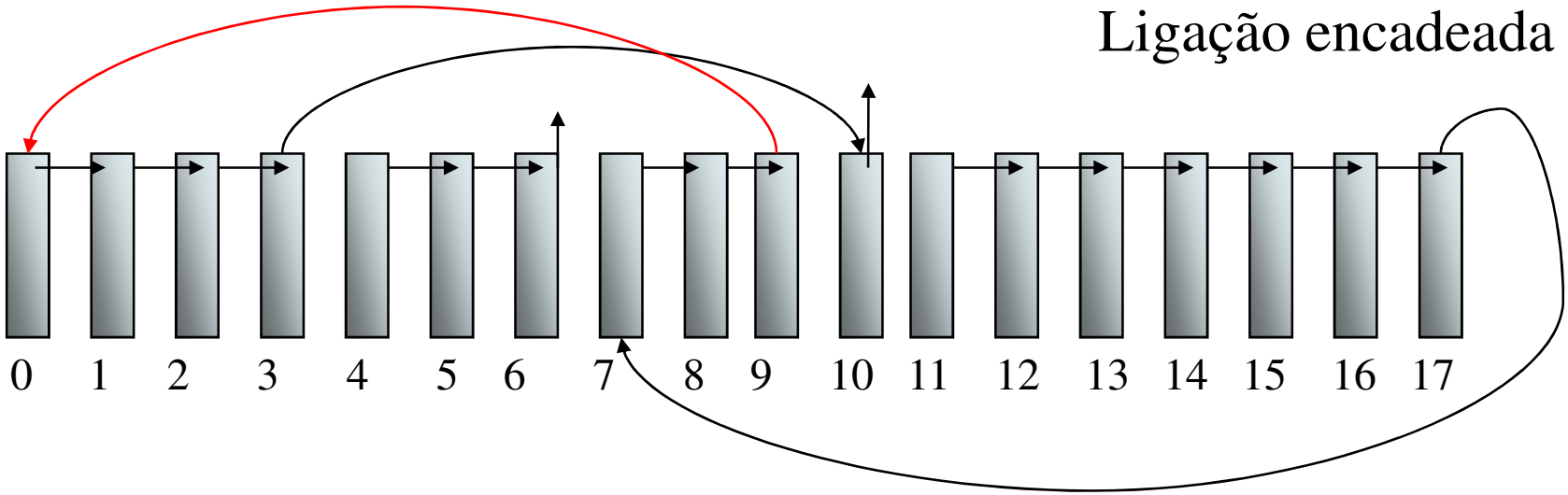
Ligação encadeada



NOME	Bloco Inicial	Tamanho
LBL	11	10K
Arquivo 1	0	4200
Arquivo 2	4	3000
?Arquivo 3	8	2000

Apagando os Arquivos 3 e 1.

Alocação de Disco

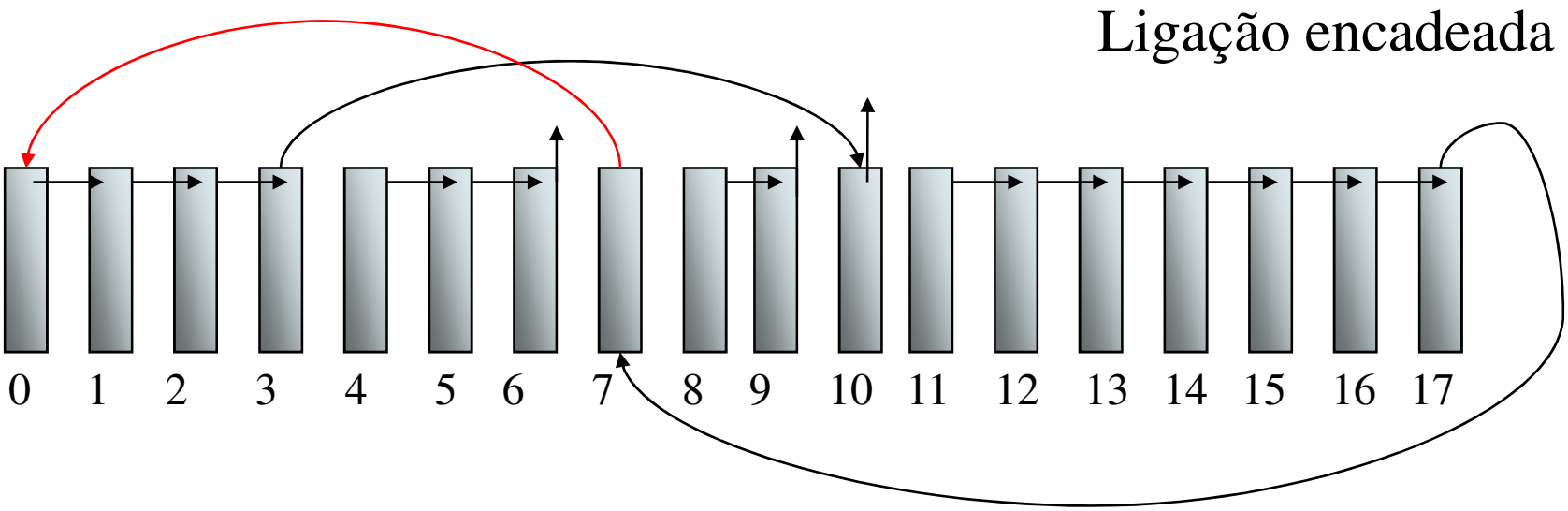


NOME	Bloco Inicial	Tamanho
LBL	11	15K
?Arquivo 1	0	4200
Arquivo 2	4	3000
?Arquivo 3	8	2000

Apagando os Arquivos 3 e 1.

Blocos da LBL
[11, 12, 13, 14, 15, 16, 17, 7, 8, 9, 0, 1, 2, 3, 10]

Alocação de Disco



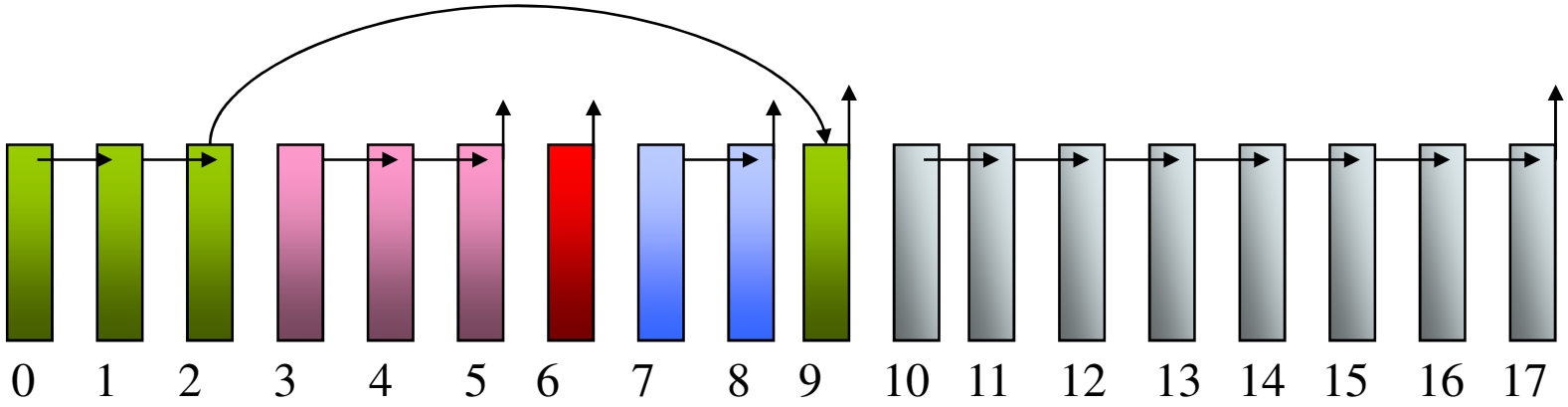
NOME	Bloco Inicial	Tamanho
LBL	11	13K
?Arquivo 1	0	4200
Arquivo 2	4	3000
Arquivo 3	8	2000

Restaurando o Arquivo 3.
O bloco 7 vai apontar para que o bloco 9 aponta. O 9 aponta para null.

Blocos da LBL
[11, 12, 13, 14, 15, 16, 17, 7, 0, 1, 2, 3, 10]

Problema na Alocação de Disco

Ligação encadeada



LBL → Bloco 10, 8KB

O que aconteceu com o bloco vermelho? Como reusá-lo?

Problema na Alocação de Disco

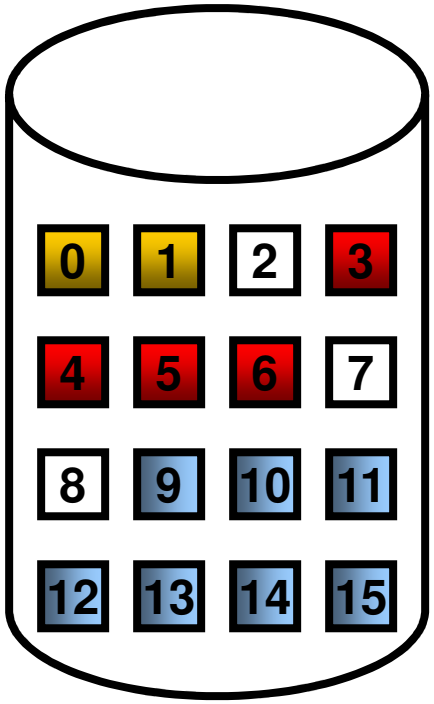
- Ligação encadeada
  - Para ler o bloco n de um arquivo, quantos blocos devem ser lidos?
    - Resp: n blocos
  - Por que?
    - Resp: Os blocos estão encadeados através de ponteiros dentro do próprio bloco
  - Existe leitura direta?
    - Resp: Não. Só seqüencial
  - Se o arquivo ficar fragmentado, a leitura seqüencial é pior que na alocação contígua.
    - Qual a solução?
      - Resp: Desfragmentação

Problema na Alocação de Disco

- Ligação encadeada
  - Qual a vantagem sobre a alocação contígua?
    - Menor desperdício, pois aloca blocos.
      - Alocação por demanda.

Alocação de Blocos em Disco

- Alocação Encadeada: BLOCO = 1024 bytes



Diretório

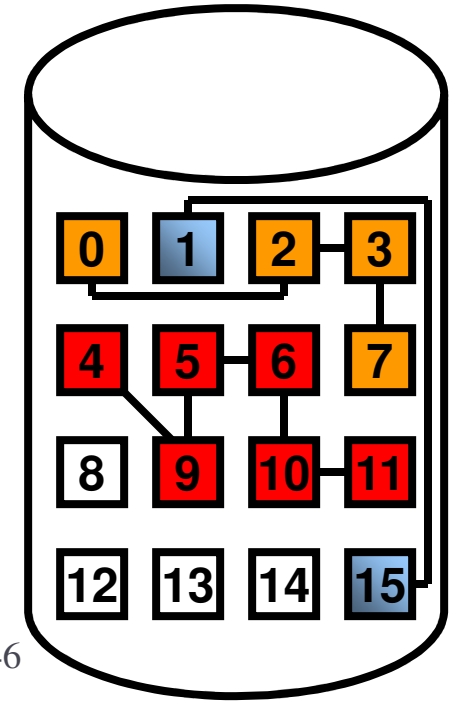
Arquivo	Início	Tamanho
Arquivo	0	2000
Arquivo	3	4010
Arquivo	9	7000

Alocação de Blocos em Disco

- Alocação Encadeada:
  - Arquivo é organizado como um conjunto de blocos ligados logicamente no disco
    - Arquivo é uma Lista Simplesmente Encadeada dentro dos blocos
  - Cada bloco possui um ponteiro para o bloco seguinte e assim sucessivamente
  - Fragmentação do disco não é problema, pois blocos não precisam estar contíguos

Alocação de Blocos em Disco

- Desvantagens:
  - Acesso aos blocos dos arquivos só pode ser seqüencial;
  - Espaço desperdiçado nos blocos para o **armazenamento dos ponteiros.**



Diretório

Arquivo	Início	Fim
Arquivo	0	7
Arquivo	4	11
Arquivo	15	1

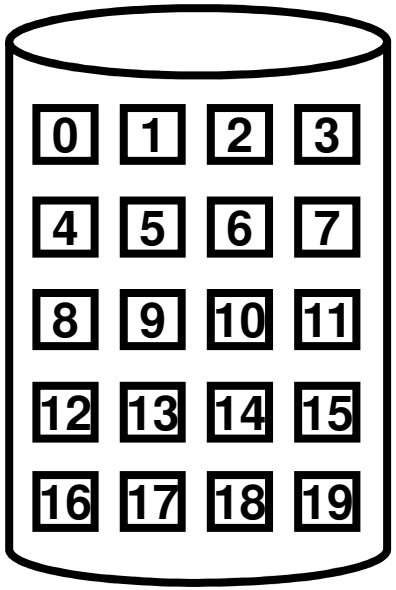
Alocação de Blocos em Disco

- Alocação Indexada:
  - Informações necessárias: Nome, endereço do índice + índice;
  - Ponteiros para os blocos do arquivo são mantidos numa estrutura chamada **bloco de índice**;
  - Permite acesso direto sem fragmentação;
  - Não utiliza informação de controle nos blocos como na alocação encadeada.

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10
10	11
11	12
12	13
13	14
14	15
15	16
16	17
17	18
18	19
19	999

FAT



Diretório

Arquivo	1o. Bloco	Tamanho
LBL	0	20480

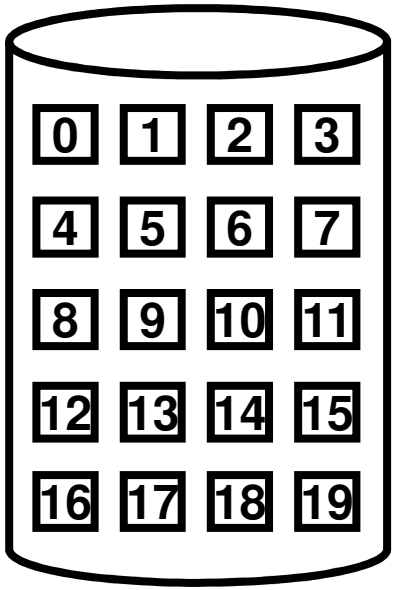
Bloco = 1 KB = 1024 bytes

A primeira coluna da tabela [linha] é virtual, pois identifica a linha.

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10
10	11
11	12
12	13
13	14
14	15
15	16
16	17
17	18
18	19
19	999

FAT



Diretório

Arquivo	1o. Bloco	Tamanho
LBL	0	20480

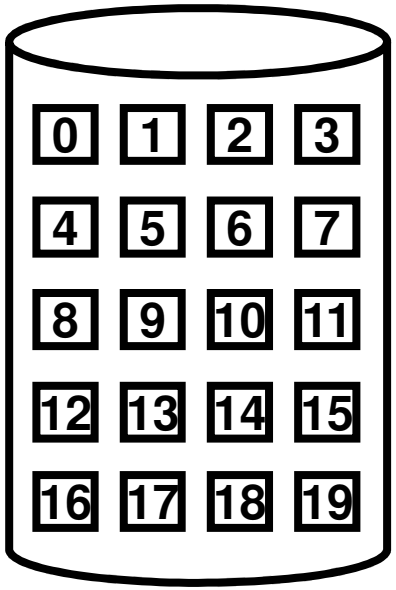
Qual o primeiro bloco da LBL?
Resp: 0
Qual o segundo bloco da LBL?
Resp: Na FAT, para onde o bloco 0 aponta? Busca-se na linha 0.

O último bloco deve apontar para EOF.
O que é EOF? Um valor predeterminado.
No exemplo assumiu-se 999.

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10
10	11
11	12
12	13
13	14
14	15
15	16
16	17
17	18
18	19
19	999

FAT



Diretório

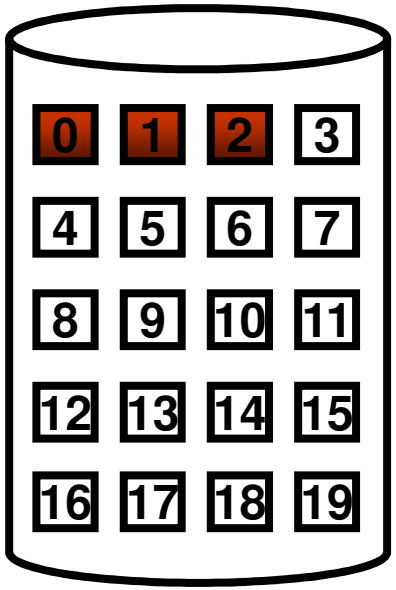
Arquivo	1o. Bloco	Tamanho
LBL	0	20480

Alocar o arquivo 1 com 2872 bytes
Precisamos de 3 blocos livres.

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	999
3	4
4	5
5	6
6	7
7	8
8	9
9	10
10	11
11	12
12	13
13	14
14	15
15	16
16	17
17	18
18	19
19	999

FAT



Diretório

Arquivo	1o. Bloco	Tamanho
LBL	3	17408
Arquivo 1	0	2872

Alocar o arquivo 1 com 2872 bytes
Precisamos de 3 blocos livres.

Bloco = 1 KB = 1024 Bytes

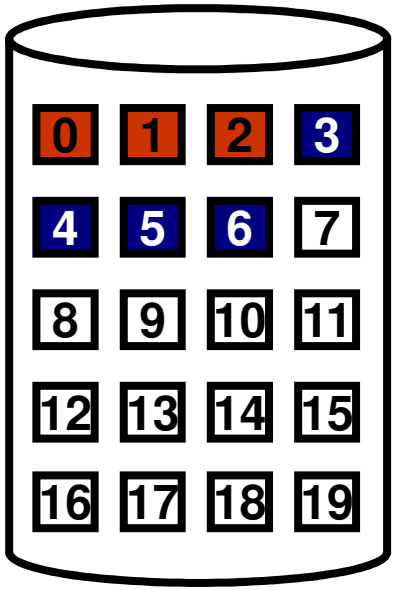
Blocos do Arquivo 1 = [0,1, 2]

Alocar o arquivo 2 com 4020 bytes

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	999
3	4
4	5
5	6
6	999
7	8
8	9
9	10
10	11
11	12
12	13
13	14
14	15
15	16
16	17
17	18
18	19
19	999

FAT



Aumentar o arquivo 1
em 1030 bytes

Diretório

Arquivo	1o. Bloco	Tamanho
LBL	7	13312
Arquivo 1	0	2872
Arquivo 2	3	4020

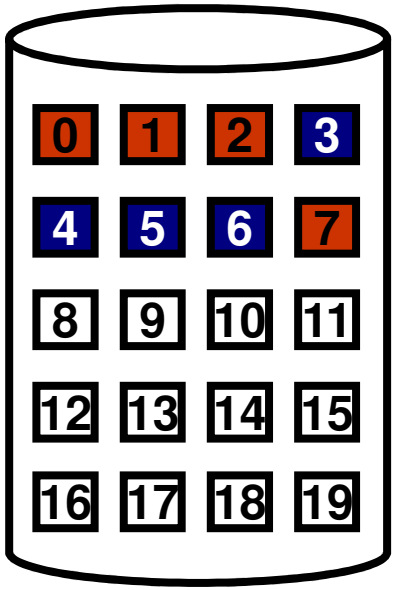
Alocar o arquivo 1 com 2872 bytes
Alocar o arquivo 2 com 4020 bytes

Bloco = 1 KB = 1024 Bytes
Blocos do Arquivo 1 = [0,1, 2]
Blocos do Arquivo 2 = [3, 4, 5, 6]

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	7
3	4
4	5
5	6
6	999
7	999
8	9
9	10
10	11
11	12
12	13
13	14
14	15
15	16
16	17
17	18
18	19
19	999

FAT



Criar o arquivo 3 com 5211 bytes

Diretório

Arquivo	1o. Bloco	Tamanho
LBL	8	12288
Arquivo 1	0	3902
Arquivo 2	3	4020

Aumentar o arquivo 1 em 1030 bytes

Bloco = 1 KB = 1024 Bytes

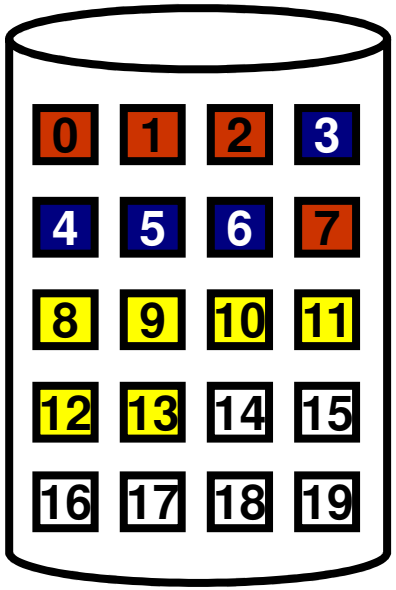
Blocos do Arquivo 1 = [0,1, 2, 7]

Blocos do Arquivo 2 = [3, 4, 5, 6]

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	7
3	4
4	5
5	6
6	999
7	999
8	9
9	10
10	11
11	12
12	13
13	999
14	15
15	16
16	17
17	18
18	19
19	999

FAT



Aumentar o arquivo 1 em 2405 bytes

Diretório

Arquivo	1o. Bloco	Tamanho
LBL	14	6144
Arquivo 1	0	3902
Arquivo 2	3	4020
Arquivo 3	8	5211

Criar o arquivo 3 com 5211 bytes

Bloco = 1 KB = 1024 Bytes

Blocos do Arquivo 1 = [0,1, 2, 7]

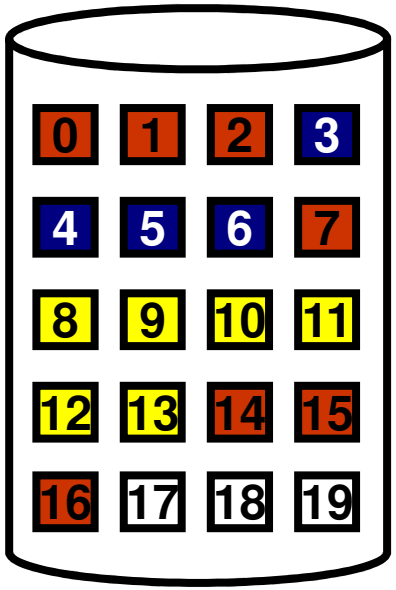
Blocos do Arquivo 2 = [3, 4, 5, 6]

Blocos do Arquivo 3 = [8, 9, 10, 11, 12, 13]

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	7
3	4
4	5
5	6
6	999
7	14
8	9
9	10
10	11
11	12
12	13
13	999
14	15
15	16
16	999
17	18
18	19
19	999

FAT



Apagar o arquivo 1

Diretório

Arquivo	1o. Bloco	Tamanho
LBL	17	3072
Arquivo 1	0	6307
Arquivo 2	3	4020
Arquivo 3	8	5211

Aumentar o arquivo 1 em 2405 bytes

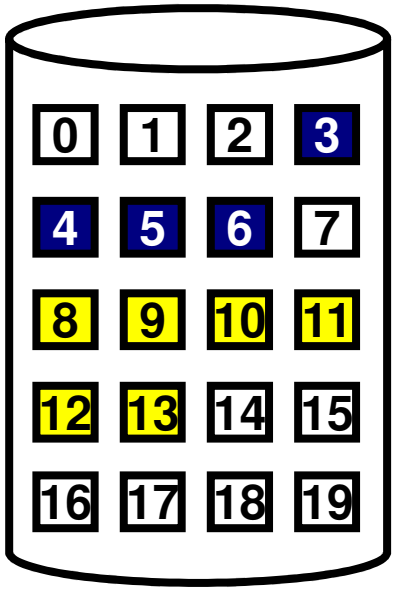
Bloco = 1 KB = 1024 Bytes
LBL = [17, 18, 19]

Blocos do Arquivo 1 = [0,1, 2, 7, 14, 15, 16]
Blocos do Arquivo 2 = [3, 4, 5, 6]
Blocos do Arquivo 3 = [8, 9, 10, 11, 12, 13]

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	7
3	4
4	5
5	6
6	999
7	14
8	9
9	10
10	11
11	12
12	13
13	999
14	15
15	16
16	999
17	18
18	19
19	0

FAT



Recuperar o arquivo 1

Diretório

Arquivo	1o. Bloco	Tamanho
LBL	17	10240
Arquivo 1	0	6307
Arquivo 2	3	4020
Arquivo 3	8	5211

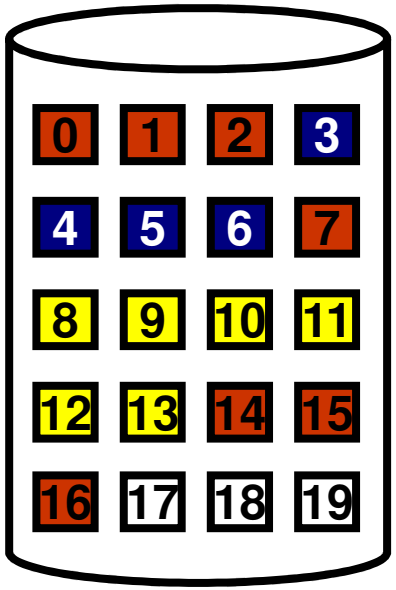
Bloco = 1 KB = 1024 Bytes
LBL = [17, 18, 19, 0, 1, 2, 7, 14, 15, 16]

Blocos do Arquivo 1 = [0,1, 2, 7, 14, 15, 16]
Blocos do Arquivo 2 = [3, 4, 5, 6]
Blocos do Arquivo 3 = [8, 9, 10, 11, 12, 13]

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	7
3	4
4	5
5	6
6	999
7	14
8	9
9	10
10	11
11	12
12	13
13	999
14	15
15	16
16	999
17	18
18	19
19	999

FAT



Apagar o arquivo 1 e o arquivo 2

Diretório

Arquivo	1o. Bloco	Tamanho
LBL	17	3072
Arquivo 1	0	6307
Arquivo 2	3	4020
Arquivo 3	8	5211

Recuperar o arquivo 1

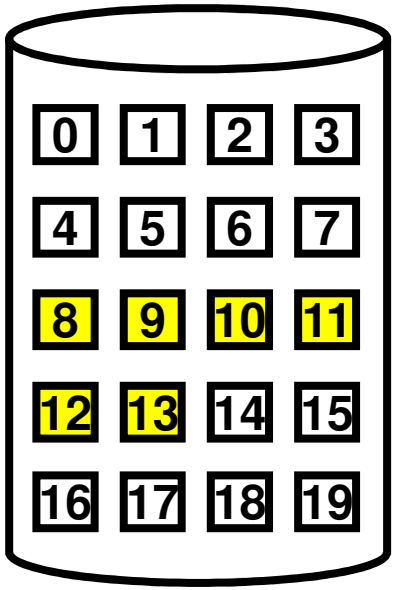
Bloco = 1 KB = 1024 Bytes
LBL = [17, 18, 19]

Blocos do Arquivo 1 = [0,1, 2, 7, 14, 15, 16]
Blocos do Arquivo 2 = [3, 4, 5, 6]
Blocos do Arquivo 3 = [8, 9, 10, 11, 12, 13]

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	7
3	4
4	5
5	6
6	999
7	14
8	9
9	10
10	11
11	12
12	13
13	999
14	15
15	16
16	3
17	18
18	19
19	0

FAT



Diretório

Arquivo	1o. Bloco	Tamanho
LBL	17	14336
Arquivo 1	0	6307
Arquivo 2	3	4020
Arquivo 3	8	5211

Bloco = 1 KB = 1024 Bytes

LBL = [17, 18, 19, 0, 1, 2, 7, 14, 15, 16, 3, 4, 5, 6]

Blocos do Arquivo 1 = [0,1, 2, 7, 14, 15, 16]

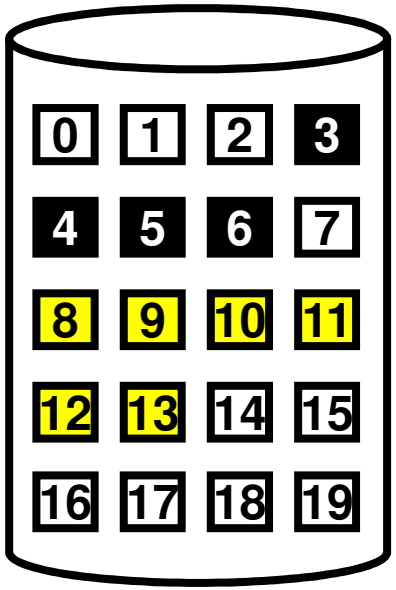
Blocos do Arquivo 2 = [3, 4, 5, 6]

Blocos do Arquivo 3 = [8, 9, 10, 11, 12, 13]

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	7
3	4
4	5
5	6
6	999
7	14
8	9
9	10
10	11
11	12
12	13
13	999
14	15
15	16
16	999
17	18
18	19
19	0

FAT



Diretório

Arquivo	1o. Bloco	Tamanho
LBL	17	10240
Arquivo 1	0	6307
Arquivo 2	3	4020
Arquivo 3	8	5211

Recuperar o arquivo 2

Bloco = 1 KB = 1024 Bytes

Blocos do Arquivo 1 = [0,1, 2, 7, 14, 15, 16]

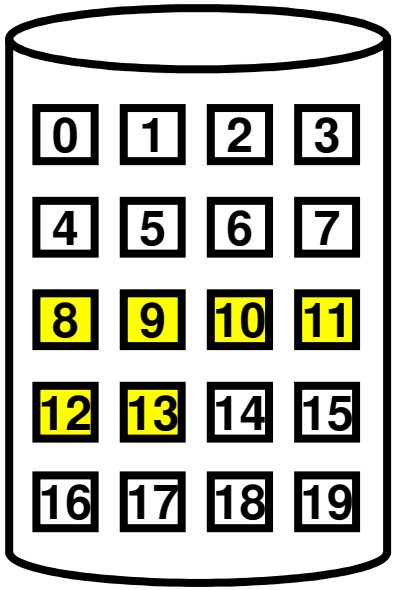
Blocos do Arquivo 2 = [3, 4, 5, 6]

Blocos do Arquivo 3 = [8, 9, 10, 11, 12, 13]

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	7
3	4
4	5
5	6
6	999
7	14
8	9
9	10
10	11
11	12
12	13
13	999
14	15
15	16
16	3
17	18
18	19
19	0

FAT



Diretório

Arquivo	1o. Bloco	Tamanho
LBL	17	14336
Arquivo 1	0	6307
Arquivo 2	3	4020
Arquivo 3	8	5211

Apagar o arquivo 1 e o arquivo 2

Bloco = 1 KB = 1024 Bytes

Blocos do Arquivo 1 = [0, 1, 2, 7, 14, 15, 16]

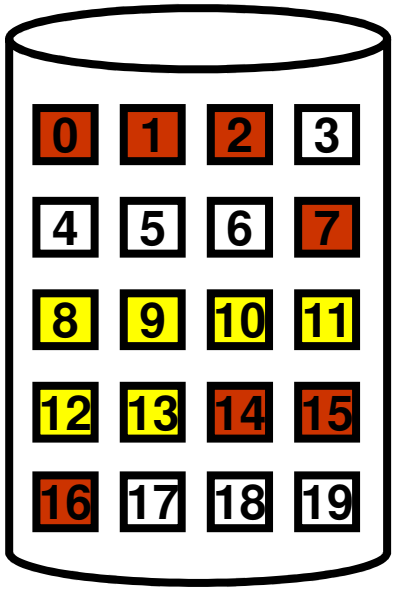
Blocos do Arquivo 2 = [3, 4, 5, 6]

Blocos do Arquivo 3 = [8, 9, 10, 11, 12, 13]

Alocação Indexada de Blocos

Linha	Valor
0	1
1	2
2	7
3	4
4	5
5	6
6	999
7	14
8	9
9	10
10	11
11	12
12	13
13	999
14	15
15	16
16	999
17	18
18	19
19	3

FAT



Diretório

Arquivo	1o. Bloco	Tamanho
LBL	17	7168
Arquivo 1	0	6307
Arquivo 2	3	4020
Arquivo 3	8	5211

Recuperar o arquivo 1

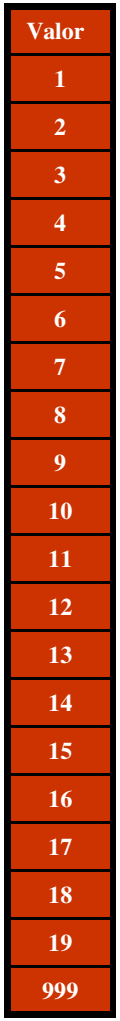
Bloco = 1 KB = 1024 Bytes

Blocos do Arquivo 1 = [0,1, 2, 7, 14, 15, 16]

Blocos do Arquivo 2 = [3, 4, 5, 6]

Blocos do Arquivo 3 = [8, 9, 10, 11, 12, 13]

FAT



FAT de 12 bits → 2^{12} = 4096 linhas → 4096 blocos
Cada linha da FAT12 possui 12 bits

Tamanho da FAT = Número de Linhas * Tamanho da Linha
= 2^{12} * 12 bits
= 4K * 1,5 bytes
= 6 K bytes

FAT 12 era usada em disquetes

Problema: como representar fim de arquivo, bloco ruim, etc?
Endereços especiais são usados.

FAT

Valor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
999

Tamanho do Bloco = Tamanho do Disco / 4096

Formatar disco de 1 MB → Bloco = $2^{20}/2^{12} = 2^8 = 256$ bytes
2 MB → Bloco = 512 bytes
4 MB → Bloco = 1024 bytes = 1KB
40 MB → Bloco = 10 KB

Quanto maior o disco, maior o tamanho do bloco.

Imagine formatar um disco de 64GB com FAT 12?
Bloco = $2^{36}/2^{12} = 2^{24}$ = 16 MBytes

FAT

Valor
1
2
3
4
5
6
7
8
9
10
11
...
1000
1002
...
...
65532
65533
65534
65535

FAT de 16 bits
Cada linha da FAT16 possui 16 bits

Tamanho da FAT16 = 2^{16} * 16 bits
= 64K * 2 bytes
= 128 K bytes

Exceção:

Precisamos de um valor para marcar Fim de Arquivo (FF), Bad Block (FE), etc.

Nem todos os blocos podem ser usados; os últimos são desperdiçados.

FAT

Valor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
999

FAT de 16 bits - 64 K Linhas - 64 K blocos
Tamanho da FAT = 128 K bytes

Formatando Discos:

64 MB → Tamanho do Bloco: 1 KB
640 MB → Bloco: 10KB
2 GB → Bloco: $2^{31}/2^{16} = 2^{15}$ = 32KB
6,4 GB → Bloco: 100 KB
64 GB → Bloco: 1 MB

A FAT precisa ficar TODA na memória para ser usada.

FAT

Valor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
999

FAT de 32 bits → 2^{32} linhas = 4 G Linhas

Tamanho da FAT 32 = 4 G * 4 bytes
= 16 GBytes

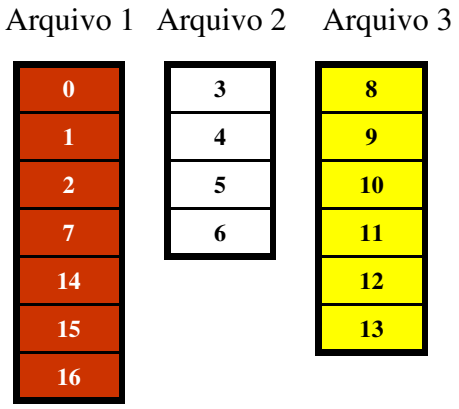
Formatando Partições:

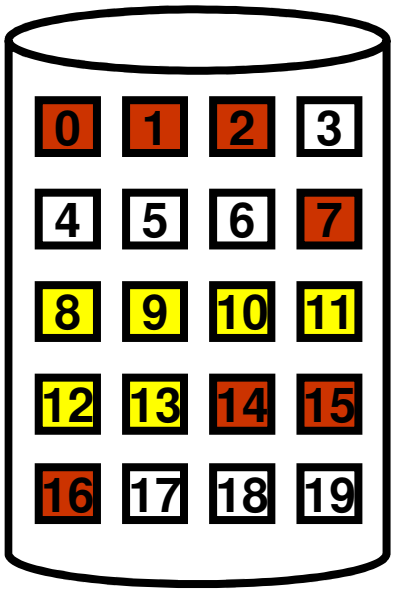
Bloco: 1 KB → Disco até 4 TB

Problema: Como manter a FAT32 na memória?

Solução: Trabalhar com tabelas com múltiplos níveis.

Alocação Indexada de Blocos





Diretório

Arquivo	1o. Bloco	Tamanho
LBL	17	7168
Arquivo 1	0	6307
Arquivo 2	3	4020
Arquivo 3	8	5211

Recuperar o arquivo 1

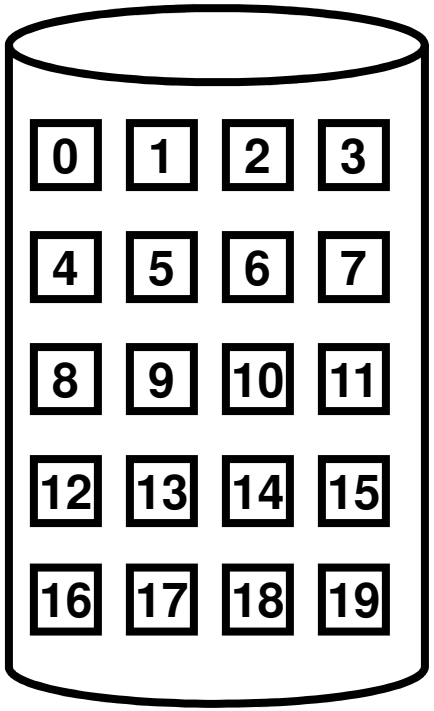
Bloco = 1 KB = 1024 Bytes

Blocos do Arquivo 1 = [0,1, 2, 7, 14, 15, 16]

Blocos do Arquivo 2 = [3, 4, 5, 6]

Blocos do Arquivo 3 = [8, 9, 10, 11, 12, 13]

Alocação Indexada de Blocos



Diretório

Arquivo	Índice
arquivo	0
arquivo	1
arquivo	2

Índice 0

0
-
-
-
-
-

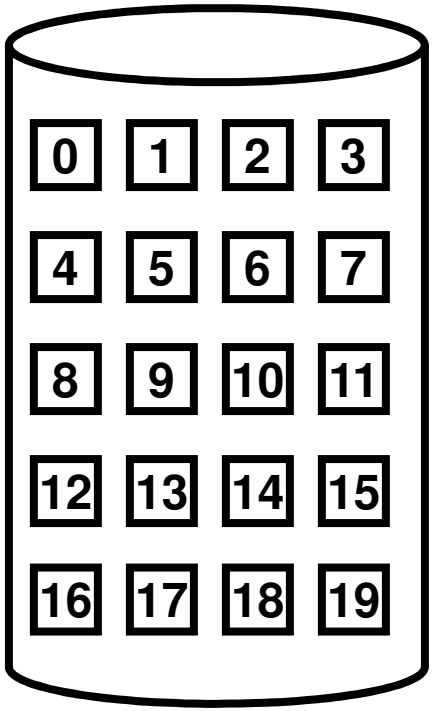
Índice 1

1
-
-
-
-
-

Índice 2

2
-
-
-
-
-

Alocação Indexada de Blocos



Diretório

Arquivo	Índice
arquivo	0
arquivo	1
arquivo	2

Índice 0

0
-
-
-
-
-

Índice 1

1
-
-
-
-
-

Índice 2

2
-
-
-
-
-

Tabelas de Índices

- Suponha um disco de 80 GB, bloco de 4 KB
- O disco está cheio
- Quanto de tabela, com 32 bits, será necessário?
  - Número de Blocos = 80 GB / 4 KB = 20 M
  - Tamanho da tabela = 20 M * 4 bytes = 80 MB
  - Arquivos têm:
    - Nome
    - Atributos (Tamanho, datas, permissões, etc)
      - Onde armazenar isso?

Diretórios

Diretórios

- Organização
- Todo diretório é um arquivo
  - Árvore
    - Raiz – Nome da Partição
      - C:
  - Diretório é um arquivo com registros

Diretórios

C:

5
9999
9999
4
6
9999
7
8
9
9999
9999

Nome	Extensão	SDAHRrrr	1o. Bloco	Tamanho	Data
Windows		11001	3	6200	15/08/2006
Autoexec	bat	00000	1	312	15/08/2006
Config	sys	00001	2	450	15/08/2006
Teste	txt	00000	10	82	29/08/2006

O que ocorre quando executamos dir C:
O arquivo C: “root” tem 2 blocos: 0, 5.
O sistema abre todos os blocos e apresenta o conteúdo formatado.

Dir C:\Windows
Dir C:\XPTO
Notepad C:\teste.txt

```
C:\>dir
 O volume na unidade C não tem nome.
 O número de série do volume é 38E7-FFAE

 Pasta de C:\
28/08/2006  15:17    <DIR>          Windows
15/08/2006  12:20               312 AUTOEXEC.BAT
15/08/2006  12:20               450 CONFIG.SYS
...
```

Alocação Indexada de Blocos

Linha	Valor
0	999
1	999
2	7
3	4
4	5
5	6
6	999
7	14
8	9
9	10
10	11
11	12
12	13
13	999
14	15
15	16
16	999
17	18
18	19
19	999

Conteúdo do Bloco 0

Nome	SDAHR	1o. Bloco	Tamanho	Data
Windows	11001	3	6200	15/08/2006
Autoexec.bat	00000	1	312	15/08/2006
Config.sys	00001	2	450	15/08/2006
Teste.txt	00000	10	82	29/08/2006

Bloco = 1 KB = 1024 Bytes
ROOT: Bloco 0

O que acontece quando executamos DIR C:

Cache

Cache

- Como garantir maior desempenho para acessos repetidos a um mesmo bloco de disco?
  - Cache de Blocos na memória
  - Desempenho
  - Contém os blocos mais recentemente referenciados do disco
    - LRU

Cache: Desempenho X Confiabilidade

- Tipos de Cache
  - Maior desempenho, Menor confiabilidade
    - Lê e grava apenas na Cache
    - O que acontece se faltar energia ao sistema?
      - Todas as alterações da Cache são perdidas.
      - Problema de consistência ao SO e aos dados dos usuários.
    - Quando os dados são realmente gravados no disco?
      - Quando houver uma falta de bloco com o cache cheio.
      - Algoritmo LRU
      - Temporizador

Cache: Desempenho X Confiabilidade

- Tipos de Cache
  - Maior confiabilidade, Menor desempenho
    - Lê na Cache e grava na Cache e no Disco
    - O que acontece se faltar energia ao sistema?
      - Nenhuma alteração é perdida.
    - Quando os dados são realmente gravados no disco?
      - A cada I/O
    - Algoritmo LRU

Cache: Desempenho X Confiabilidade

- Tipos de Cache
  - Equilíbrio entre confiabilidade e desempenho
    - Os blocos dos arquivos do SO e do usuário têm a mesma importância ao sistema?
      - Não.
      - Tipos de Blocos:
        - Essenciais → do SO (Diretórios, Estruturas)
        - Não Essenciais → usuários

Cache: Desempenho X Confiabilidade

- Tipos de Cache
  - Equilíbrio entre confiabilidade e desempenho
    - I/O de bloco essencial
      - Lê na Cache e grava na Cache e no Disco
    - I/O de bloco não essencial
      - Lê e Grava na Cache
    - O que acontece se faltar energia ao sistema?
      - O sistema fica íntegro e as alterações do usuário são perdidas.