



SISTEMAS OPERACIONAIS

Diretórios
Gerenciamento de Espaço em Disco

Implementação de Diretórios (1)

- Contém informações que permitem acessar os arquivos
 - As entradas do diretório fornecem informações para encontrar os blocos de discos
- Possui várias entradas, uma por arquivo:
 - nome
 - tipo; tamanho
 - proprietário; proteção
 - data de criação; data da última modificação
 - **lista de blocos usados**

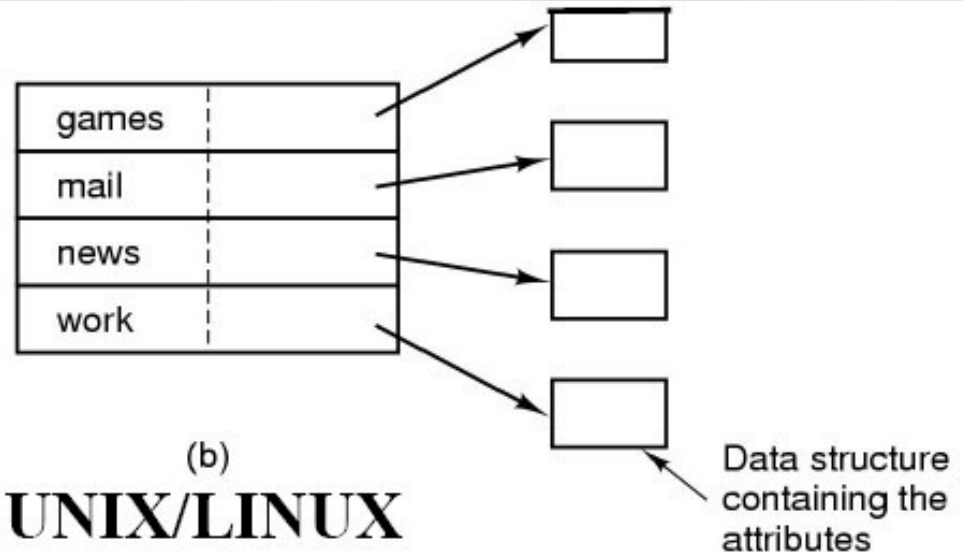
Implementação de Diretórios (2)

- (a) Diretório simples com
 - Entradas de dimensão fixa
 - Endereços de disco e atributos na entrada de diretório
- (b) Diretório em que cada entrada apenas refere um i-node

games	attributes
mail	attributes
news	attributes
work	attributes

(a)

MS-DOS/WINDOWS

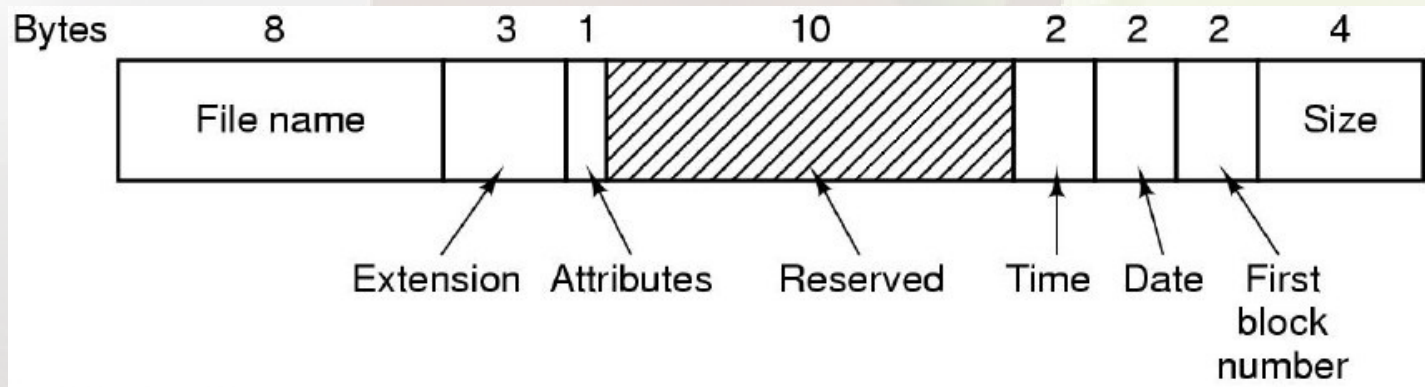


(b)

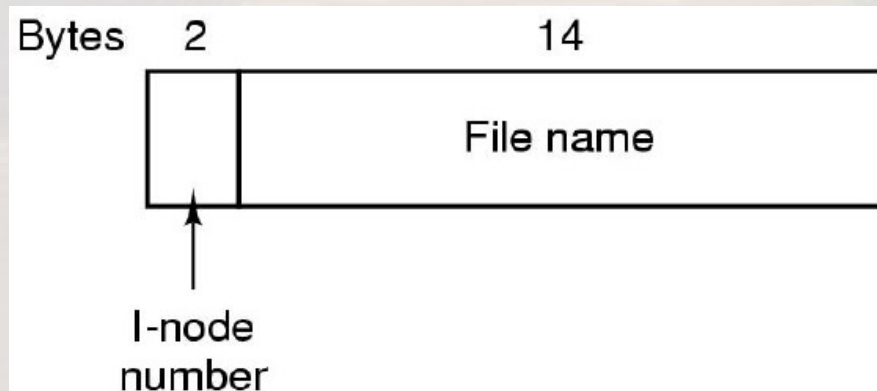
UNIX/LINUX

Implementação de Diretórios (3)

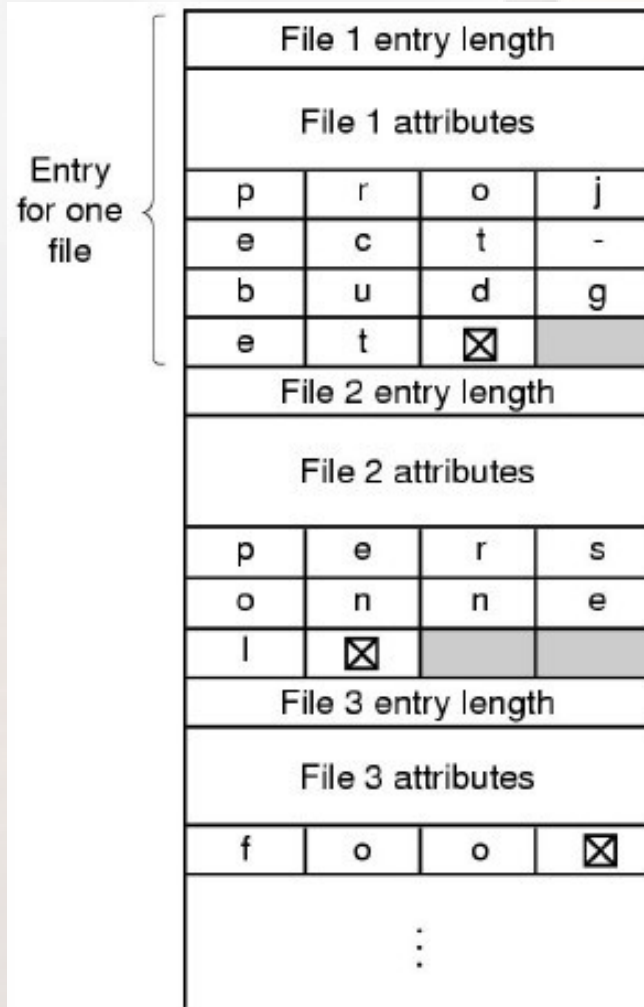
□ Entrada de diretório no DOS



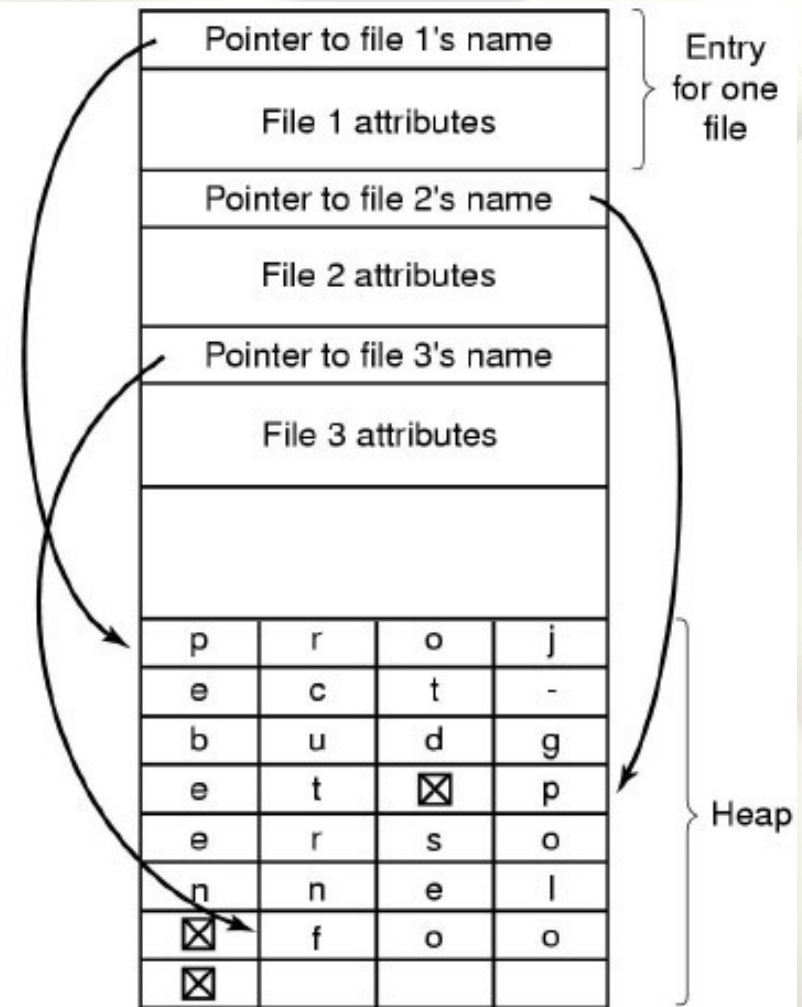
□ No Unix



Implementação de Diretórios (4)

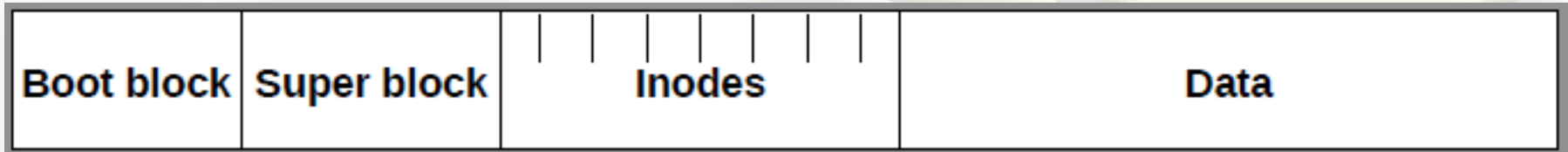


In-line



Em heap

Implementação de Diretórios (5)



Root directory

1	.
1	..
4	bin
7	dev
14	lib
9	etc
6	usr
8	tmp

I-node 6
is for /usr

Mode size times
132

Block 132
is /usr
directory

6	.
1	..
19	dick
30	erik
51	jim
26	ast
45	bal

I-node 26
is for
/usr/ast

Mode size times
406

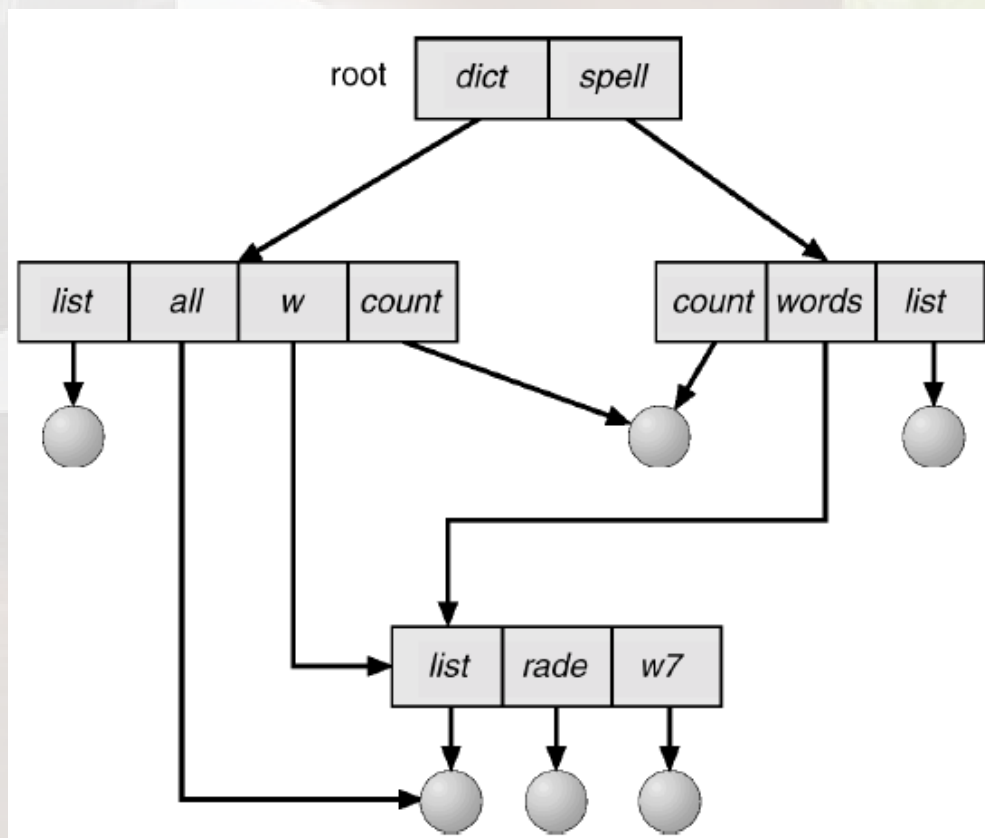
Block 406
is /usr/ast
directory

26	.
6	..
64	grants
92	books
60	mbox
81	minix
17	src

Quais os passos para alcançar o arquivo
/usr/ast/mbox?

Arquivos Compartilhados (1)

- Hierarquia de diretórios: Grafo acíclico orientado

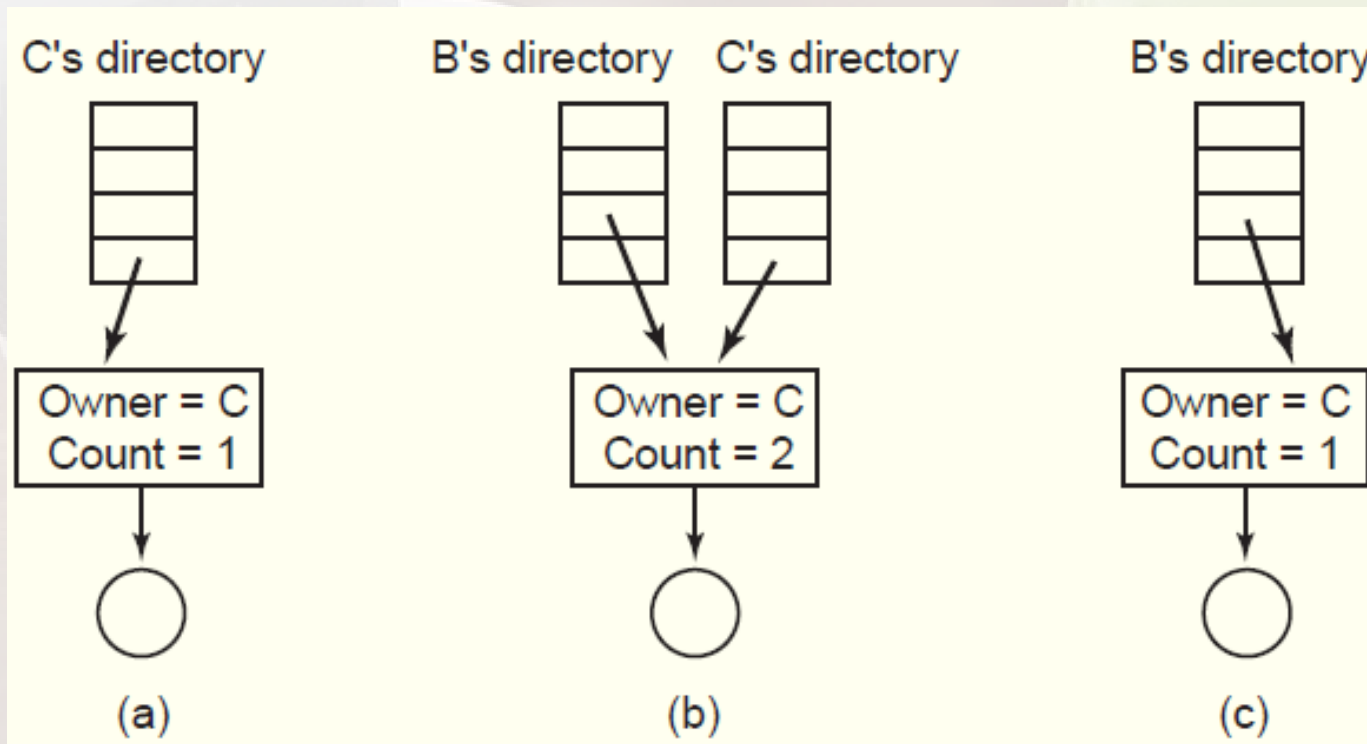


Arquivos Compartilhados (2)

- O diretório contém a lista (endereços) dos blocos que pertencem ao arquivo
 - é feita uma cópia dos endereços dos blocos para o diretório do arquivo “link”
 - Problema: não existe compartilhamento,
 - mudanças em uma versão (e.g. append no fim do arquivo) não são vistas em outra
- Soluções
 - 1a. Solução: os blocos não fazem parte do diretório, mas sim de estruturas de dados associadas aos descritores. O diretório aponta para essa estrutura de dados (UNIX)
 - 2a. Solução: “link” simbólico - o diretório contém o nome do arquivo “linkado”

Arquivos Compartilhados (3)

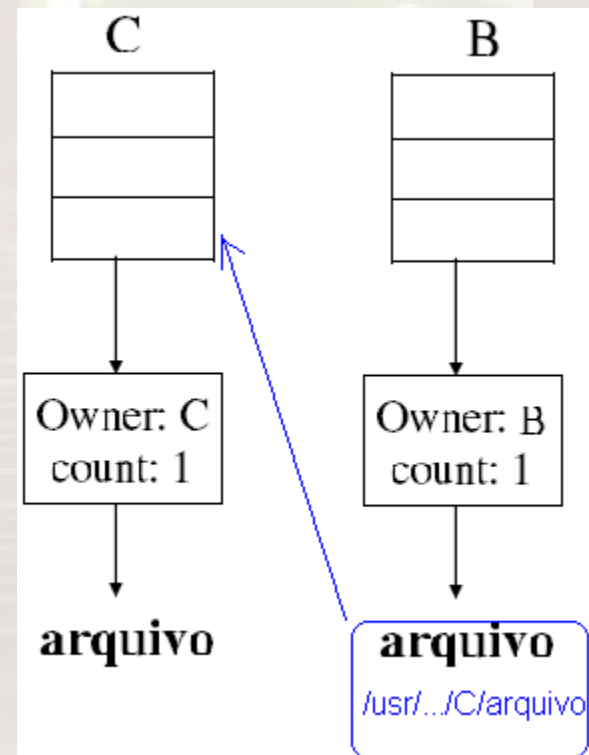
□ 1a. Solução



Arquivos Compartilhados (3)

□ 2a. Solução

- não existe o problema de deleção do arquivo por parte do proprietário
- Problema: número de acessos a disco pode ser elevado
- Vantagem: link de arquivos em máquinas diferentes



Arquivos Compartilhados (4)

Hard Link

directory entry in /dirA

inode	name
12345	name1

directory entry in /dirB

inode	name
12345	name2

inode 12345

⋮
2
⋮
23567
⋮

block 23567

"This is the
text in the
file."

Soft Link

directory entry in /dirA

inode	name
12345	name1

directory entry in /dirB

inode	name
13579	name2

inode 12345

⋮
1
⋮
23567
⋮

block 23567

"This is the
text in the
file."

inode 13579

⋮
1
⋮
15213
⋮

block 15213

"/dirA/name1"

Criando Hard Links (SVC)

```
#include <sys/unistd.h>

int link (const char *path1, const char *path2);
    // Cria um hard link (path2 -> path1)
int unlink (const char *path1, const char *path2);
    // Apaga um hard link
```

□ Exemplo: criando um hard link

```
#include <stdio.h>
#include <sys/stat.h>
...
    if (link("/dirA/name1", "/dirB/name2") == -1)
        perror("Failed to make a new link in /dirB");
...
```

In /dirA/name1 /dirB/name2

Criando Soft Links (SVC)

□ SVC p/ criação de link simbólicos

```
#include <sys/unistd.h>
```

```
int symlink (const char *path1, const char *path2);  
    // Cria um link simbólico (path2 -> path1)
```

```
ln -s path1 path2
```

Gerenciamento de Espaço em Disco (1)

□ Tamanho de Bloco

- Bloco Grande
 - Menos acessos a disco
 - Aumenta fragmentação interna
- Bloco Pequeno
 - Diminui a fragmentação interna
 - Arquivo contendo muitos blocos => acesso mais lento
- Tempo para se ler um bloco

Tmp médio de seek + latência rotacional + tempo de leitura

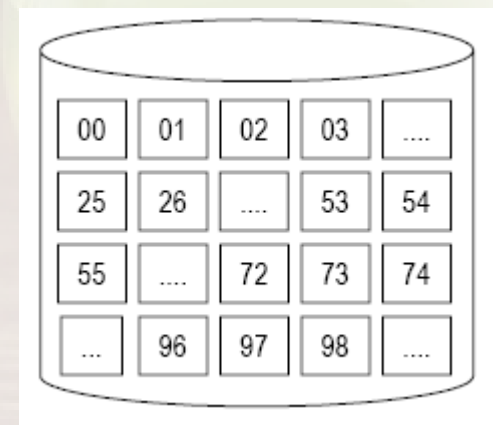
□ Exemplo:

- Considere um disco com 131.072 bytes por trilha, tmp de rotação = 8,33 ms tmp médio de seek = 10 ms. Tmp p/ ler um bloco de k bytes?
 - ler um bloco de k bytes = $10 + 4,16 + (k/131072) \times 8,33$

Gerenciamento de Espaço em Disco (2)

□ Gerenciamento do Espaço Livre

- Necessário manter a informação de blocos livres e ocupados
- Métodos Básicos
 - Mapa de bits
 - Lista de blocos livres

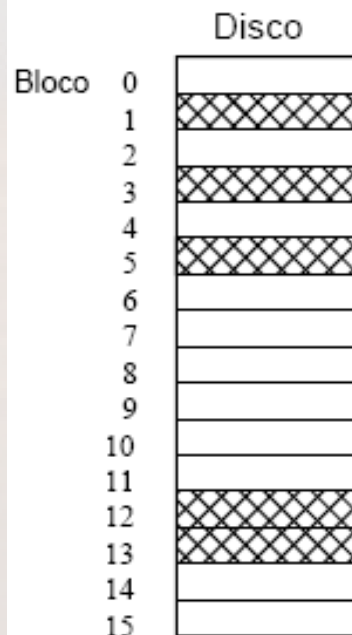


- ## □ Ambos os métodos consideram que os blocos são numerados sequencialmente

Gerenciamento de Espaço em Disco (3)

□ Mapa de bits

- Forma simples de gerenciar o espaço em disco
- Um disco c/ n blocos requer um bitmap de



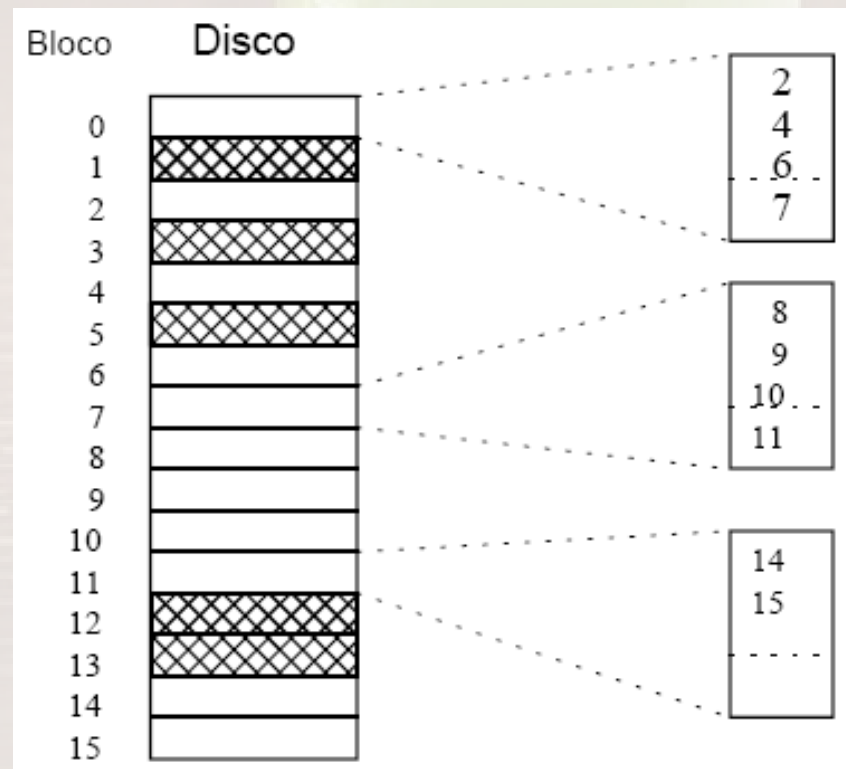
Mapa de Bits.

0	0	1	0	1	0	1	0
0	0	1	1	0	0	0	0

$$tamanho_bit_map = \frac{Capacidade_disco(bytes)}{8 \times tamanho_bloco(bytes)}$$

Gerenciamento de Espaço em Disco (4)

- Lista encadeada de blocos livres
 - A lista é mantida no próprio disco
 - Problema: tamanho da lista
 - Paliativo: a medida que o espaço em disco é ocupado, a lista diminui, liberando blocos de tamanho
 - Alternativa: manter uma lista de “áreas livres” ao invés de blocos

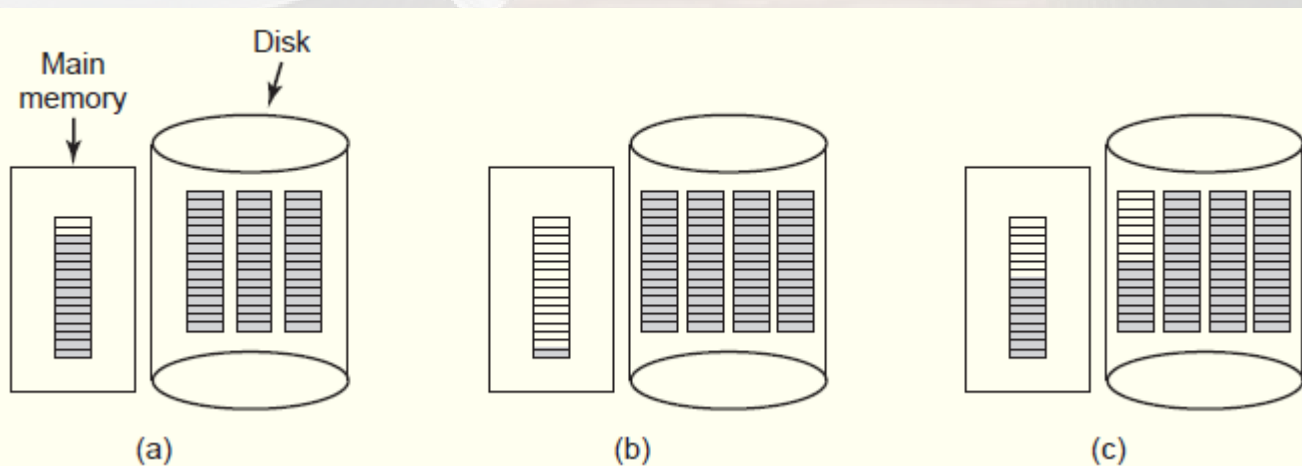
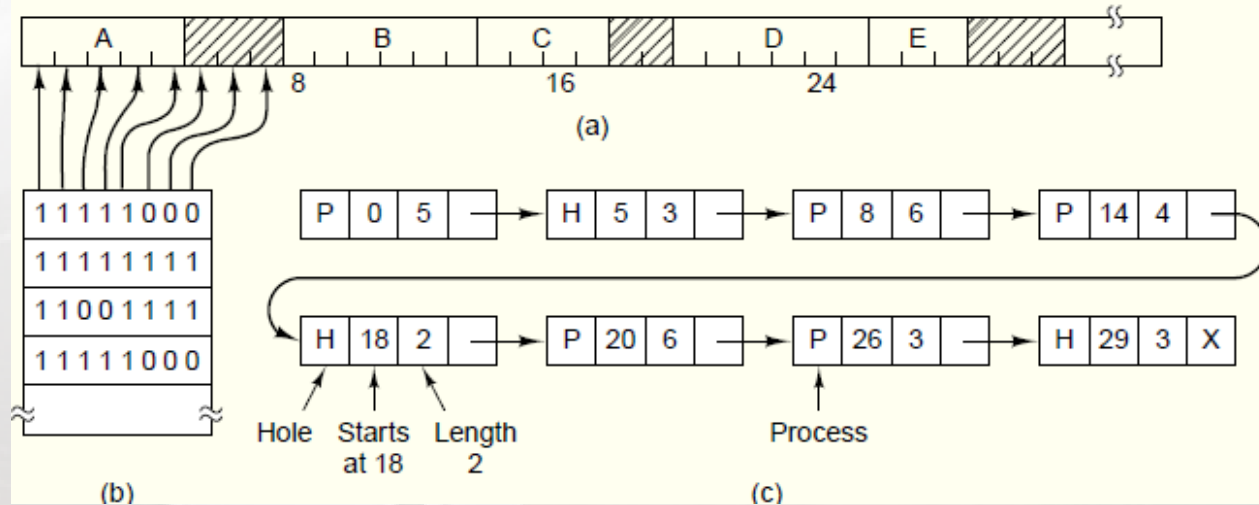


Gerenciamento de Espaço em Disco (5)

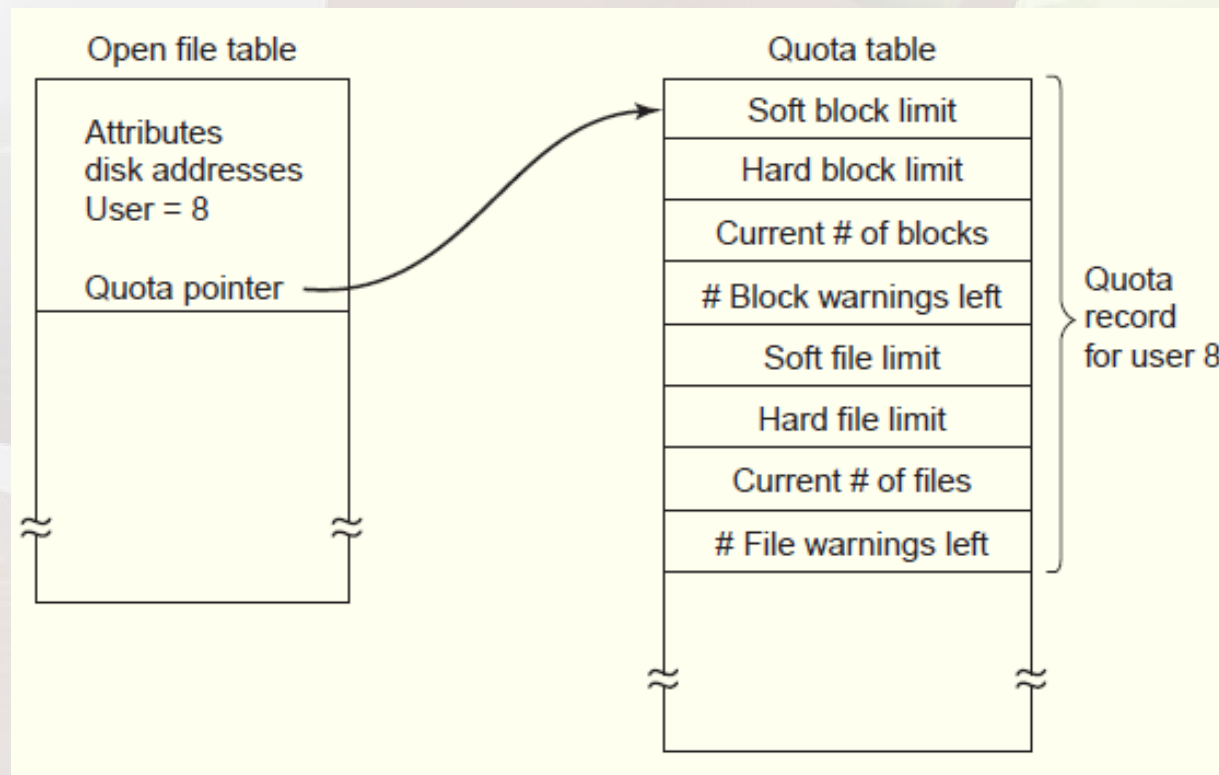
□ Lista de blocos livres

- Quantos blocos precisamos para armazenar a lista de blocos livres?
- Exemplo:
 - Disco de 40 M com Blocos de 1k
 - 2 bytes para identificar bloco livre
 - Cada bloco pode conter 512 entradas
 - Disco de 40 M => dividido em 40 k blocos
 - no de blocos = $40k/512 = 80$ blocos

Gerência de Memória x Gerenciamento de Espaço em Disco



Gerenciamento de Cotas



Consistência do Sistema de Arquivos

Block number															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0
Blocks in use															
0	0	1	0	1	0	0	0	0	1	1	0	0	0	1	1
Free blocks															

(a)

(a) consistente

Block number															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0
Blocks in use															
0	0	1	0	2	0	0	0	0	1	1	0	0	0	1	1
Free blocks															

(c)

(c) duplicação na
lista de livres

Block number															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0
Blocks in use															
0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	1
Free blocks															

(b)

(b) bloco faltando

Block number															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	1	0	2	1	1	1	0	0	1	1	1	0	0
Blocks in use															
0	0	1	0	1	0	0	0	0	1	1	0	0	0	1	1
Free blocks															

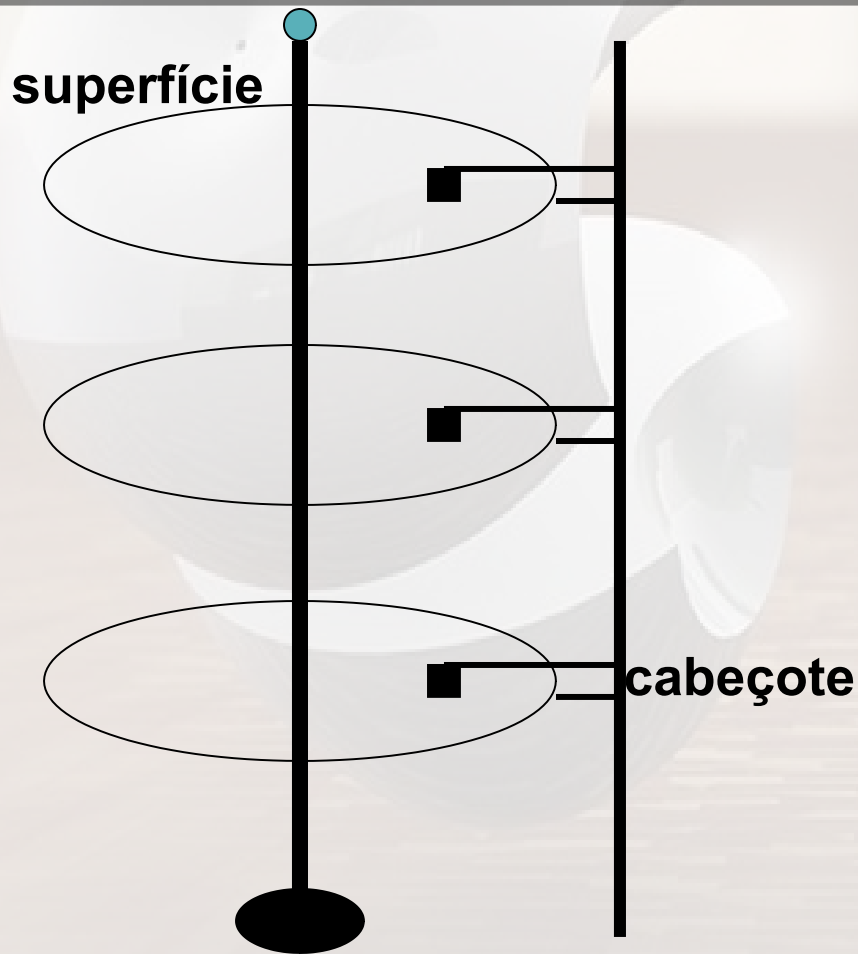
(d)

(d) duplicação nos
dados

Desempenho

- Reduzir o movimento do braço do disco: colocando os blocos que são mais prováveis de serem acessados próximos uns dos outros em sequência (mesmo cilindro do disco)
 - O gerenciamento do disco é feito por grupos de blocos consecutivos e não somente por blocos;

Dispositivos de E/S - Discos



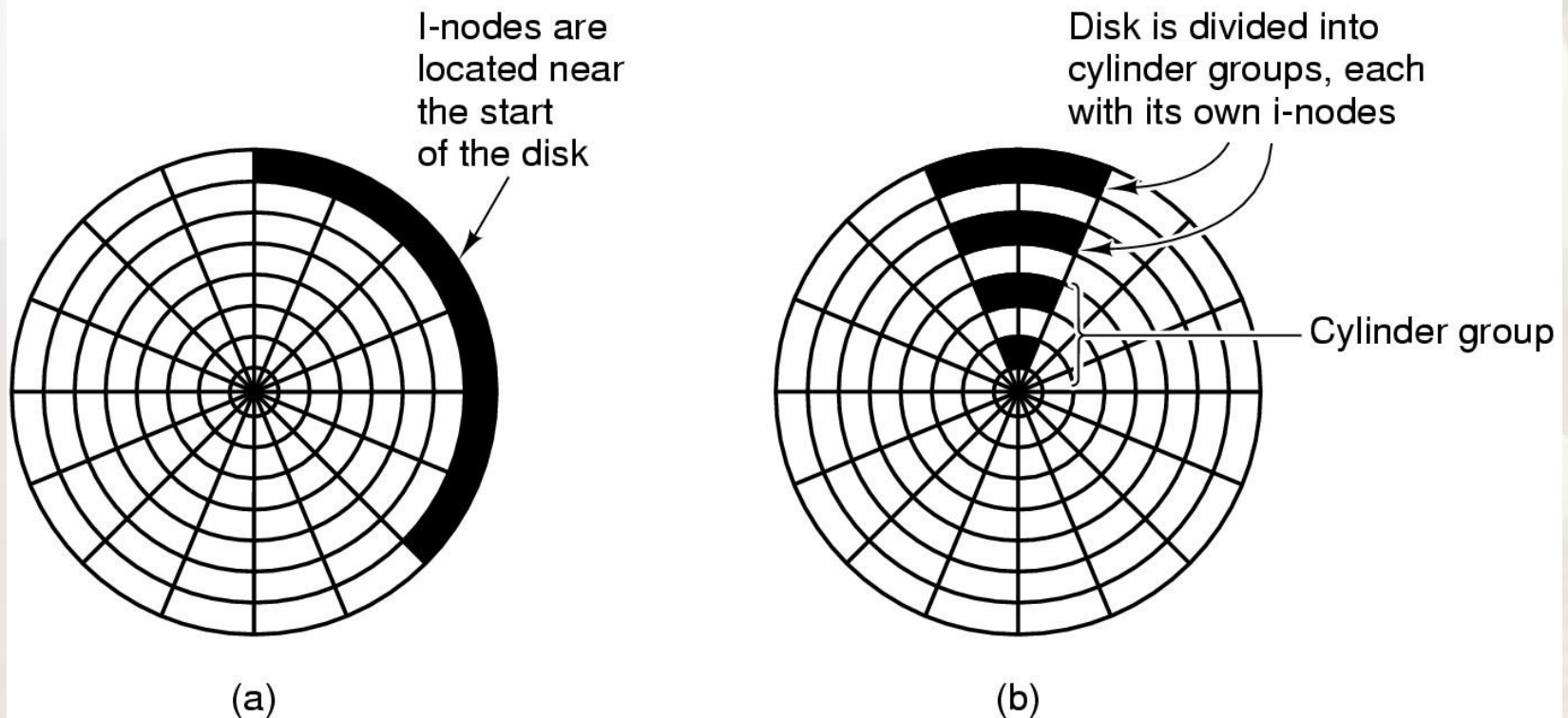
Disco Magnético

- Cada superfície é dividida em **trilhas**;
- Cada trilha é dividida em **setores** ou **blocos** (512 bytes a 32K);
- Um conjunto de trilhas (com a mesma distância do eixo central) formam um **cilindro**;
- Cabeças de leitura e gravação;
- Tamanho do disco:
nº cabeças (faces) x
nº cilindros (trilhas) x
nº setores x
tamanho_setor;

Desempenho

- Para sistemas que utilizam os i-nodes, são necessários dois acessos: uma para o bloco e outro para o i-node;
- Três estratégias podem ser utilizadas para armazenamento dos i-nodes:
 - A) Os i-nodes são colocados no início do disco;
 - B) Os i-nodes são colocados no meio do disco;
 - C) Dividir o disco em grupos de cilindros, nos quais cada cilindro tem seus próprios i-nodes, blocos e lista de blocos livres (bitmap);

Implementando o Sistema de Arquivos – Desempenho



Desempenho

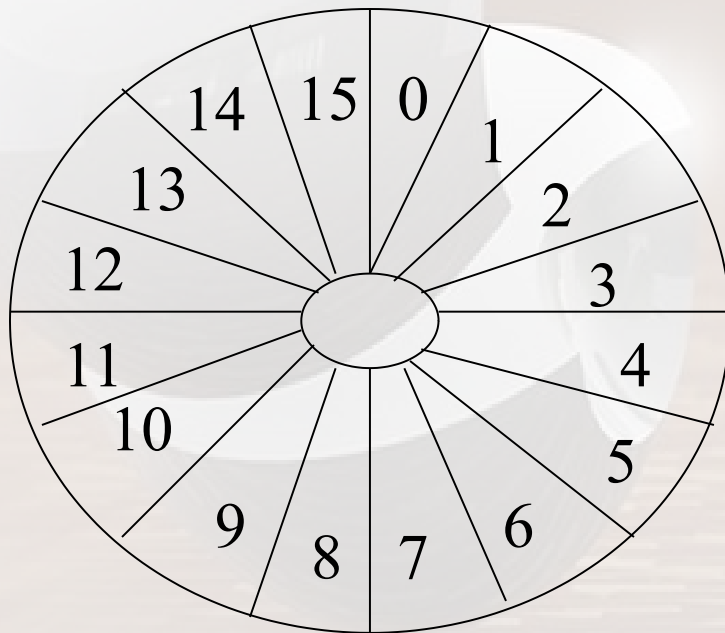
- Discos Magnéticos:
 - Grande evolução em relação a:
 - Velocidade de acesso (*seek*): tempo de deslocamento do cabeçote até o cilindro correspondente à trilha a ser acessada;
 - Transferências: tempo para transferência (leitura/escrita) dos dados;
 - Capacidade;
 - Preço;

Desempenho

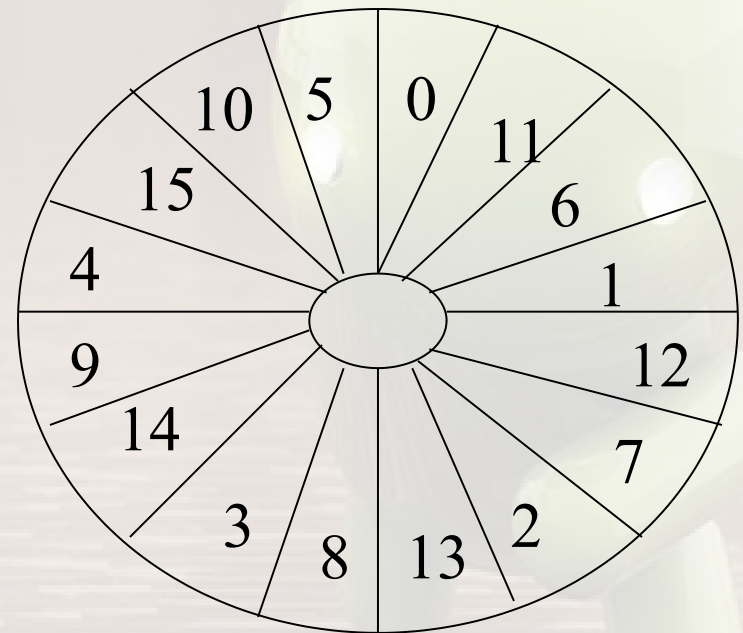
- Técnica para reduzir o tempo de acesso: entrelaçamento (*interleaving*):
 - Setores são numerados com um espaço entre eles;
 - Entre o **setor K** e o **setor K+1** existem **n** (fator de entrelaçamento) setores;
 - Número **n** depende da velocidade do processador, do barramento, da controladora e da velocidade de rotação do disco;

Desempenho

Trilhas com 16 setores



Disco A
 $N = 0$



Disco B
 $N = 2$

Desempenho

□ Drivers de Disco:

- Fatores que influenciam tempo para leitura/escrita no disco:
 - Velocidade de acesso (seek) → tempo para o movimento do braço até o cilindro;
 - Delay de rotação (latência) → tempo para posicionar o setor na cabeça do disco;
 - Tempo da transferência dos dados;
- Tempo de acesso:
 - $T_{seek} + T_{latência}^* + T_{transferência}$;

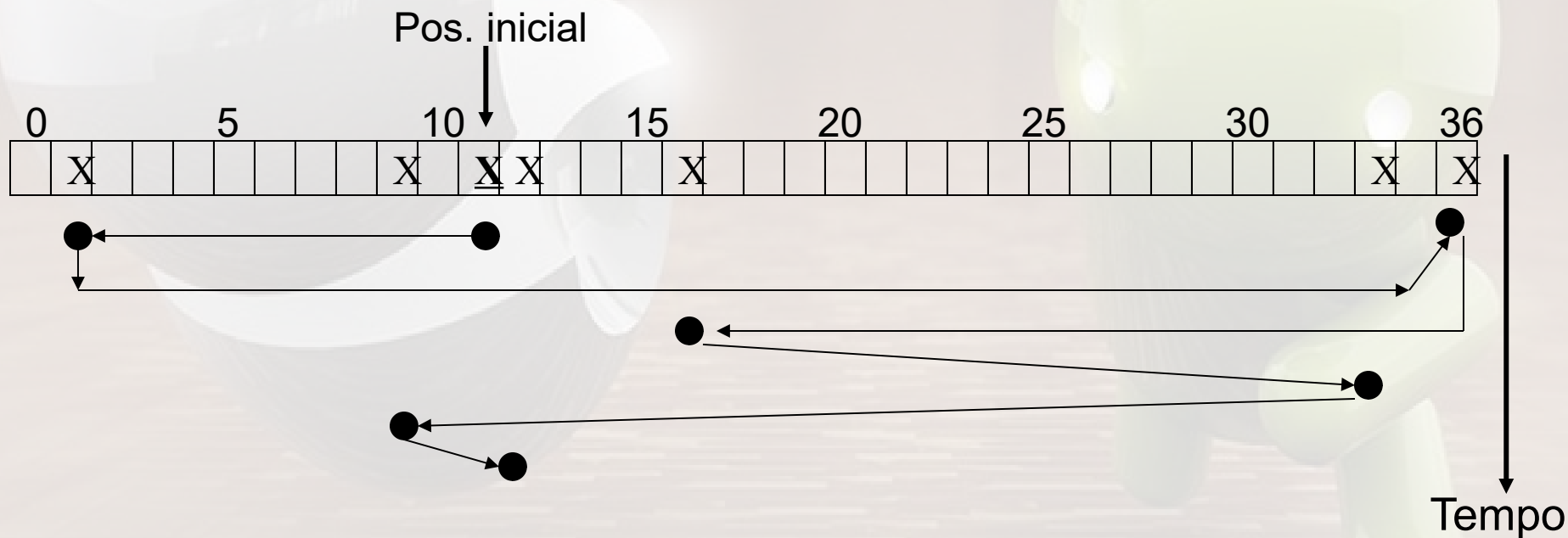
* Tempo necessário para o cabeçote se posicionar no setor de escrita/leitura;

Desempenho

- Algoritmos de escalonamento no disco:
 - FCFS (FIFO) → *First-Come First-Served*;
 - SSF → *Shortest Seek First*;
 - Elevator (também conhecido como SCAN);
- Escolha do algoritmo depende do número e do tipo de pedidos;
- Driver mantém uma lista encadeada com as requisições para cada cilindro;

Desempenho

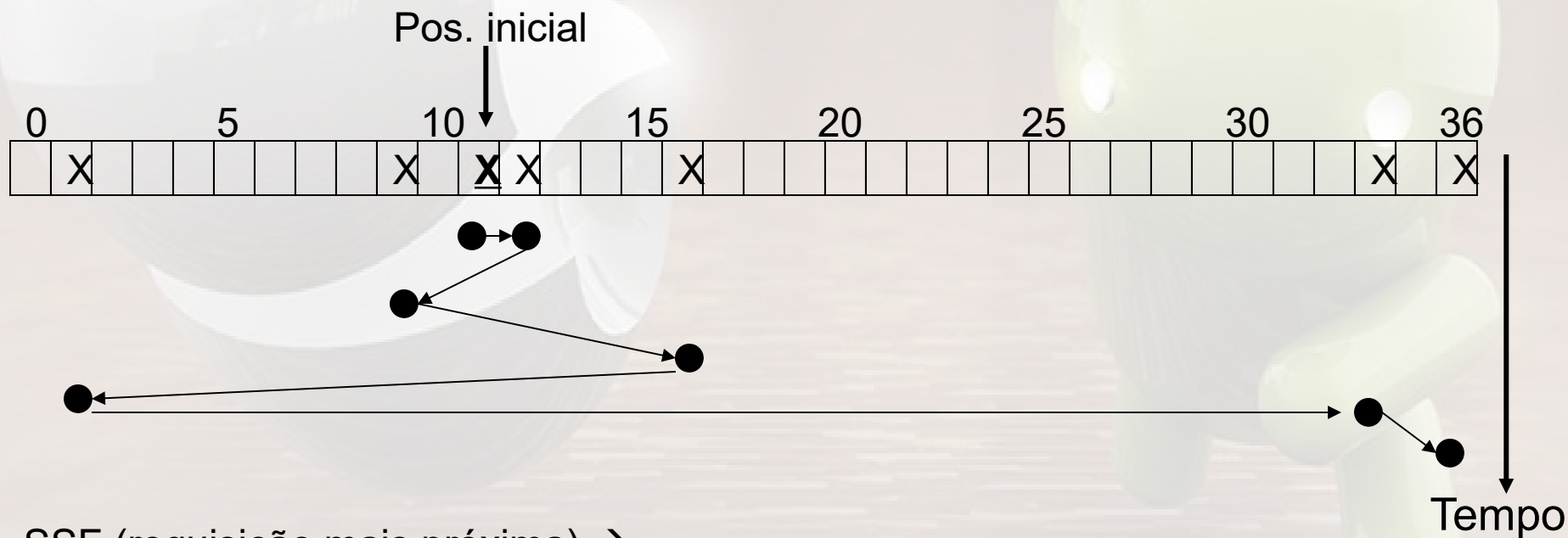
Disco com 37 cilindros;
Lendo bloco no cilindro 11;
Requisições: 1,36,16,34,9,12, nesta ordem



FCFS → atendimento: 1,36,16,34,9,12;
movimentos do braço (número de cilindros): $10, 35, 20, 18, 25, 3 = 111$;

Desempenho

Disco com 37 cilindros;
Lendo bloco no cilindro 11;
Requisições: 1,36,16,34,9,12, nesta ordem



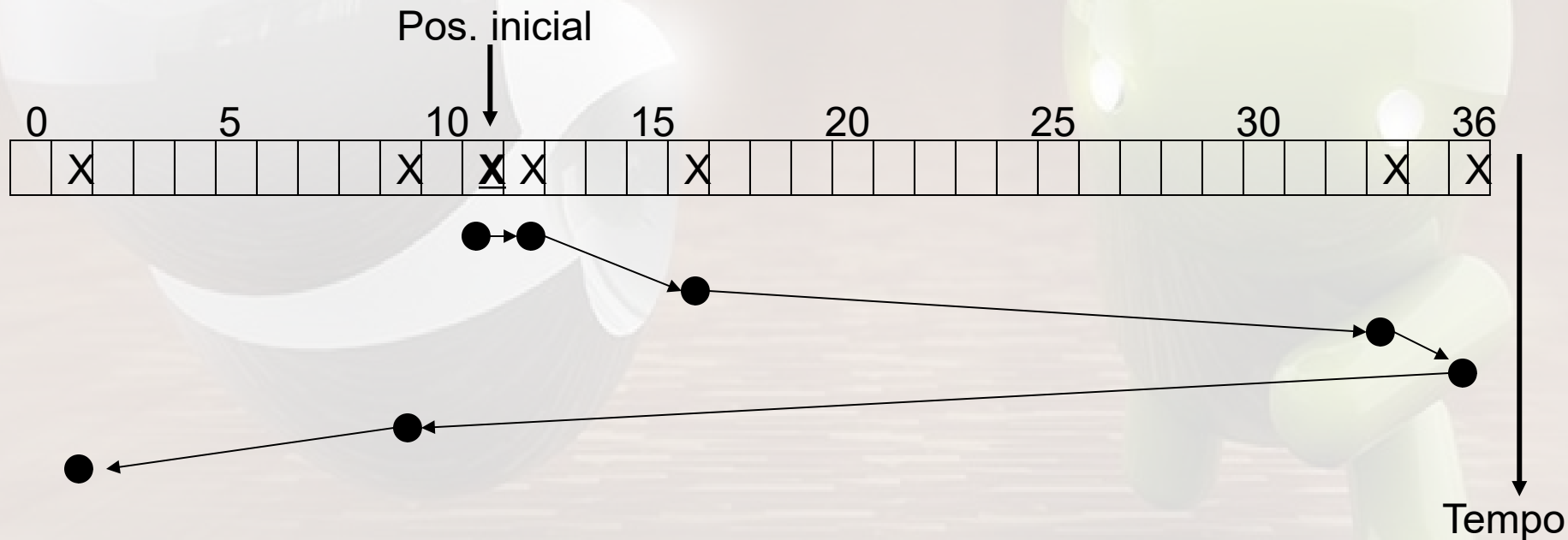
SSF (requisição mais próxima) →
atendimento: 12,9,16,1,34,36;

movimentos do braço (número de cilindros): 1,3,7,15,33,2 = 61;

Desempenho

Disco com 37 cilindros;
Lendo bloco no cilindro 11;
Requisições: 1,36,16,34,9,12, nesta ordem

Bit de direção corrente (*driver*):
Se *Up* → atende próxima requisição;
senão *Bit* = *Down*;
muda direção e atende requisição;



Elevator (requisições na mesma direção) →
atendimento: 12,16,34,36,9,1

movimentos do braço (número de cilindros): 1,4,18,2,27,8 = 60;

Referências

- Slides adaptados de Roberta Lima Gomes (UFES)
- Bibliografia
 - A. S. Tanenbaum, "**Sistemas Operacionais Modernos**", 3a. Edição, Editora Prentice-Hall, 2010.
 - Capítulo 4
 - Silberschatz A. G.; Galvin P. B.; Gagne G.; "**Fundamentos de Sistemas Operacionais**", 6a. Edição, Editora LTC, 2004.
 - Seção 11.3 inclusa
 - Deitel H. M.; Deitel P. J.; Choffnes D. R.; "**Sistemas Operacionais**", 3ª. Edição, Editora Prentice-Hall, 2005
 - Capítulo 13