



Laboratório de Pesquisa em Redes e Multimídia

A grayscale, semi-transparent image of a computer keyboard is positioned in the background, angled from the bottom left towards the top right. The keys are clearly visible but faded, serving as a backdrop for the text.

Sistemas de Arquivos

Diretórios

Gerenciamento de Espaço em Disco



Universidade Federal do Espírito Santo
Departamento de Informática

Implementação de Diretórios (1)

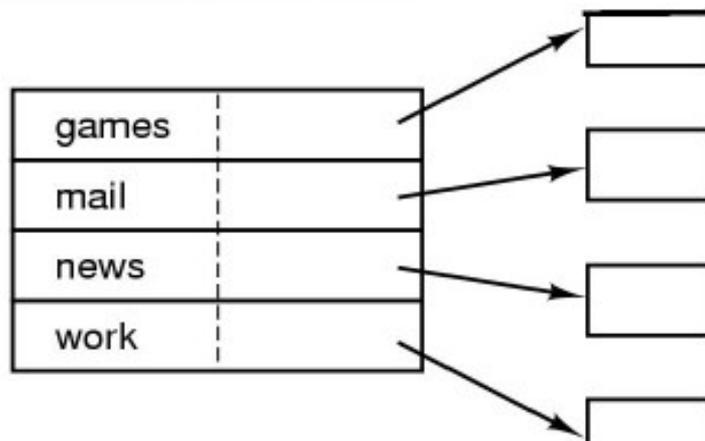
- Contém informações que permitem acessar os arquivos
 - As entradas do diretório fornecem informações para encontrar os blocos de discos
- Possui várias entradas, uma por arquivo:
 - nome
 - tipo; tamanho
 - proprietário; proteção
 - data de criação; data da última modificação
 - **lista de blocos usados**

Implementação de Diretórios (2)

games	attributes
mail	attributes
news	attributes
work	attributes

(a)

MS-DOS/WINDOWS



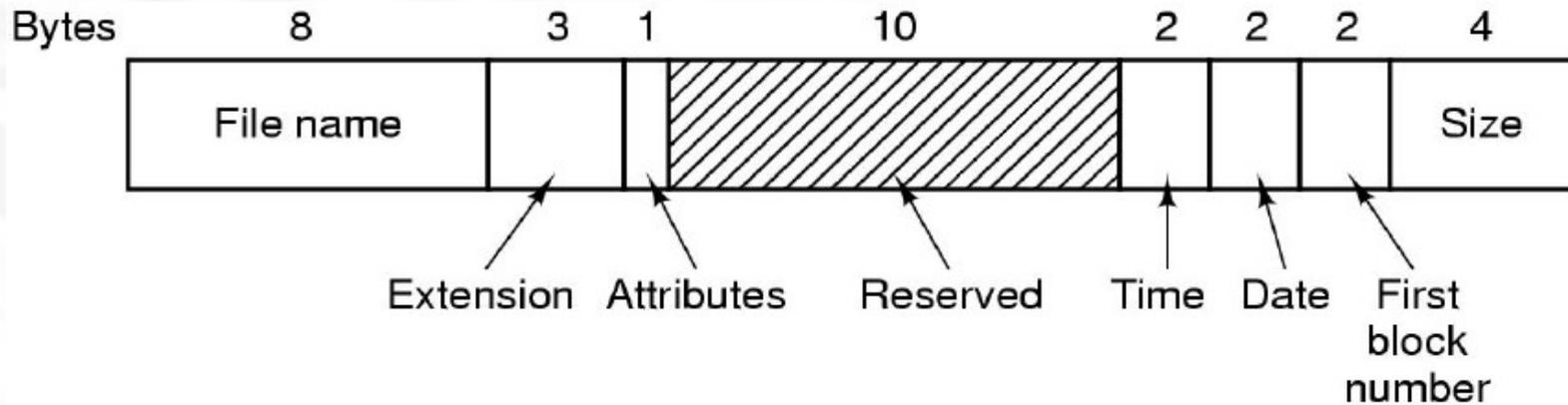
(b)

UNIX/LINUX

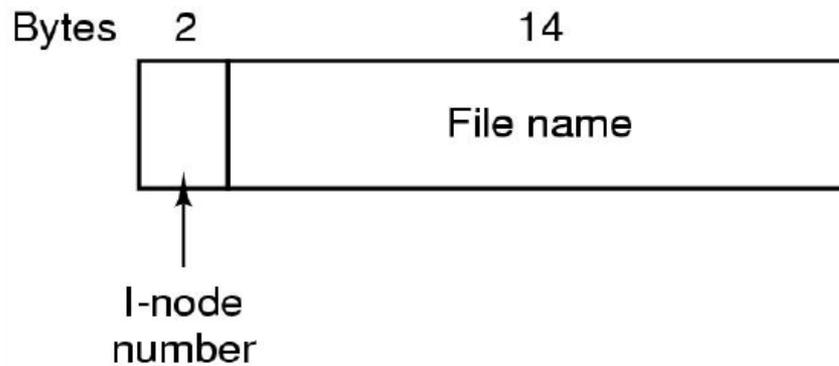
- (a) Diretório simples com
 - Entradas de dimensão fixa
 - Endereços de disco e atributos na entrada de diretório
- (b) Diretório em que cada entrada apenas refere um i-node

Implementação de Diretórios (3)

- Entrada de diretório no DOS



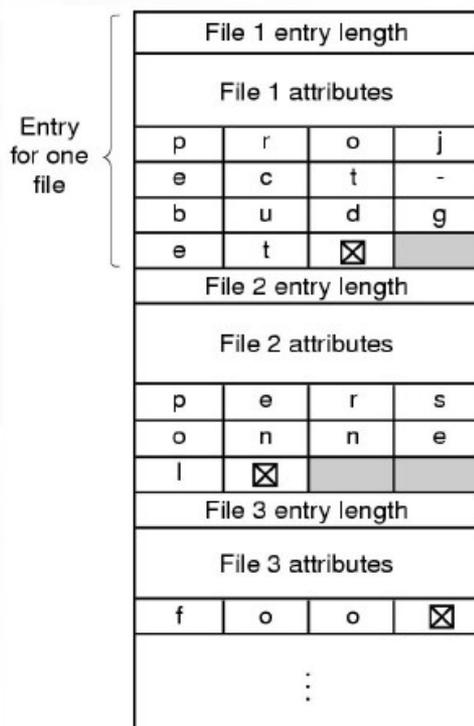
- No Unix (eg. System V)



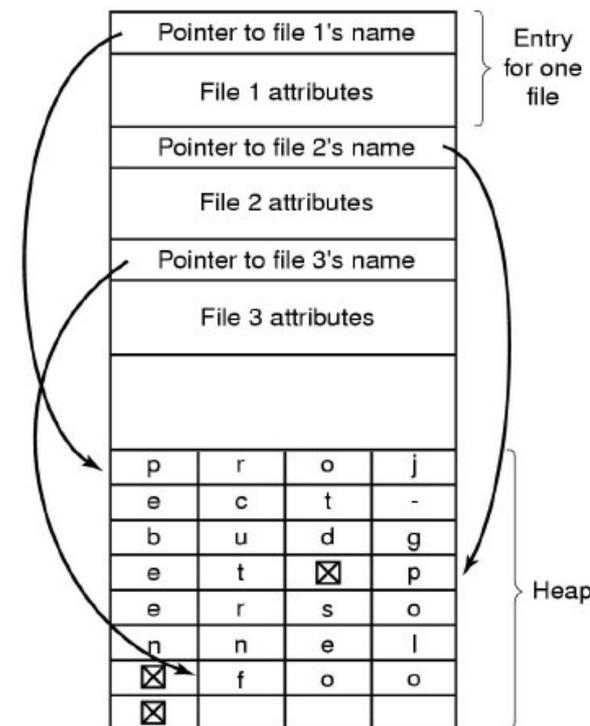
Implementação de Diretórios (4)

- Implementar “Long File Names”

In-line

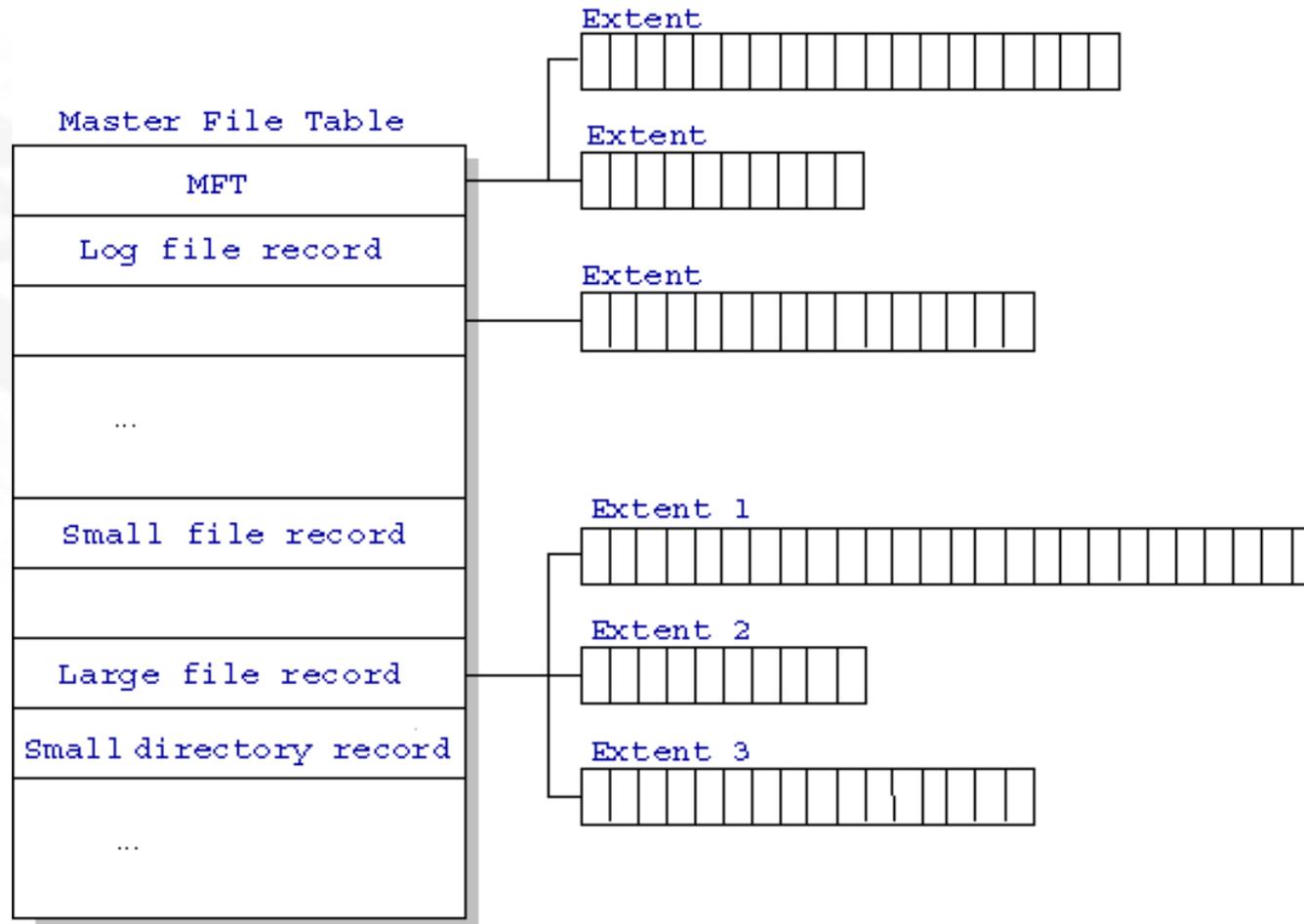


Em heap

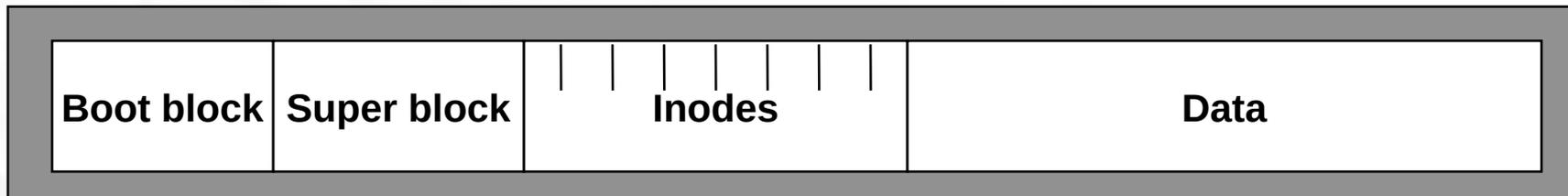


- Linux & NTFS
– 255 bytes

Implementação de Diretórios - NTFS



Implementação de Diretórios - UNIX



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

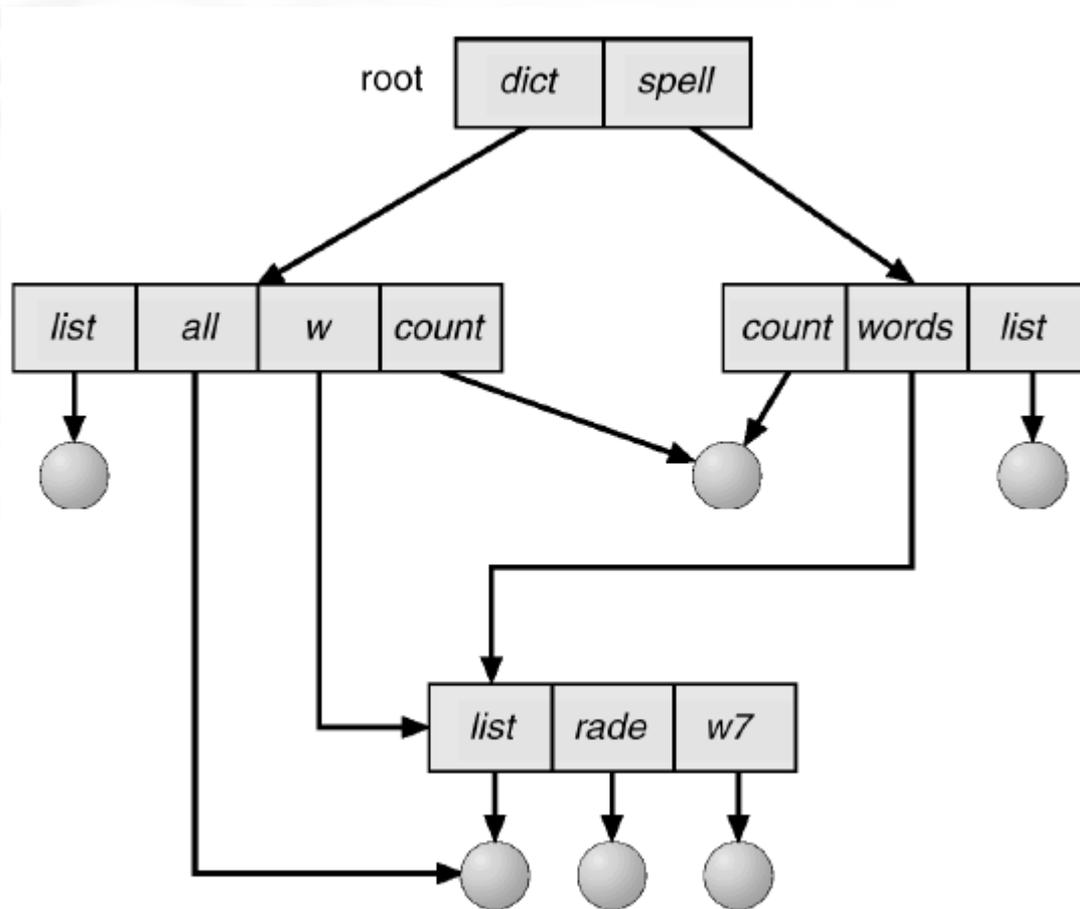
Block 406
is /usr/ast
directory

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

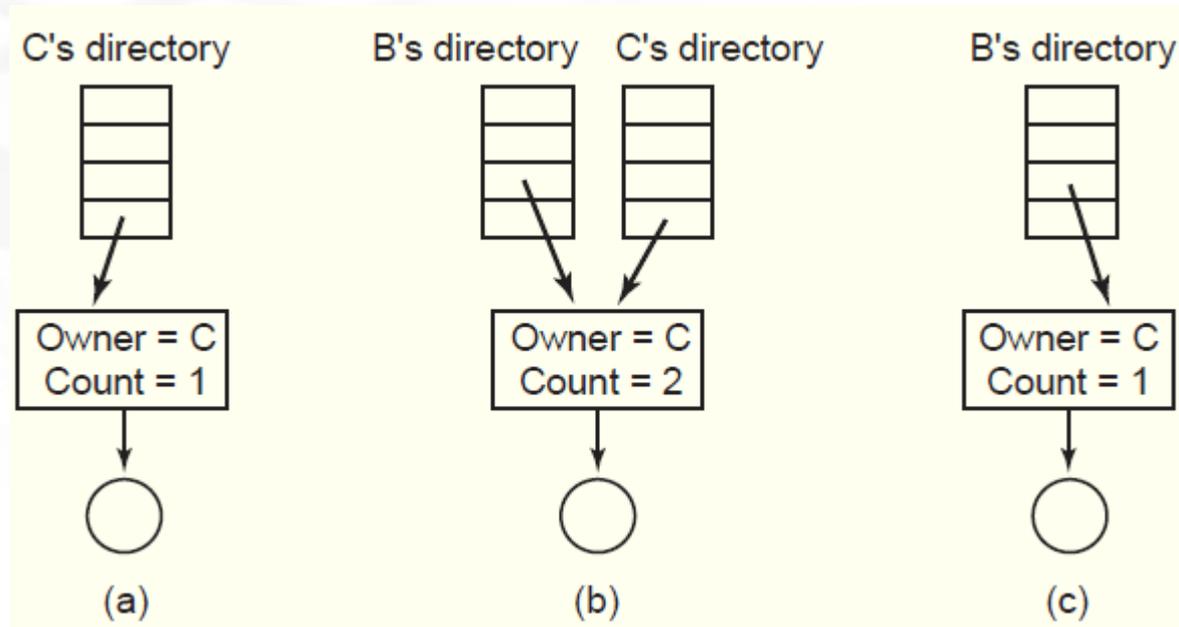


Arquivos Compartilhados (2)

- 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”

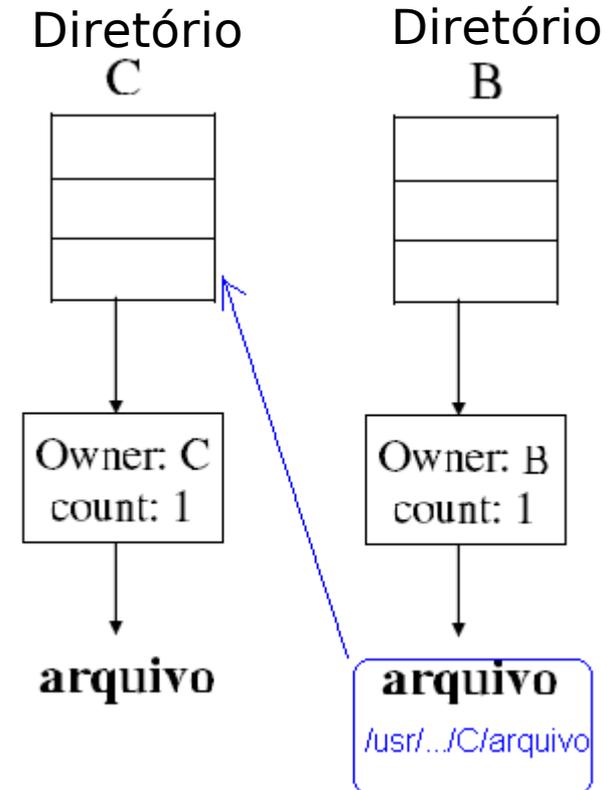
Arquivos Compartilhados (3)

- 1a. Solução



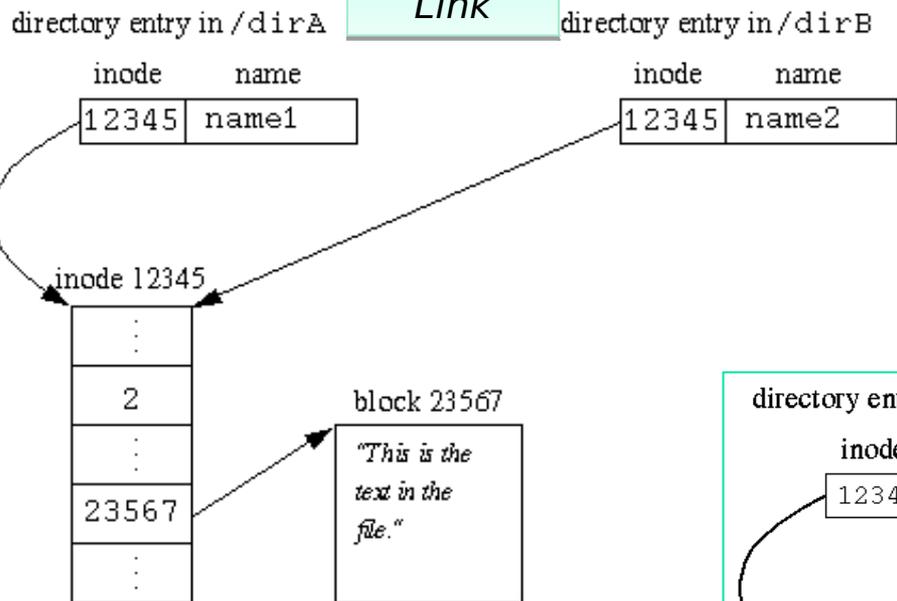
Arquivos Compartilhados (3)

- 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

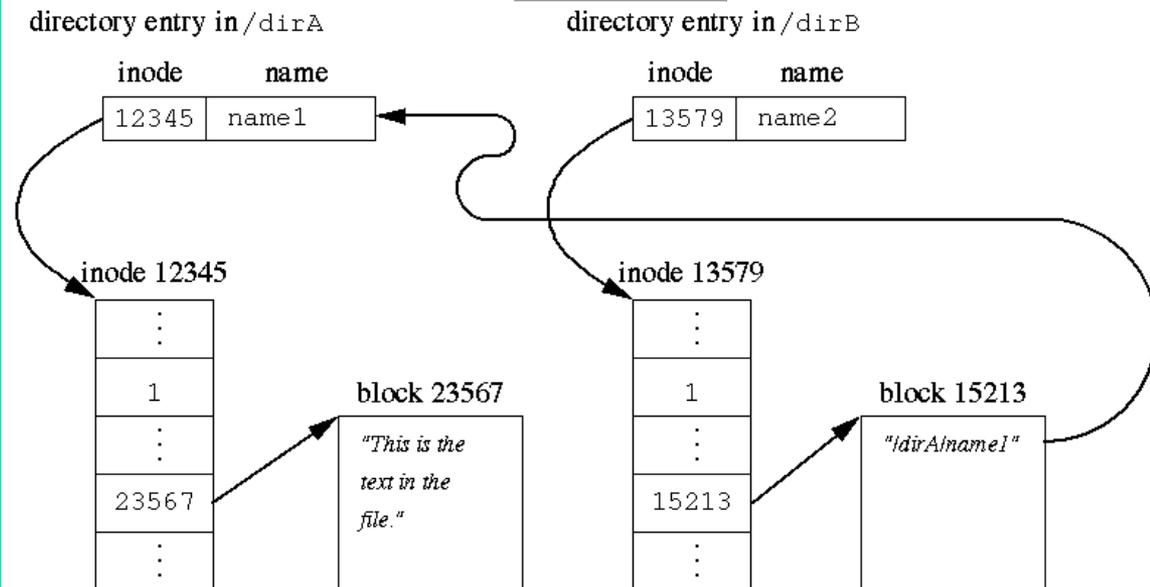


Arquivos Compartilhados (4)

Hard Link



Soft Link



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");
...
```

```
ln /dirA/name1 /dirB/name2
```

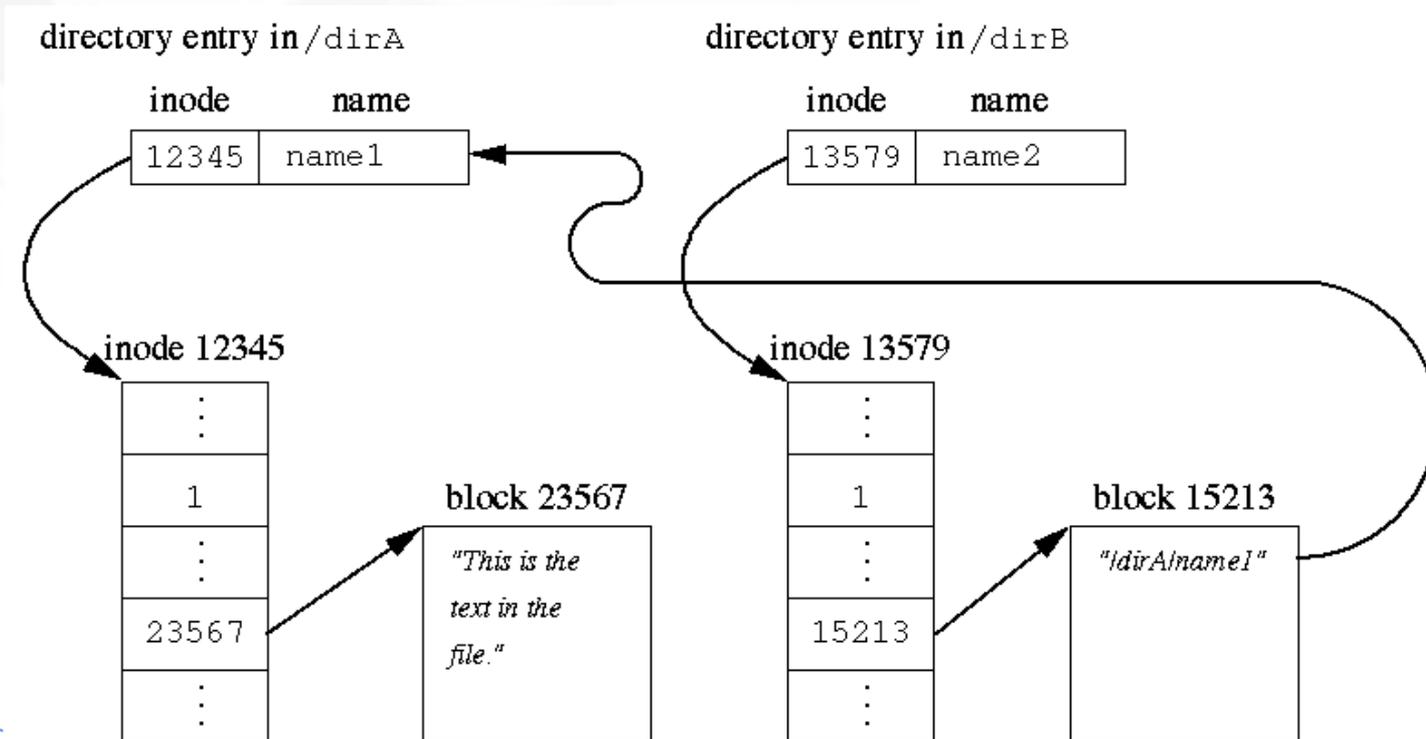
Criando Soft Links (SVC)

- 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

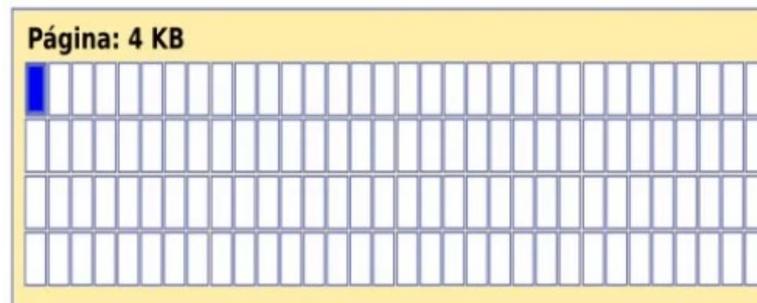


Gerenciamento de Espaço em Disco (HDs) (1)

- 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

Tempo médio de seek + latência rotacional + tempo de leitura

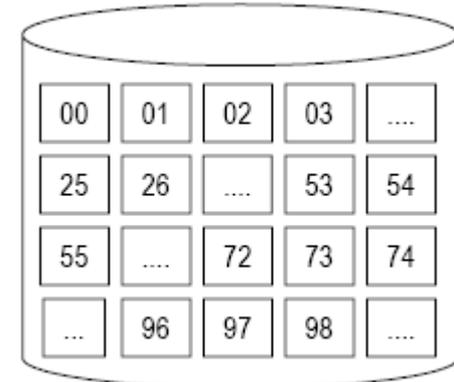
- E no SSD?



Bloco: 512 KB I/O

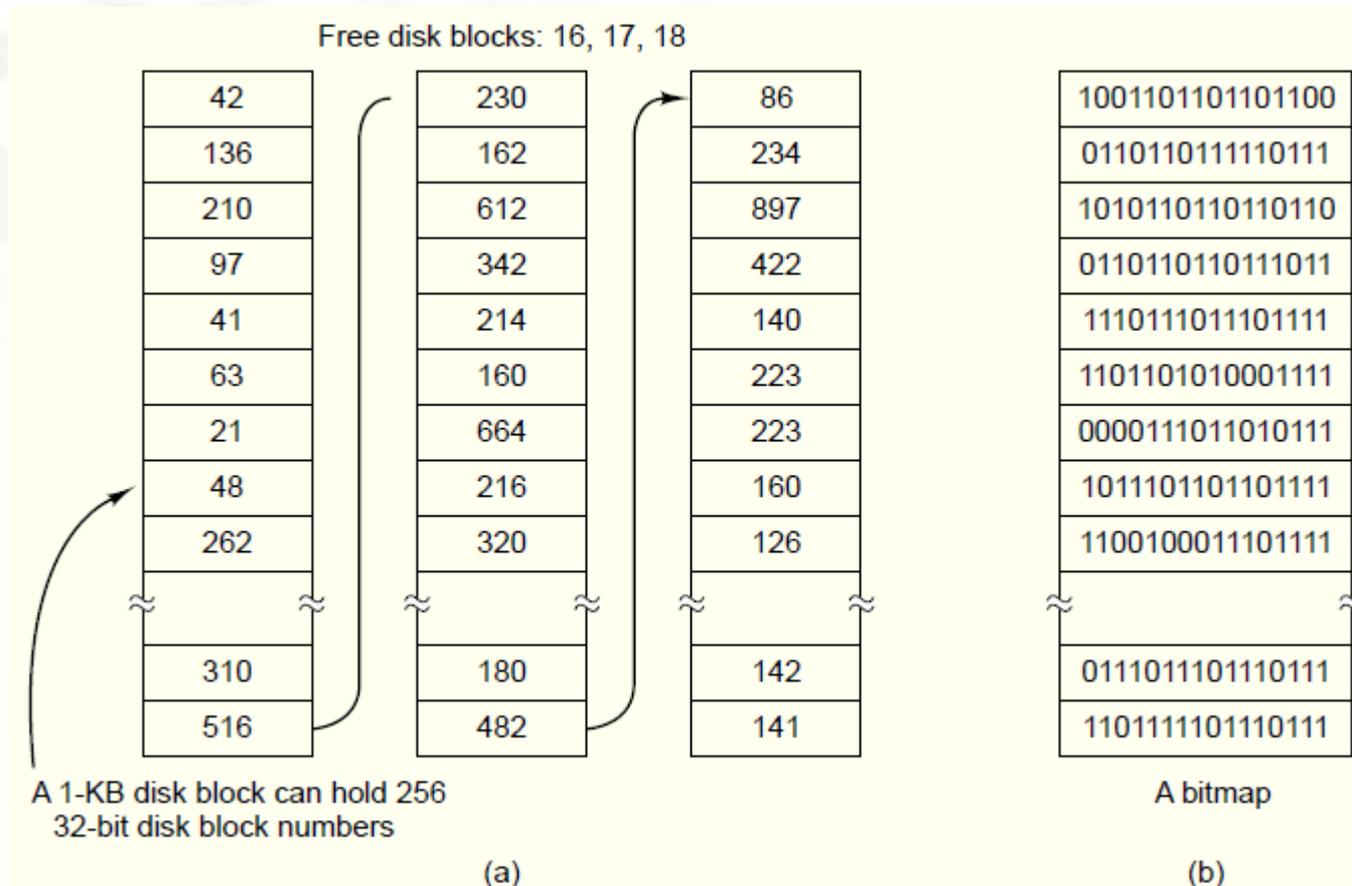
Gerenciamento de Espaço em Disco (2)

- 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



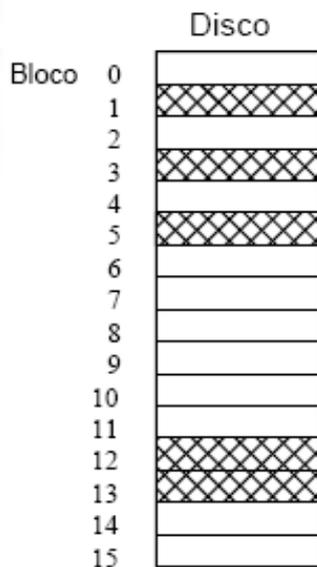
Gerenciamento de Espaço em Disco (2)

■ Gerenciamento do Espaço Livre (cont.)



Gerenciamento de Espaço em Disco (3)

- Gerenciamento do Espaço Livre (cont.)
 - Mapa de bits
 - Forma simples de gerenciar o espaço em disco
 - Um disco c/n blocos requer um bitmap de n bits



Mapa de Bits.

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

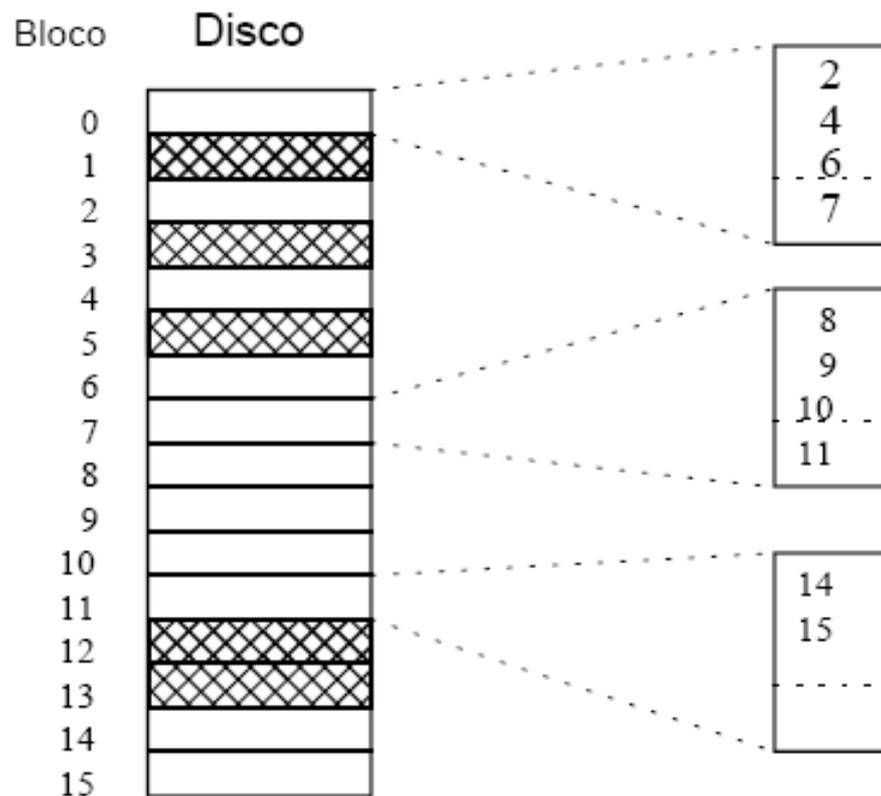
$$\text{tamanho_bit_map} = \frac{\text{Capacidade_disco}(\text{bytes})}{8 \times \text{tamanho_bloco}(\text{bytes})}$$

Gerenciamento de Espaço em Disco (4)

■ Gerenciamento do Espaço Livre (cont.)

■ 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 (4)

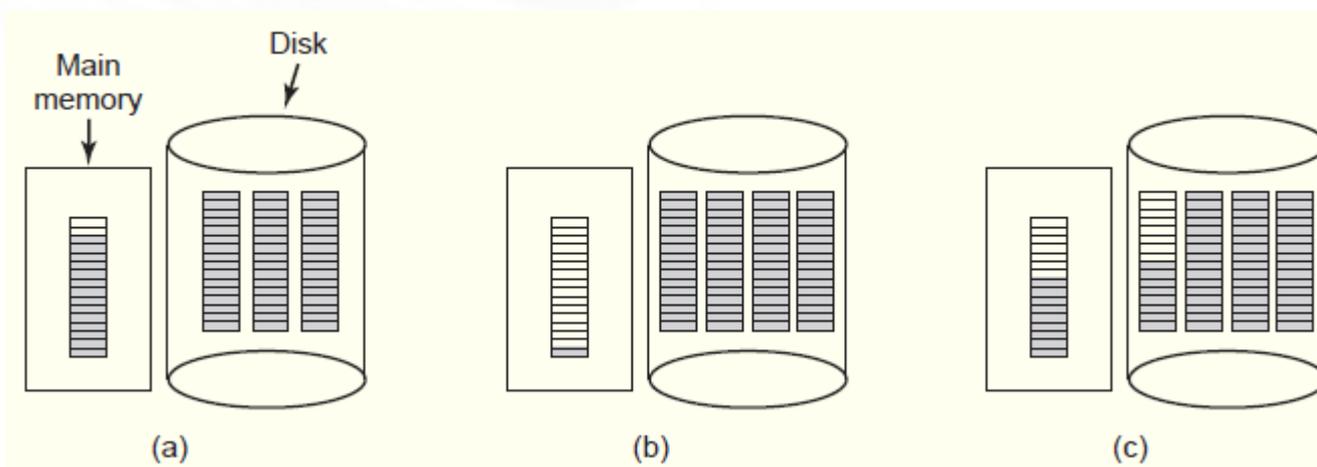
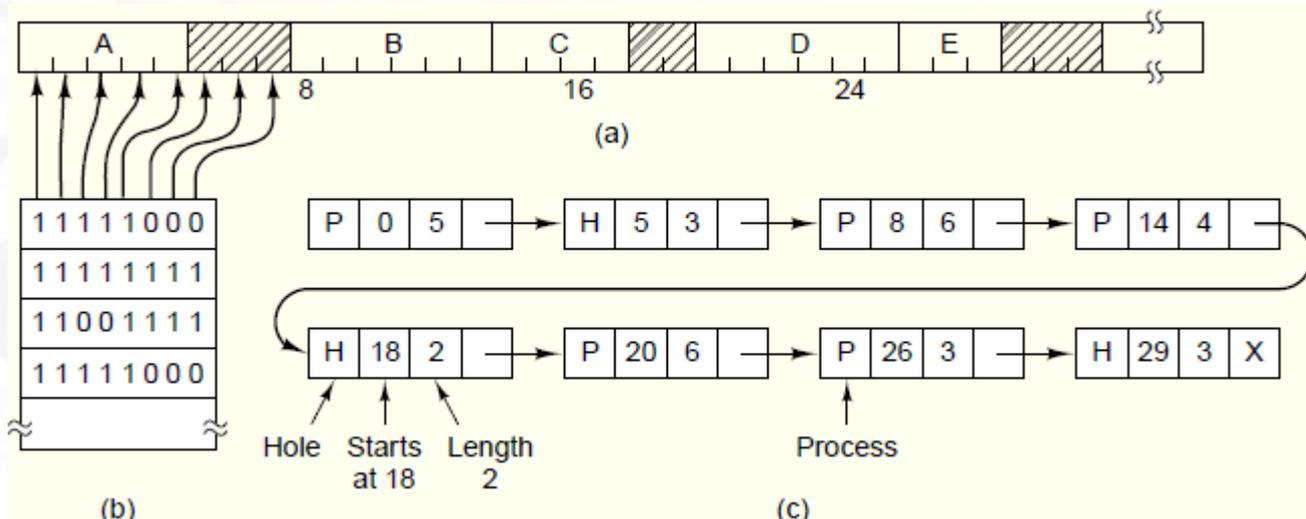
- Gerenciamento do Espaço Livre (cont.)
 - Lista de blocos livres
 - Quantos blocos precisamos para armazenar a lista de blocos livres?
 - Exemplo:
 - Disco de 40 M com Blocos de 1k
 - 2 bytes para identificar bloco livre

Cada bloco pode conter 512 entradas

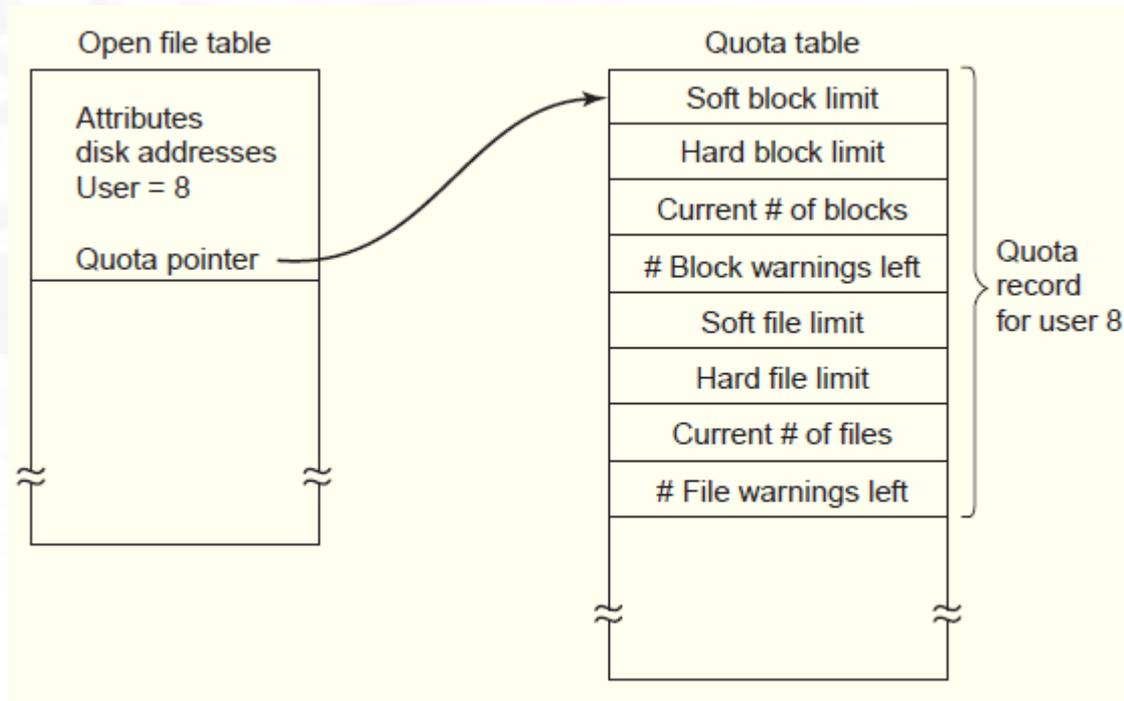
Disco de 40 M => dividido em 40 k blocos

$$\text{nº de blocos} = \frac{40k}{512} = 80 \text{ blocos}$$

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)

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)

(a) consistente
 (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)

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)

(b) bloco faltando
 (d) duplicação nos
 dados