Laboratório de Pesquisa em Redes e Multimídia

Sistemas de Arquivos

Universidade Federal do Espírito Santo
Departamento de Informática

Funções de um SO

- Gerência de processos
- Gerência de memória
- **Gerência de Arquivos**
- Gerência de I/O
- Sistema de Proteção

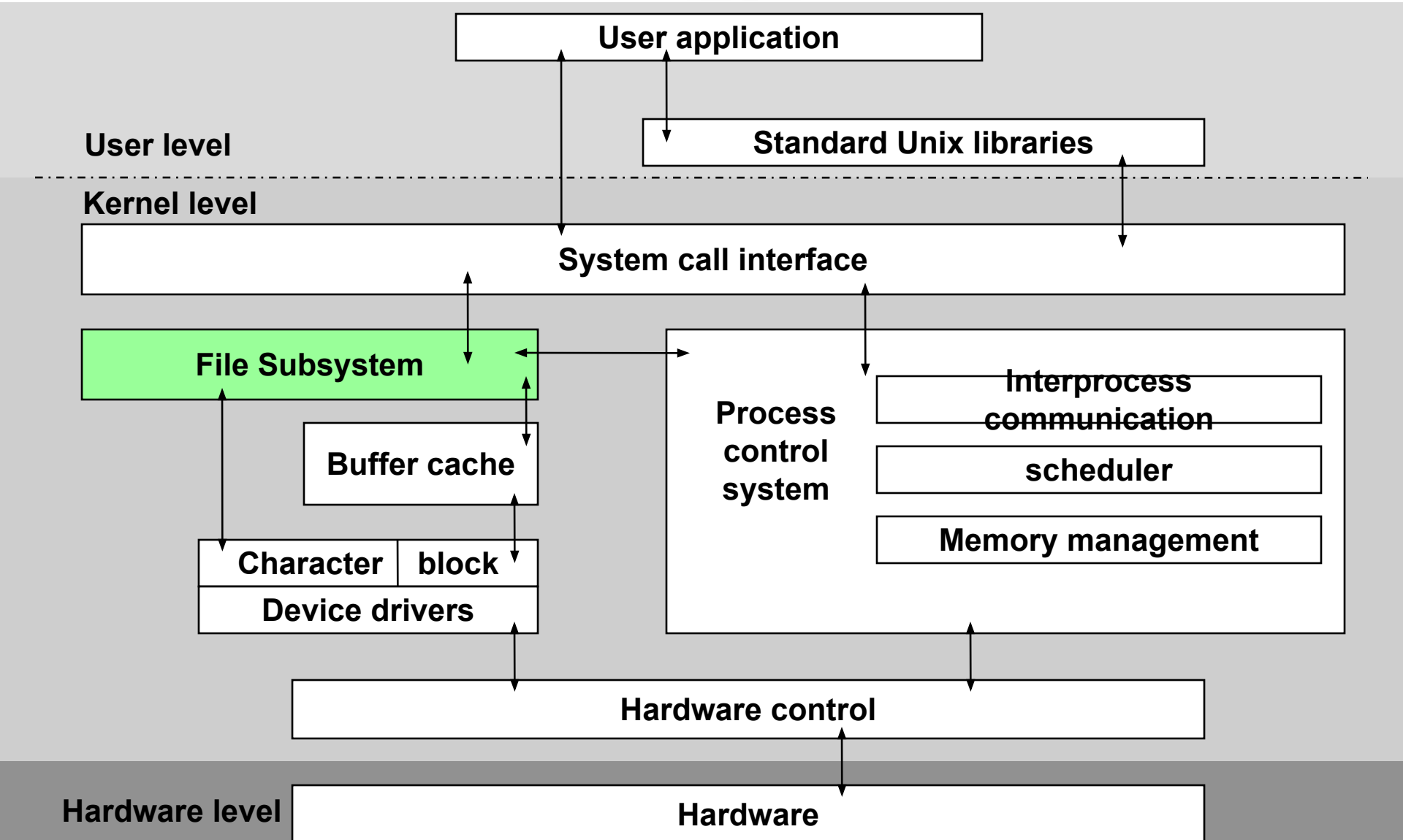
Necessidade de Armazenamento

- Grandes quantidades de informação têm de ser armazenadas
- Informação armazenada tem de sobreviver ao fim do processo que a utiliza
- Múltiplos processos devem poder acessar a informação de um modo concorrente
- ARQUIVO
  - Abstração criada pelo S.O. para gerenciar e representar os dados

Gerência de Arquivos

- Oferece a abstração de arquivos (e diretórios)
- Atividades suportadas
  - Primitivas para manipulação (chamadas de sistema para manipulação de arquivos)
    - criar, deletar
    - abrir, fechar
    - ler, escrever
    - posicionar
  - Mapeamento para memória secundária

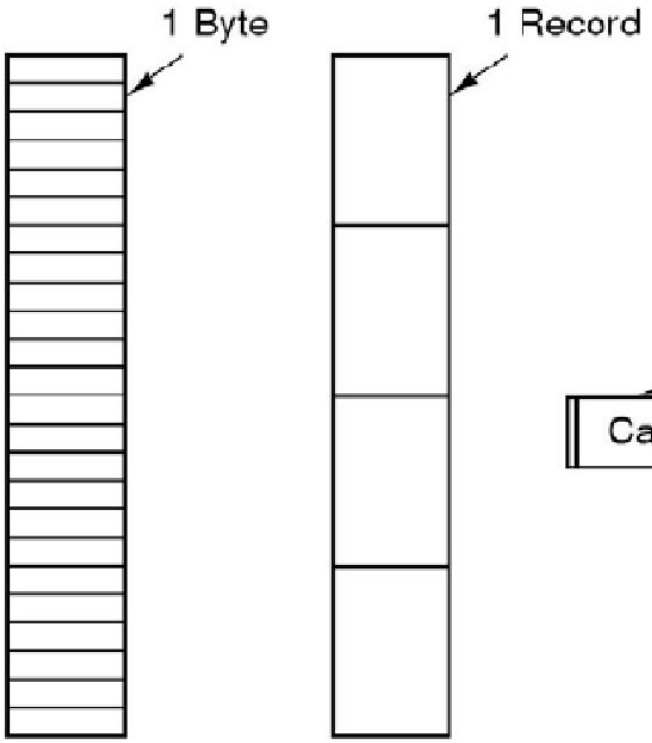
Estrutura Interna do Kernel UNIX

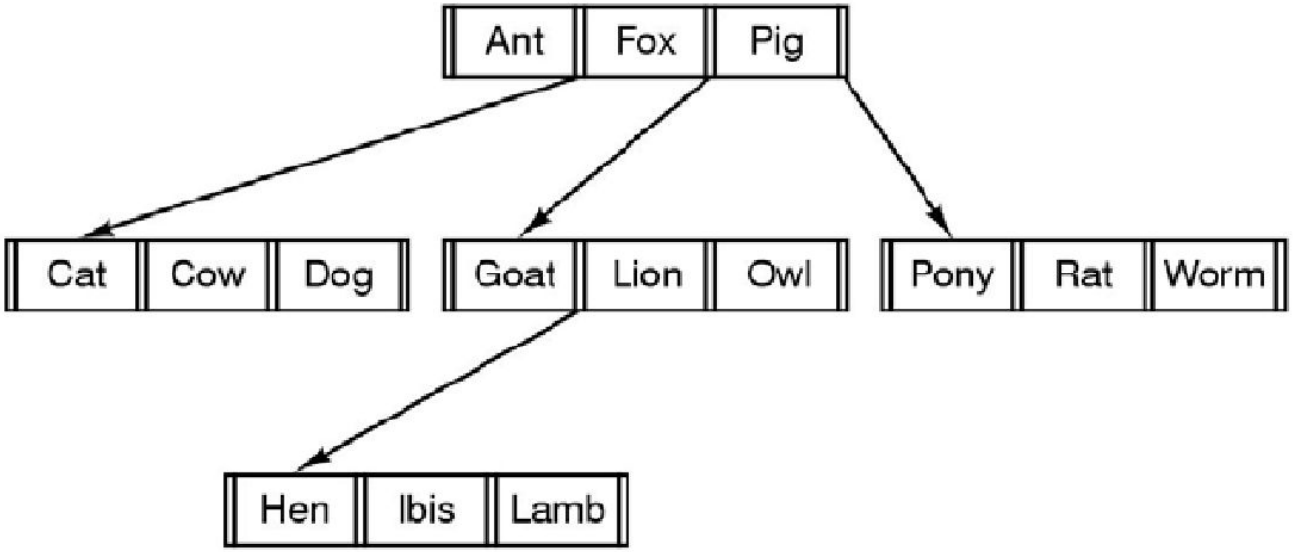


Sistema de Arquivos

- O que é?
  - Um conjunto de arquivos, diretórios, descritores e estruturas de dados auxiliares gerenciados pelo sub sistema de gerência de arquivos
  - Permitem estruturar o armazenamento e a recuperação de dados persistentes em um ou mais dispositivos de memória secundária (discos ou bandas magnéticas)
- Arquivo
  - Um conjunto de dados persistentes, geralmente relacionados, identificado por um nome
  - É composto por:
    - **Nome**: identifica o arquivo perante o utilizador
    - **Descritor de arquvo**: estrutura de dados em memória secundária com informação sobre o arquivo (dimensão, datas de criação, modificação e acesso, dono, autorizações de acesso)
    - Informação: dados guardados em memória secundária

Estrutura Interna de Arquivos (1)





Seqüência não-estruturada de bytes

Seqüência de Registros

Árvore de Registros

Estrutura Interna de Arquivos (2)

- Seqüência não-estruturada de bytes
  - Forma mais simples de organização de arquivos
  - Sistema de arquivos não impõe nenhuma estrutura lógica para os dados, a aplicação deve definir toda a organização
  - Vantagem: flexibilidade para criar estruturas de dados, porém todo o controle de dados é de responsabilidade da aplicação
  - Estratégia adotada tanto pelo UNIX quanto pelo Windows

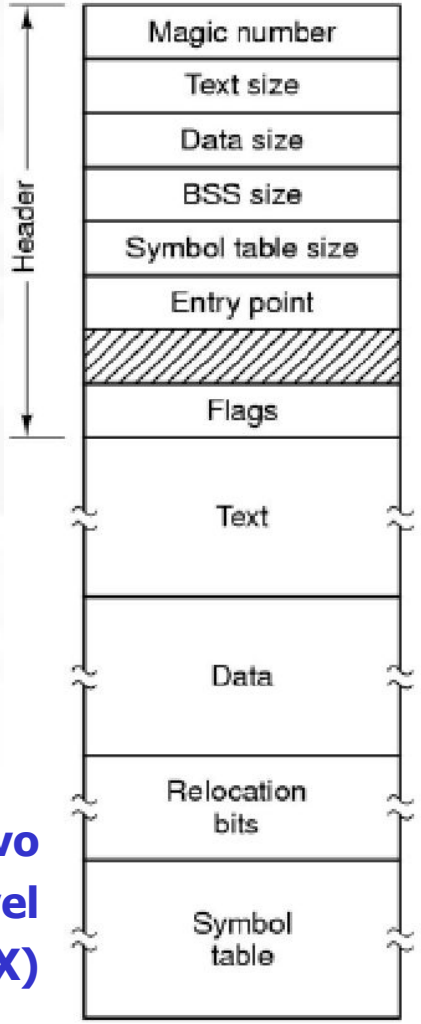
Estrutura Interna de Arquivos (3)

- Seqüência de Registros
  - Em geral, registros de tamanho fixo
  - Operação de leitura retorna um registro
  - Operação de escrita sobrepõe/anexa um registro
- Árvore de Registros
  - Cada registro é associado a uma chave
  - Árvore ordenada pela chave
  - Computadores de grande porte / aplicações que fazem muita leitura aleatória

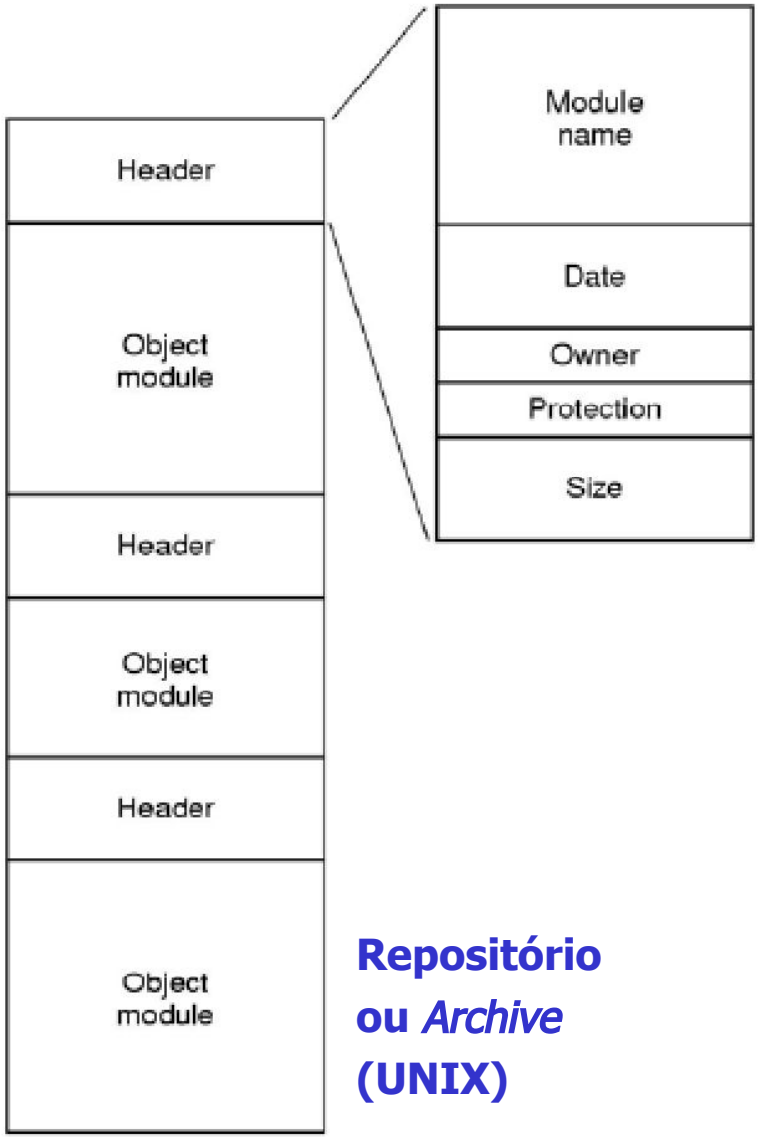
Tipos de Arquivos (1)

- **Arquivos Regulares**
  - Arquivos ASCII
  - Binários
    - Apresentam uma estrutura interna conhecida pelo S.O.
- **Diretórios**
  - Arquivos do sistema
  - Mantêm a estrutura do Sistemas de Arquivos

Arquivos Binários no Unix



Arquivo Executável (UNIX)



Repositório ou *Archive* (UNIX)

Operações sobre Arquivos

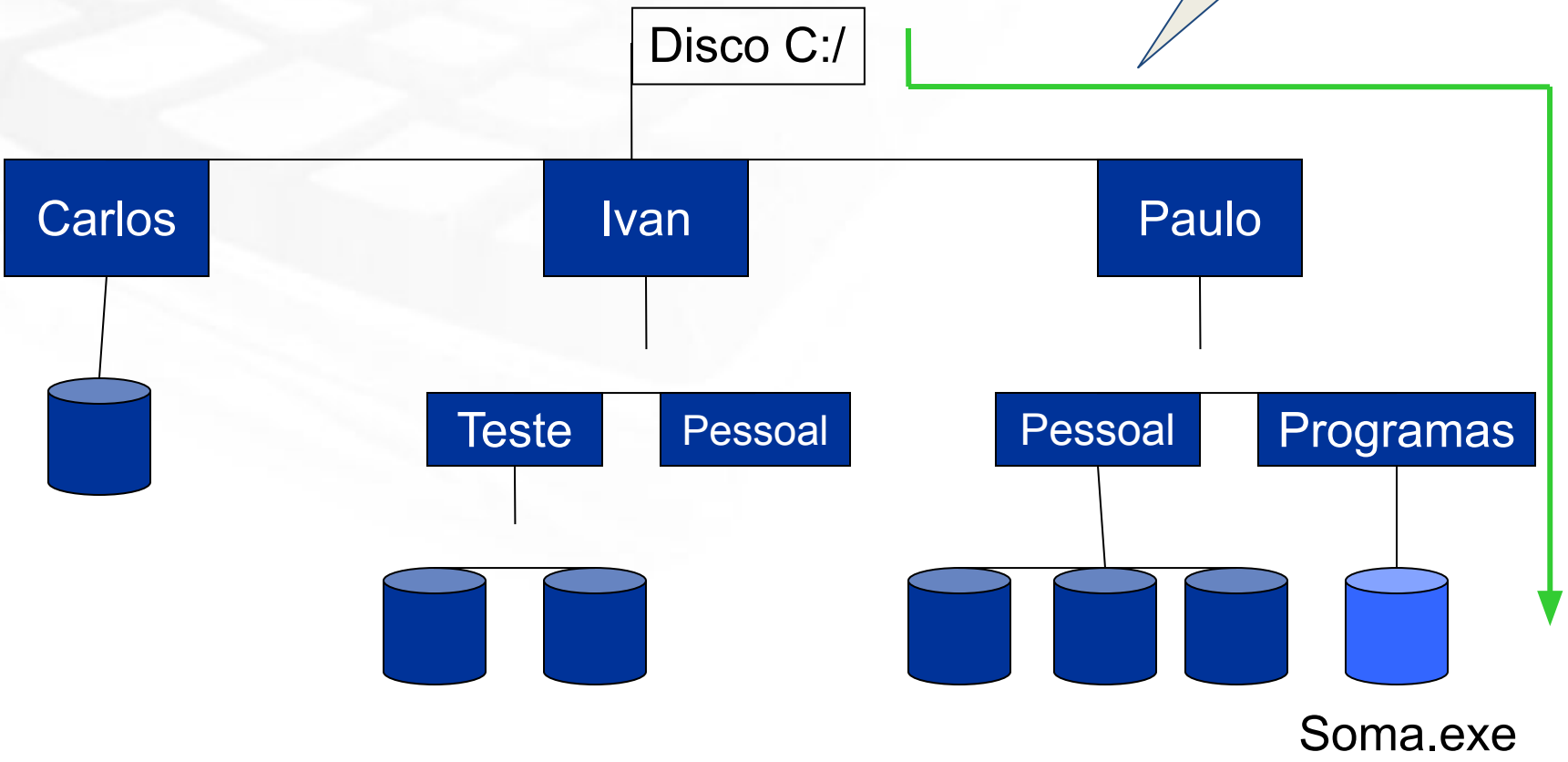
- Dependem do tipo
  - create
  - delete
  - open
  - close
  - read
  - write
  - append
  - seek
  - get attributes
  - set attributes
  - rename

Diretórios

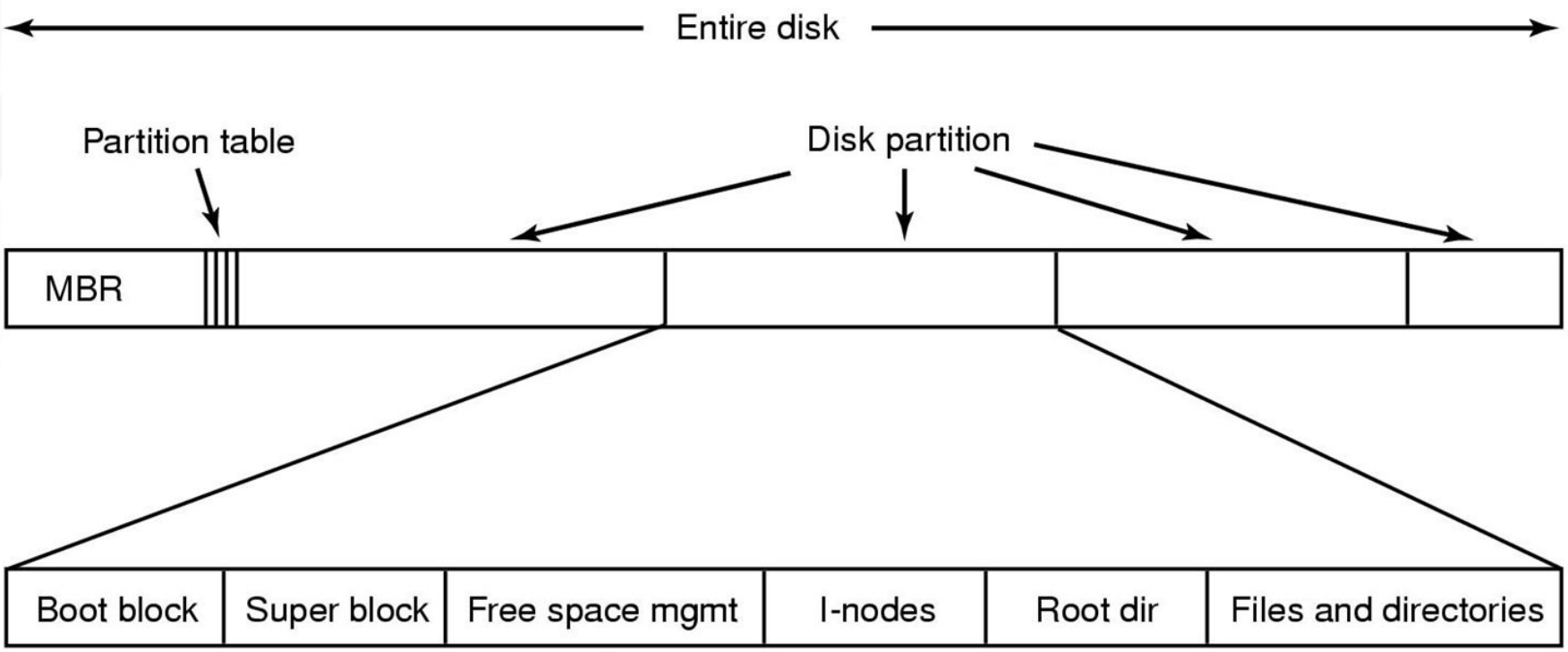
- ATUALMENTE!! Estrutura de diretórios Hierárquicos
  - Adotado pela maioria dos sistemas operacionais
  - Logicamente melhor organizado
  - É possível criar quantos diretórios quiser
  - Um diretório pode conter arquivos e outros diretórios (chamados subdiretórios)
  - Cada arquivo possui um **path** único que descreve todos os diretórios da raiz (MFD – Master File Directory) até o diretório onde o arquivo está ligado
  - Na maioria dos S.O.s os diretórios são tratados como arquivos tendo atributos e identificação

Diretórios (4)

- Estrutura de diretórios Hierárquicos (cont.)



Esquema do Sistema de Arquivos (1)



Esquema do Sistema de Arquivos (2)

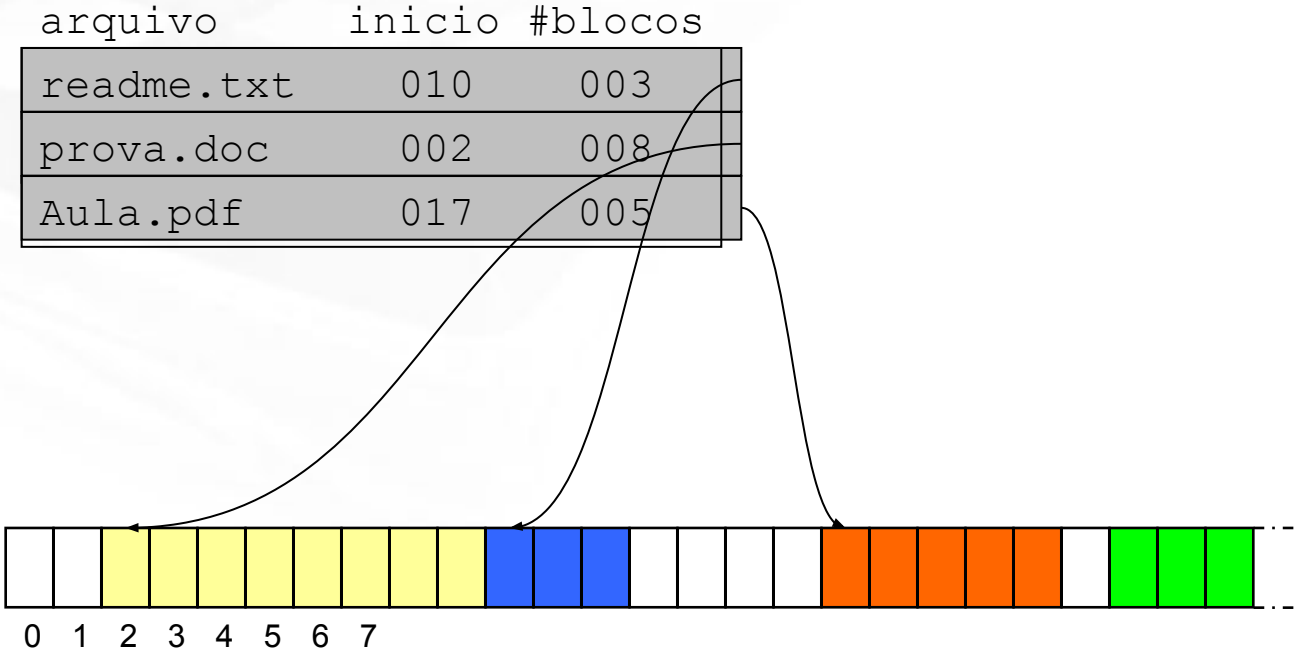
- A maioria dos discos é dividida em uma ou mais partições com Sistemas de arquivos independentes para cada partição
- O setor 0 do disco é chamado de *Master Boot Record* (MBR)
- Na inicialização do sistema, a BIOS lê e executa o MBR
  - O programa do MBR localiza a partição ativa, lê seu primeiro bloco, chamado de **bloco de boot**
  - O programa no bloco de boot carrega o S.O. contido na partição
  - Havendo um boot manager, como o GRUB, no MBR se encontra uma parte "inicial" dele
- O esquema da partição varia de um S.O. para outro, mas é comum:
  - A definição de um **SuperBloco**: contém os principais parâmetros do sistema de arquivos (tipo, no. de blocos, etc.)
  - As informações sobre os blocos livres

Implementação do Sistema de Arquivos (1)

- Alocação Contígua
  - Consiste em armazenar um arquivo em blocos sequencialmente dispostos
  - O sistema localiza um arquivo por meio do endereço do primeiro bloco e da sua extensão em blocos
  - O acesso é bastante simples
  - Seu principal problema é a alocação de novos arquivos nos espaços livres
    - Para armazenar um arquivo que ocupa **n** blocos, é necessário uma cadeia com **n** blocos dispostos seqüencialmente no disco
  - Além disso, como determinar o espaço necessário a um arquivo que possa se estender depois da sua criação?
    - Pré-alocação (fragmentação interna)

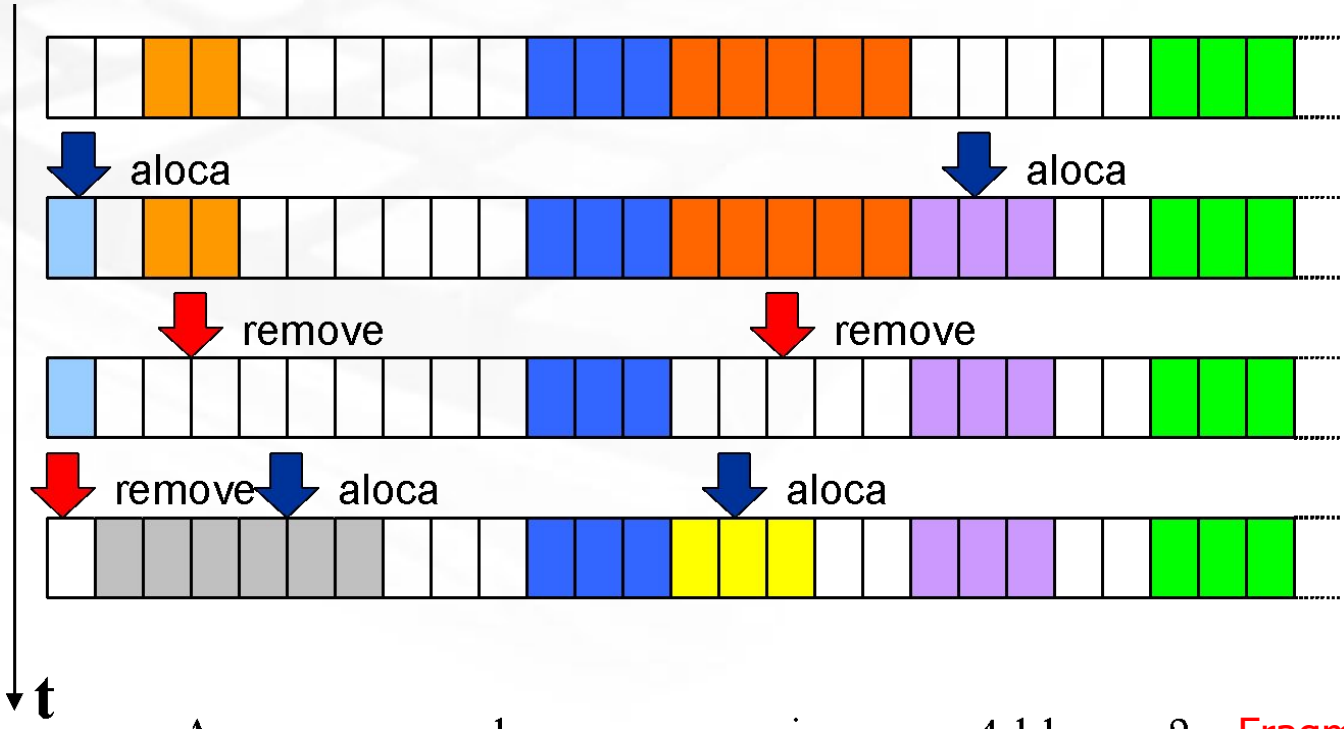
Implementação do Sistema de Arquivos (2)

- Alocação Contígua (cont.)



Implementação do Sistema de Arquivos (3)

- Alocação Contígua (cont.)



Agora, como alocar um arquivo com 4 blocos ?

Fragmentação Externa !

E se o arquivo fosse dividido em Blocos lógicos?

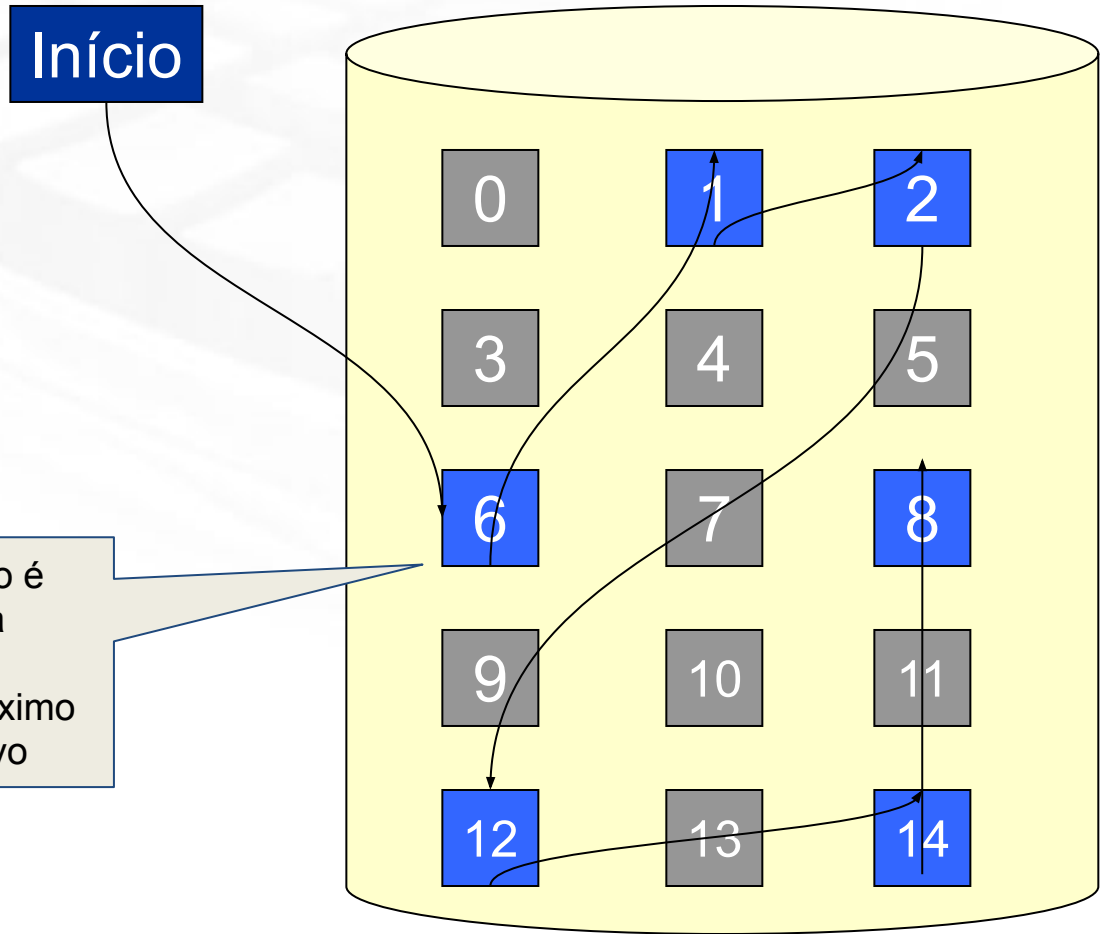
Implementação do Sistema de Arquivos (4)

- Alocação por Lista Encadeada
  - O arquivo é organizado como um conjunto de blocos ligados no disco
  - Cada bloco deve possuir um ponteiro para o bloco seguinte
  - Aumenta o tempo de acesso ao arquivo, pois o disco deve deslocar-se diversas vezes para acessar todos os blocos
    - É necessário que o disco seja desfragmentado periodicamente
  - Esta alocação só permite acesso sequencial
  - Compromete-se parte do espaço nos blocos com armazenamento de ponteiros

Implementação do Sistema de Arquivos (5)

- Alocação por Lista Encadeada (cont.)

Solução pouco eficiente!
Por exemplo, para se adicionar uma informação no final do arquivo, é necessário fazer vários acessos a disco



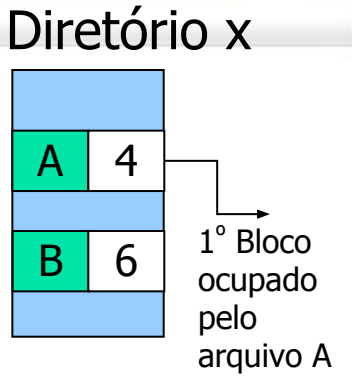
O final do bloco é reservado para armazenar o número do próximo bloco do arquivo

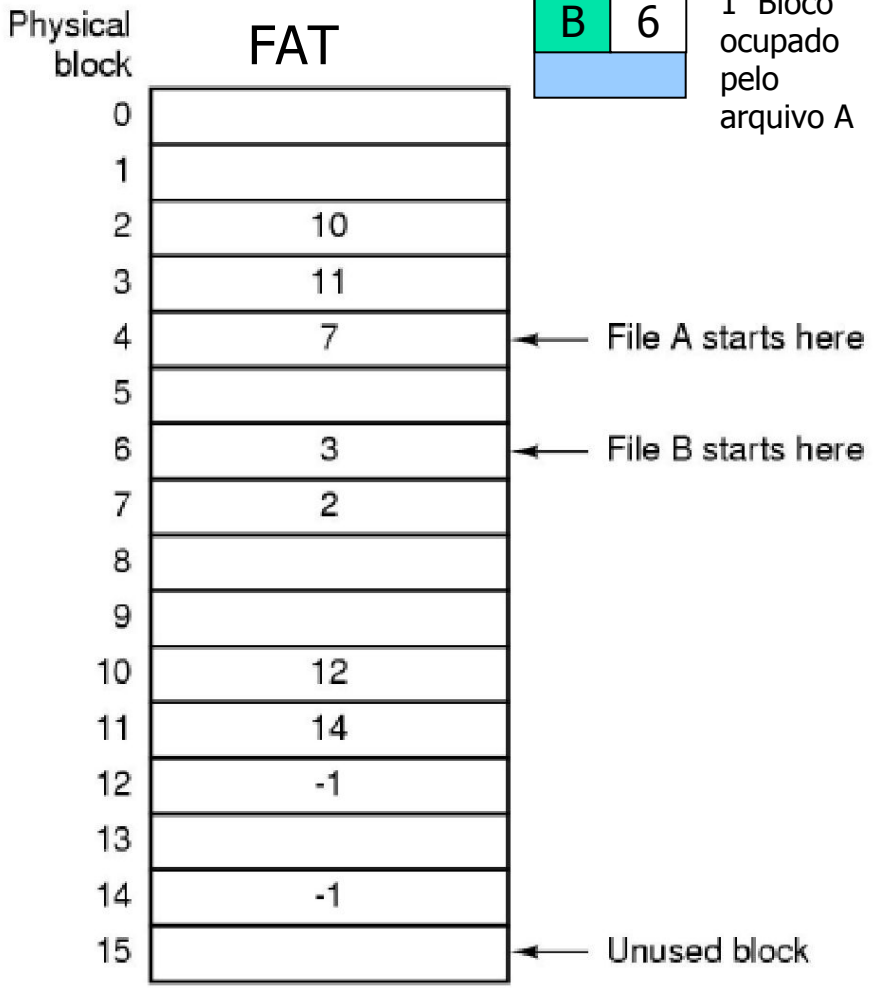
Implementação do Sistema de Arquivos (6)

- Alocação por **Lista Encadeada usando Tabela na Memória**
  - Mantém os ponteiros de todos os blocos de arquivos em uma única estrutura denominada **Tabela de Alocação de Arquivos**
    - **FAT** (File Allocation Table)
  - Vantagens:
    - Permitir o acesso direto aos blocos
    - Não mantém informações de controle dentro dos blocos de dados
  - FAT : Esquema usado pelo MS-DOS (FAT-16), Win95,Win98, Windows Millennium Edition (FAT-32) ... discos externos

Implementação do Sistema de Arquivos (7)

- Alocação por Lista Encadeada usando Tabela na Memória (cont.)

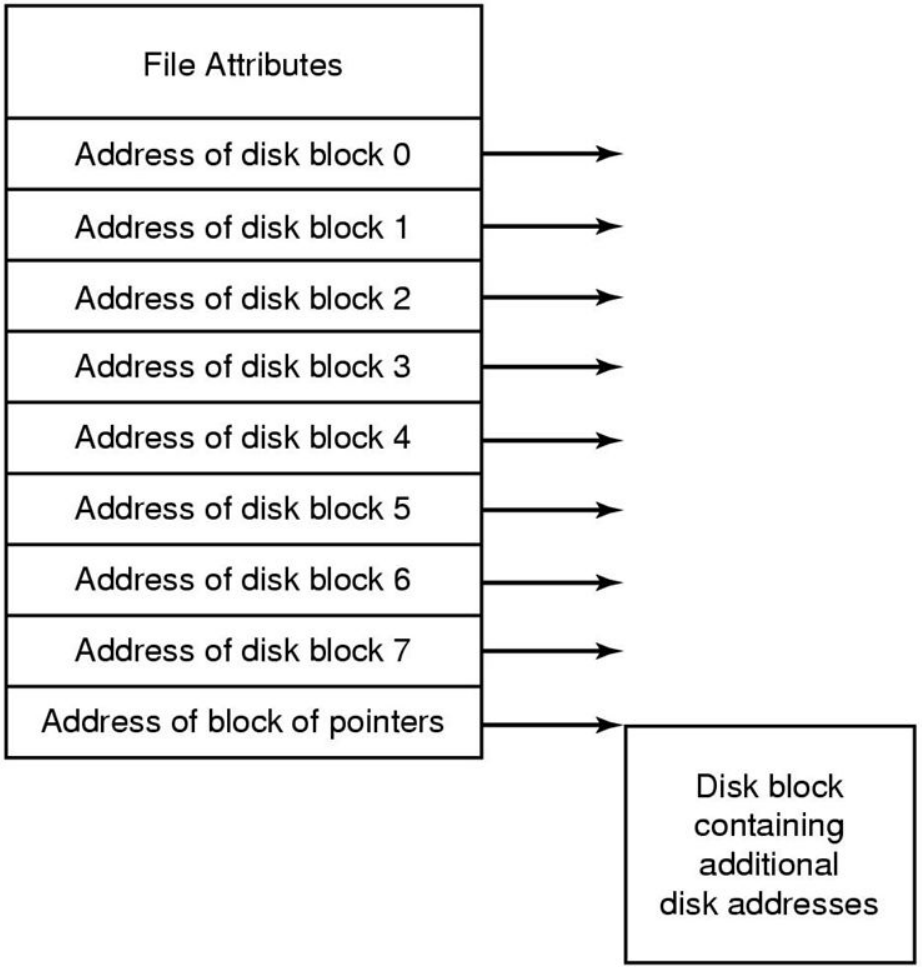




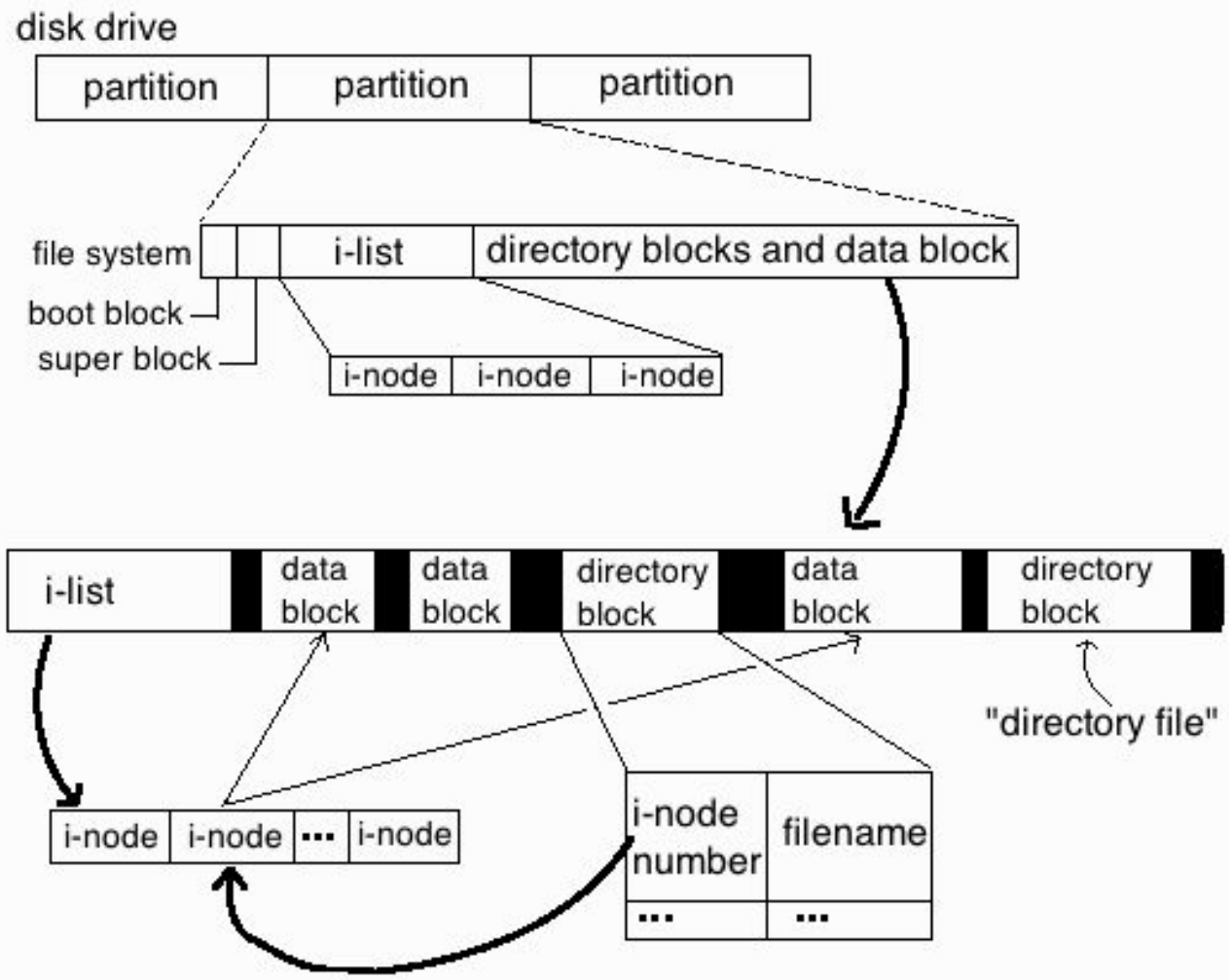
- Desvantagem
  - A tabela deve estar na memória o tempo todo
  - Disco de 20 G, blocos de 1k?
  - Mas é possível paginar a FAT!
    - Com isso "partes" dela são carregadas em memória

Implementação do Sistema de Arquivos (8)

- i-nodes
  - **Cada arquivo está associado a um único** i-node no disco
  - O i-node só precisa estar na memória quando o arquivo correspondente estiver aberto
  - Ocupa menos espaço que a FAT
    - Tamanho da FAT cresce linearmente com o tamanho do disco
    - I-nodes requerem um espaço proporcional à quantidade máxima de arquivos abertos
  - Usados por sistemas baseados no UNIX



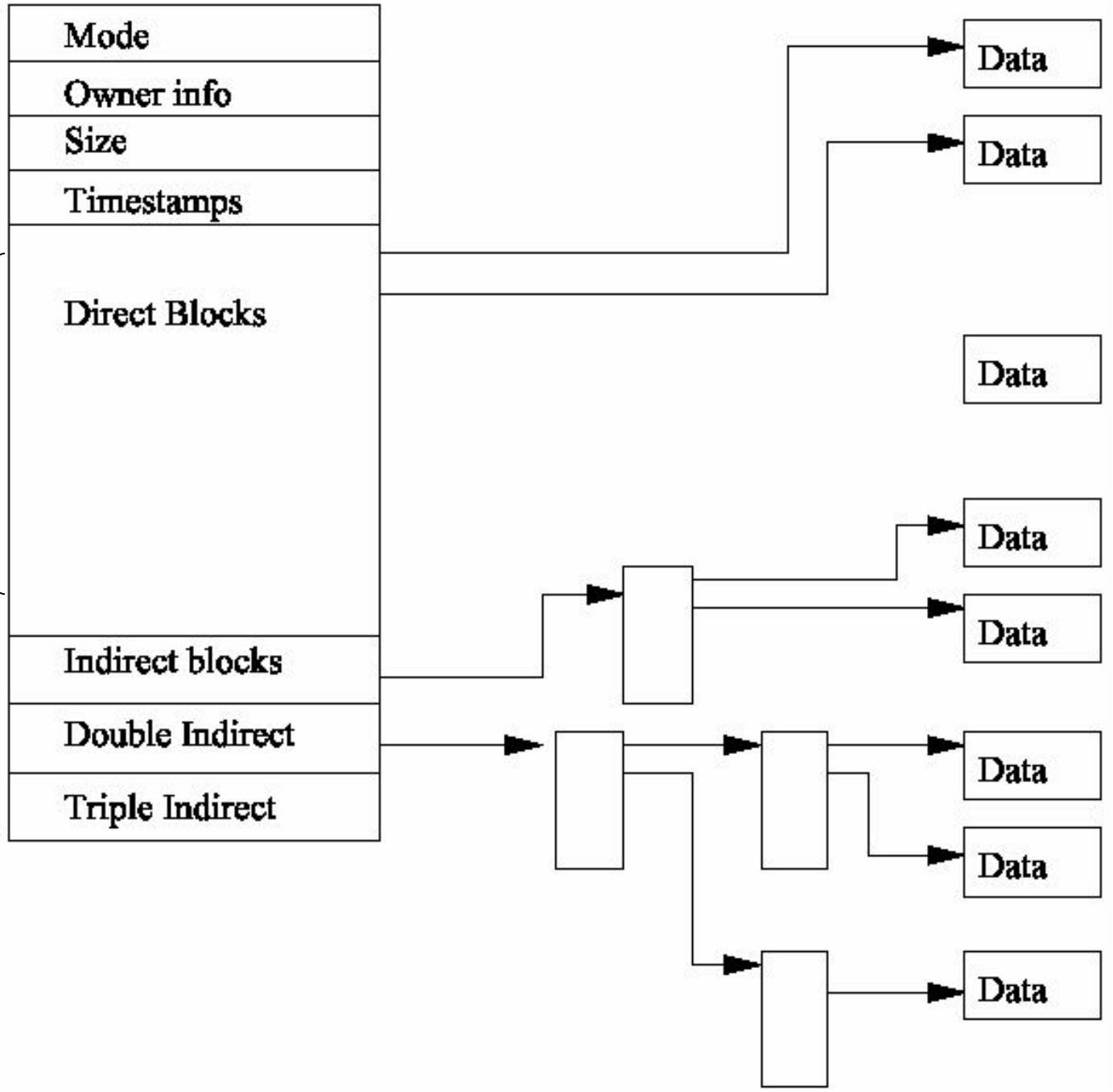
i-nodes (1)


disk drive
partition
partition
partition
file system
i-list
directory blocks and data block
boot block
super block
i-node
i-node
i-node
i-list
data block
data block
directory block
data block
directory block
"directory file"
i-node
i-node
i-node
i-node number
filename

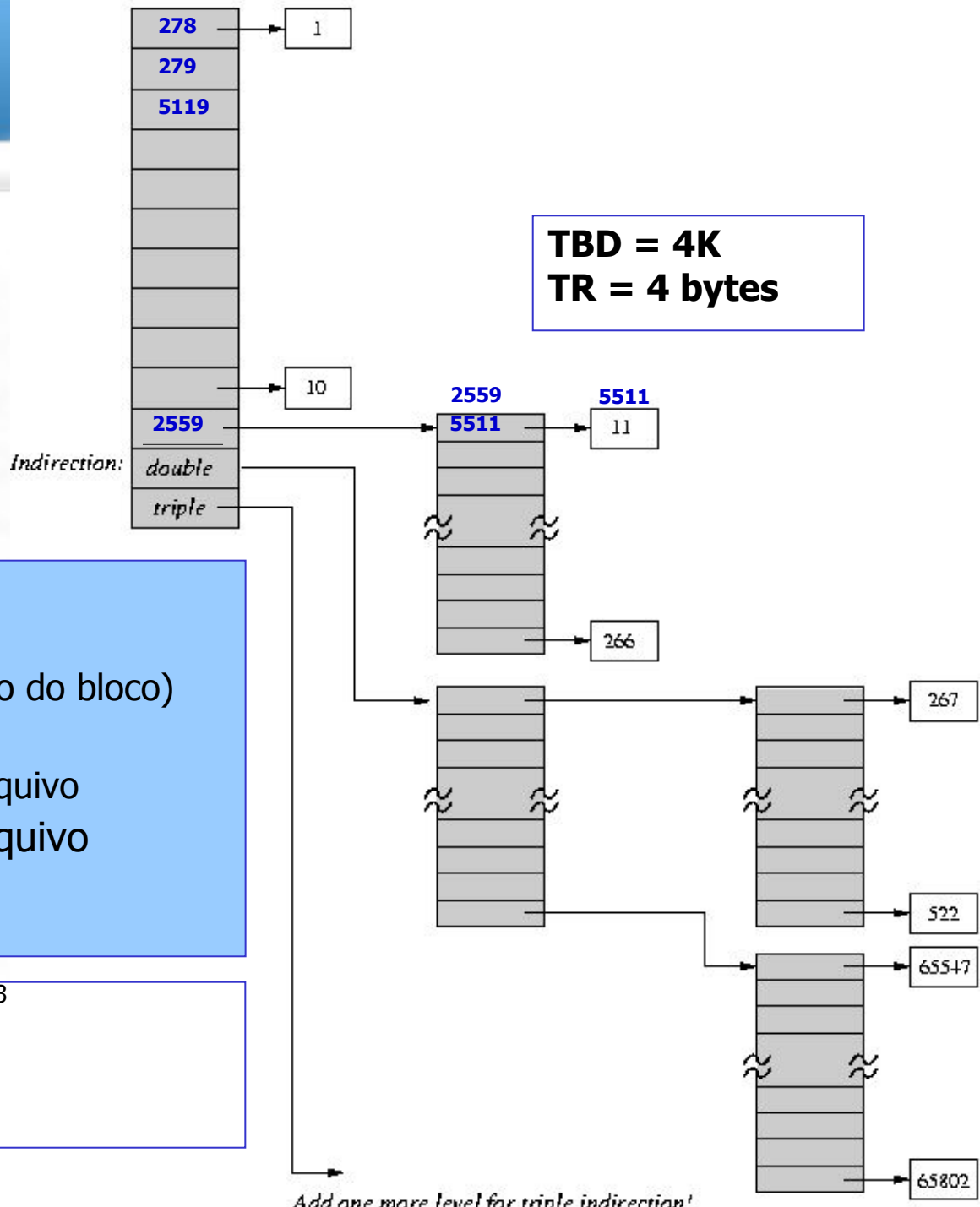
i-nodes (2)

10 entradas diretas

A quantidade de entradas diretas (*direct block pointers*) **varia** de implementação para implementação



i-nodes (3)



TBD = 4K
TR = 4 bytes

Dado que:

TBD – Tamanho bloco de dados

TR – Tamanho referência (endereço do bloco)

Qual será...?

B_{max} – Nº Blocos máximo de um arquivo

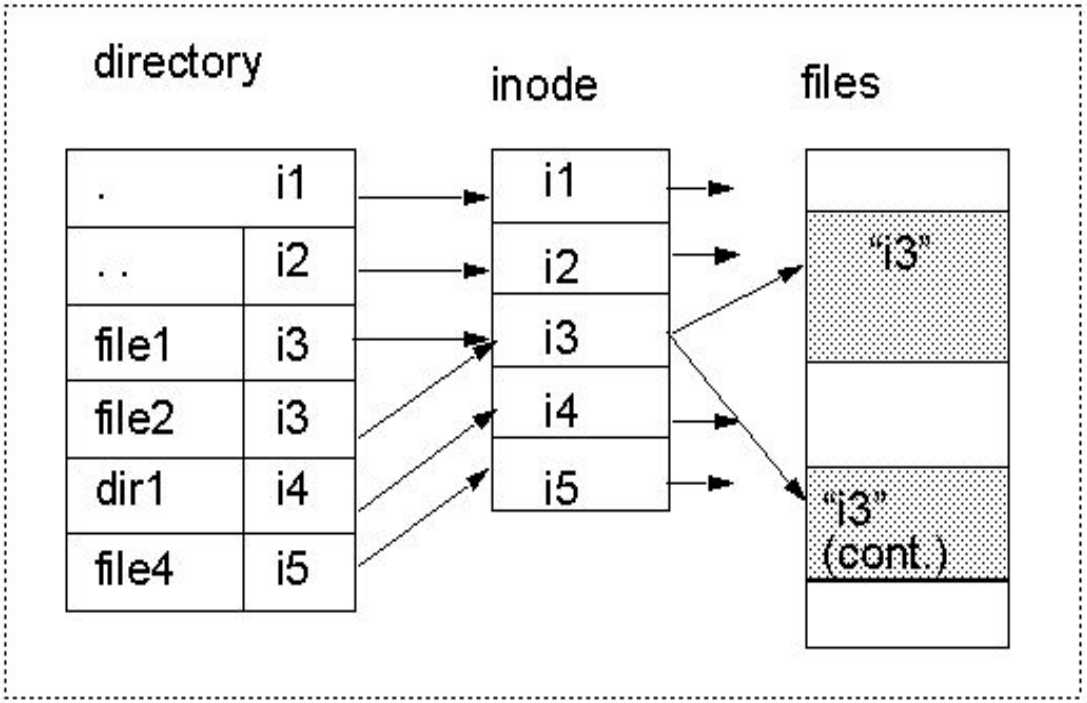
F_{max} – dimensão máxima de um arquivo

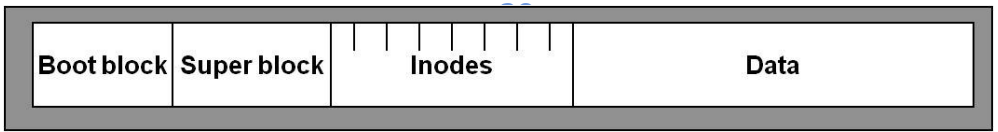
$$B_{max} = 10 + \frac{TBD}{TR} + \left(\frac{TBD}{TR}\right)^2 + \left(\frac{TBD}{TR}\right)^3$$

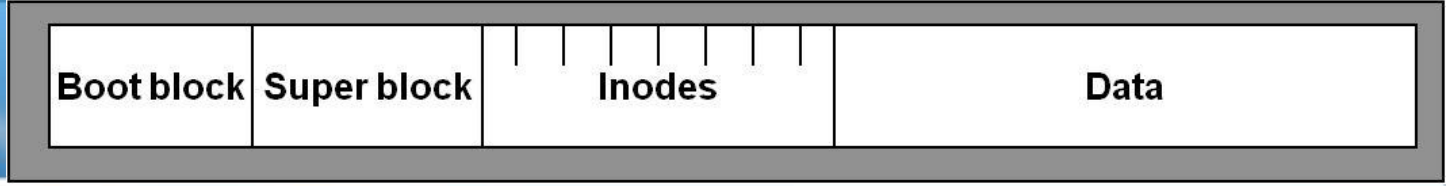
$$F_{max} = B_{max} \times TBD$$

Relação entre Diretórios e i-nodes (1)

- Diretórios incluem nomes de arquivos e referências para os respectivos i-nodes

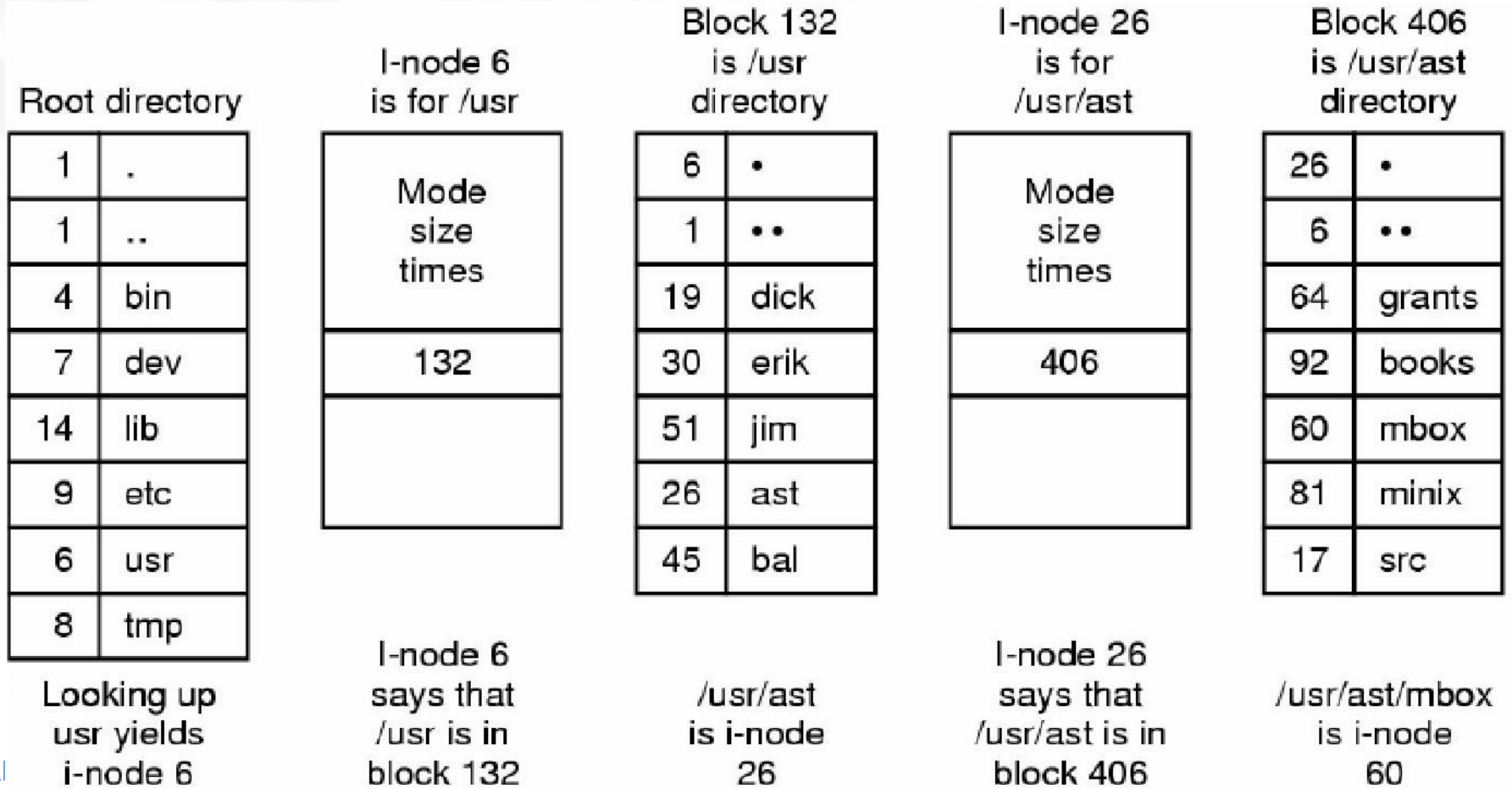






Relação entre Diretórios e i-nodes (2)

- Passos para alcançar /usr/ast/mbox



Laboratório de Pesquisa em Redes e Multimídia

Sistemas de Arquivos

Diretórios

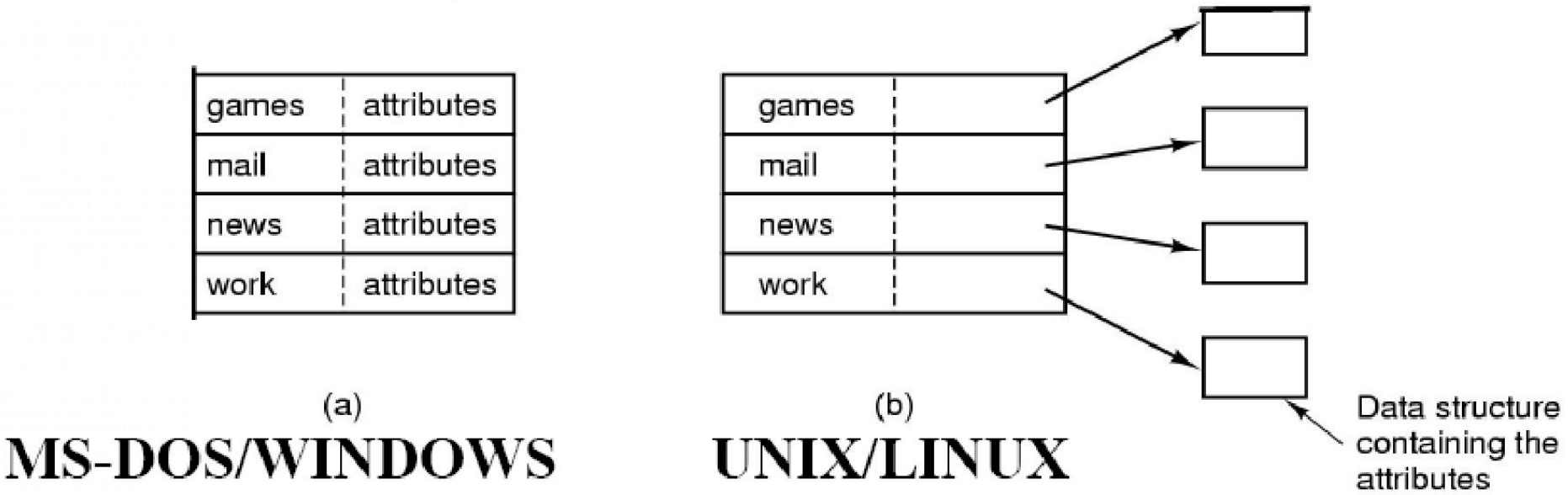
Gerenciamento de Espaço em Disco

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**

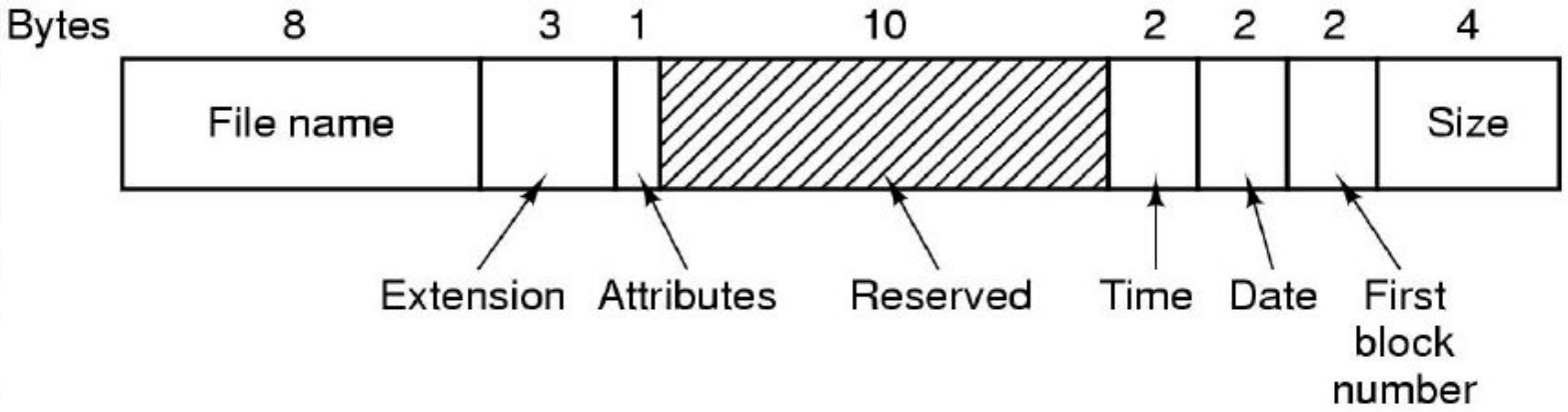
Implementação de Diretórios (2)



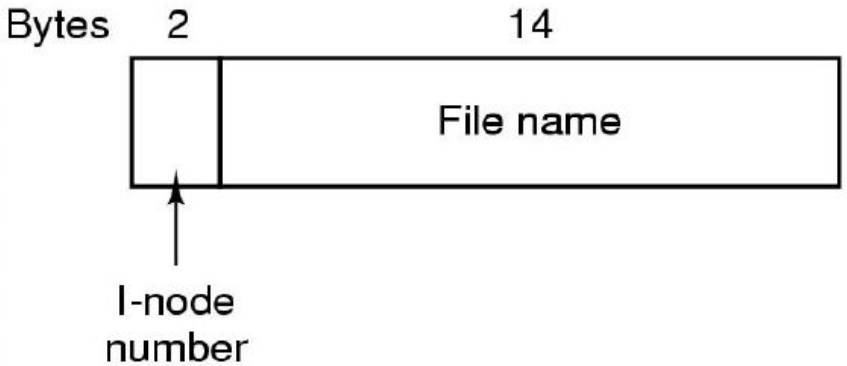
- (a) Diretório simples com
  - Entradas de dimensão fixa
  - Endereços de disco e atributos na entrada de diretório
- (b) Diretório em que cada entrada apenas refere um i-node

Implementação de Diretórios (3)

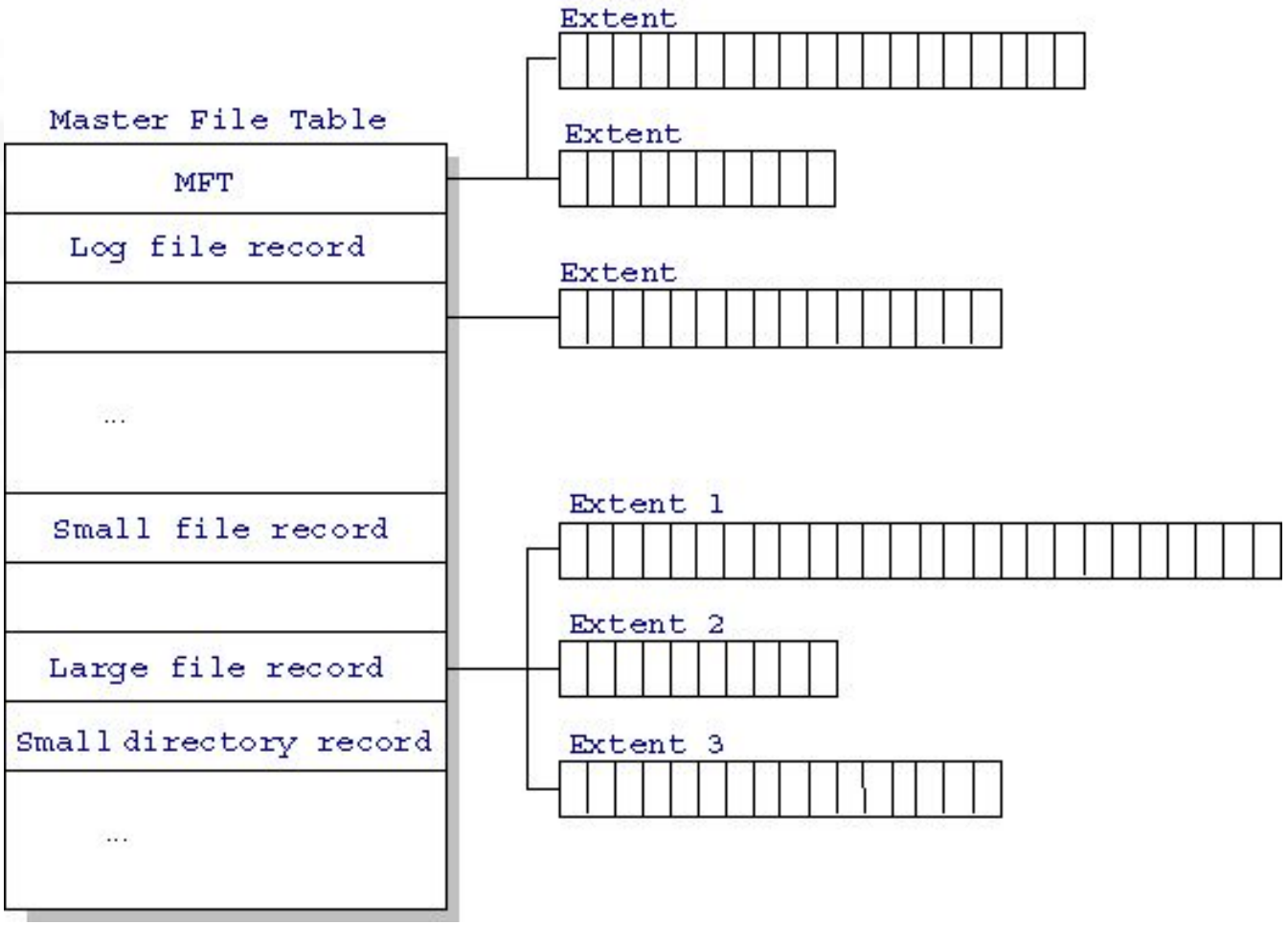
- Entrada de diretório no DOS



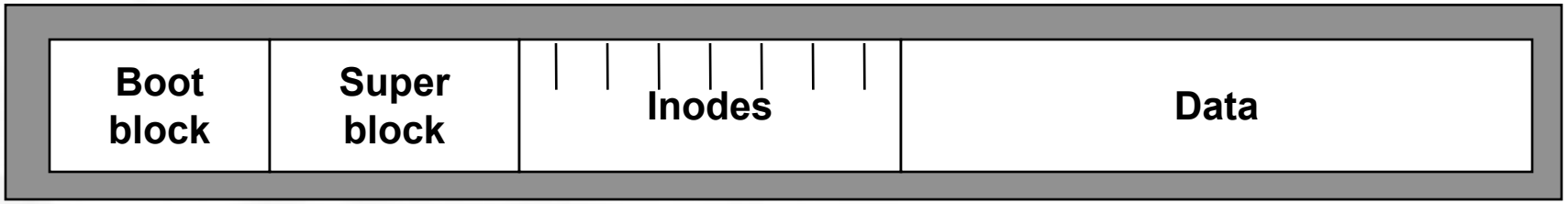
- No Unix (*eg*. System V)



Implementação de Diretórios - 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

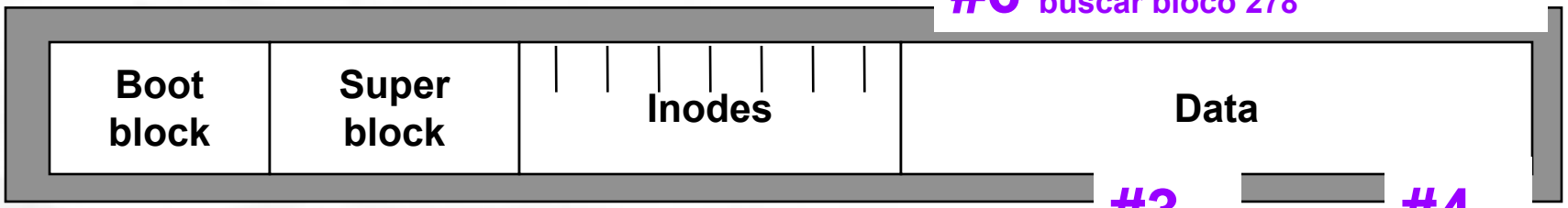
26	•
6	••
64	grants
92	books
60	mbox
81	minix
17	src

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

Implementação de Diretórios UNIX

#5 buscar inode 92

#6 buscar bloco 278



#1 I-node 6 is for /home

Mode
size
times
132

#2 Block 132 is /home directory

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

#3 I-node 26 is for /home/joao

Mode
size
times
406

#4 Block 406 is /usr/ast directory

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

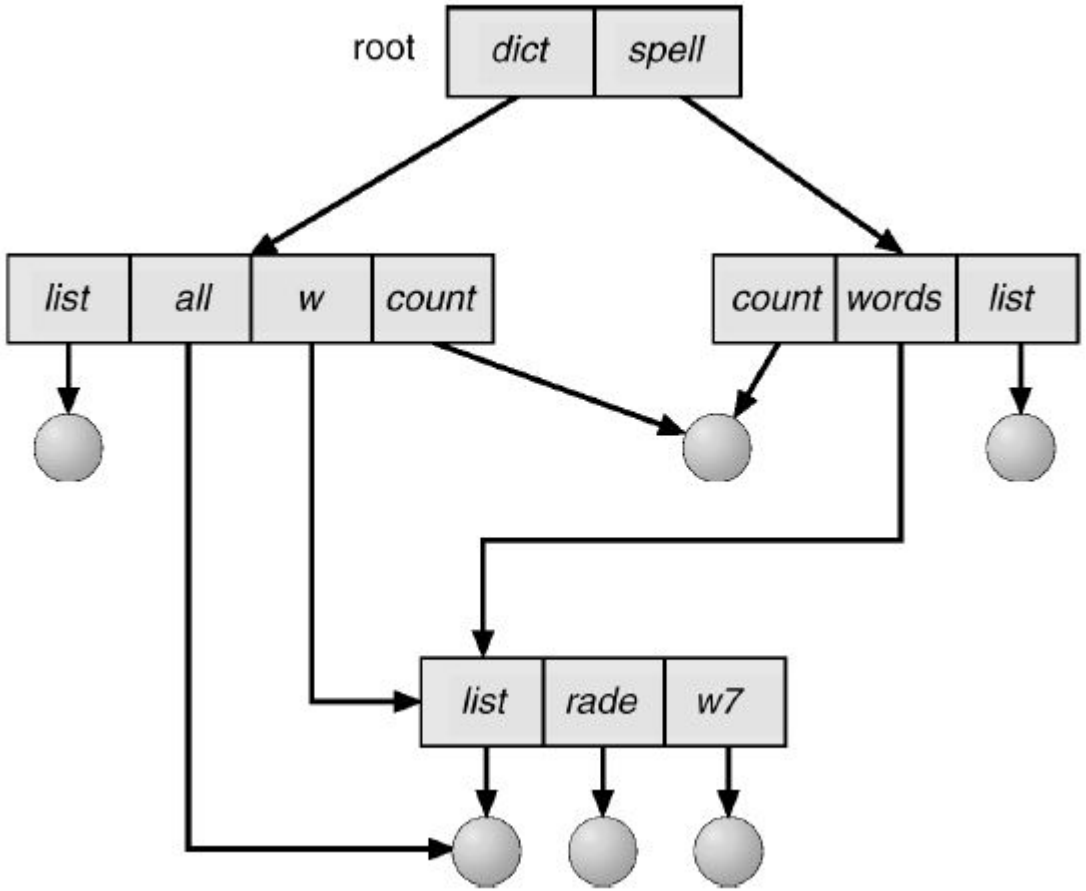
Root directory

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

- Quais os passos para alcançar o arquivo /home/joao/psh.c?

Arquivos Compartilhados (1)

- Hierarquia de diretórios: Grafo acíclico orientado

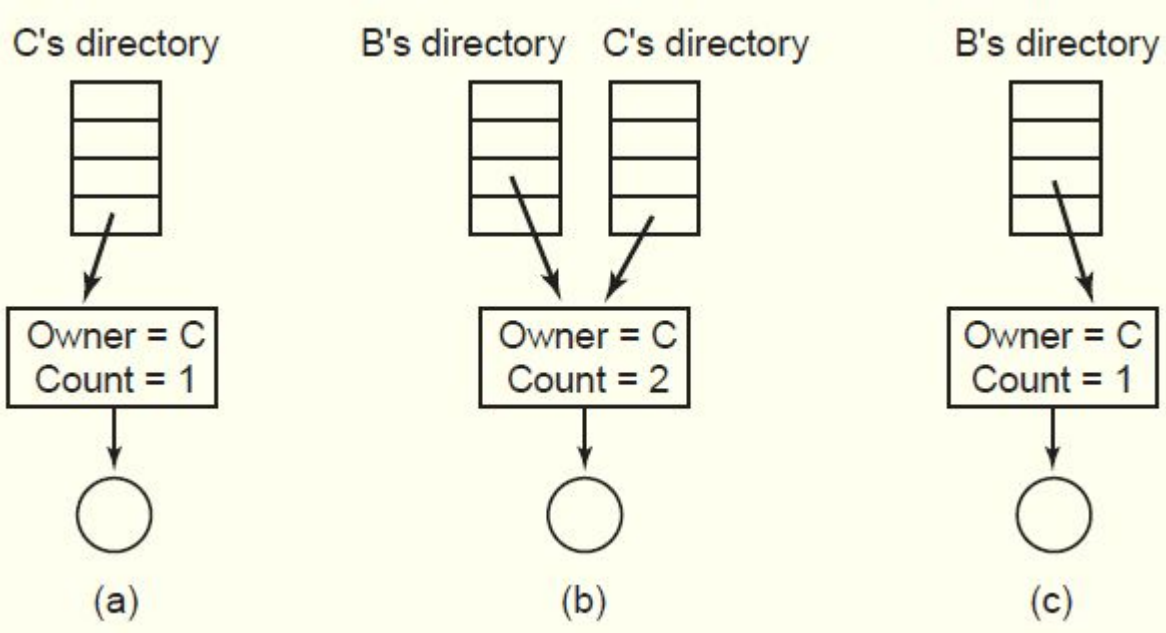


Arquivos Compartilhados (2)

- 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"

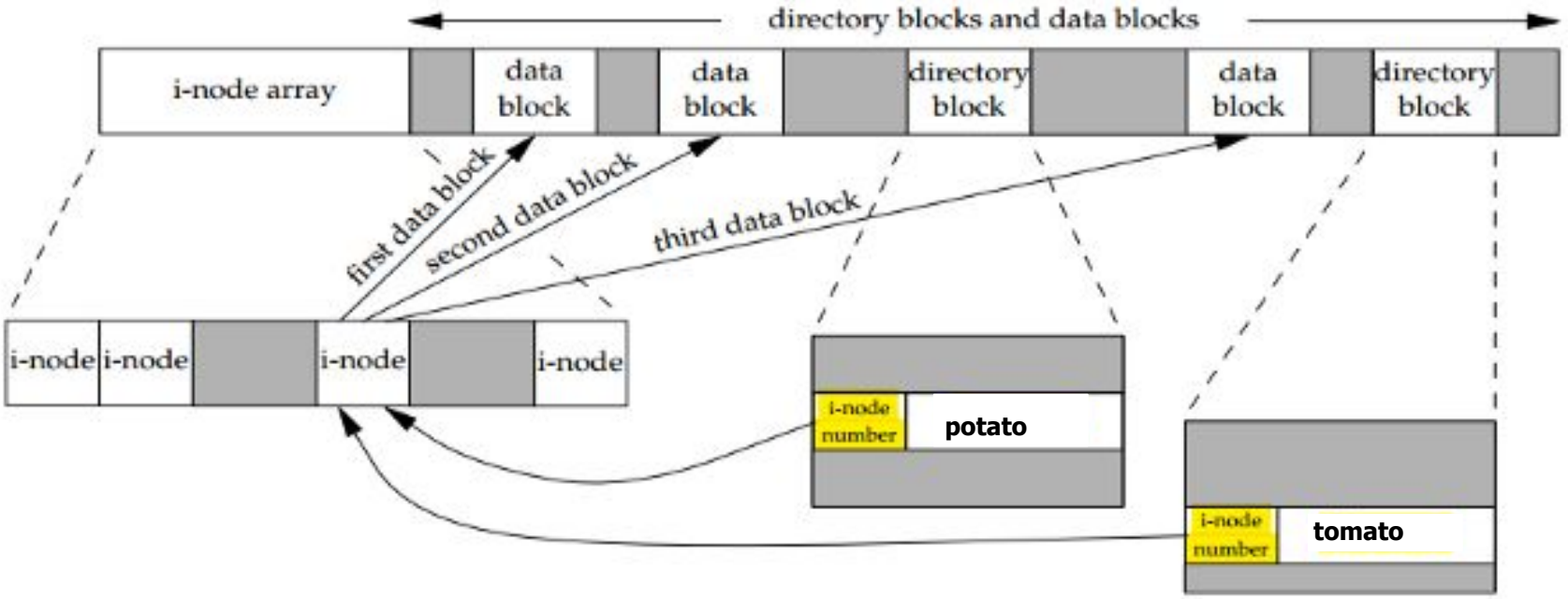
Arquivos Compartilhados (3)

- 1a. Solução



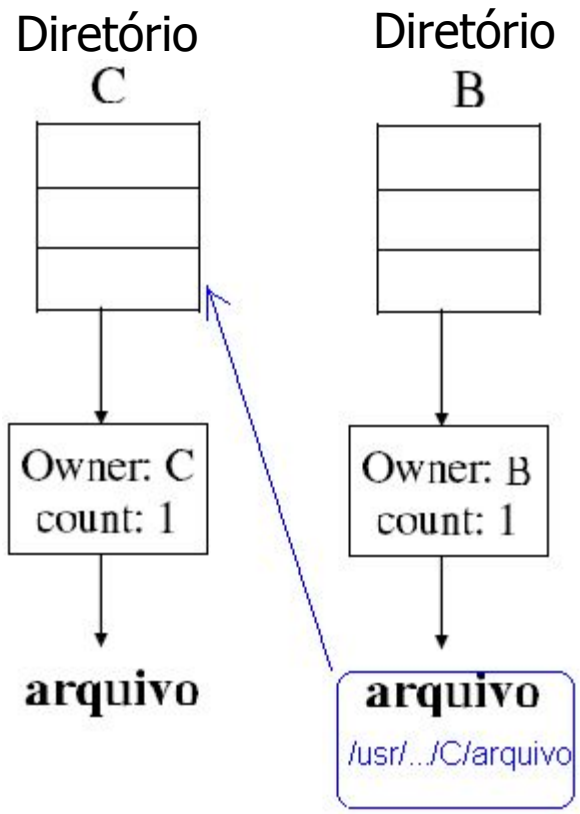
Arquivos Compartilhados (3)

- 1a. Solução - Como é possível no Unix?
  - Temos diferentes entradas em diferentes diretórios que referenciam **o mesmo i-node**



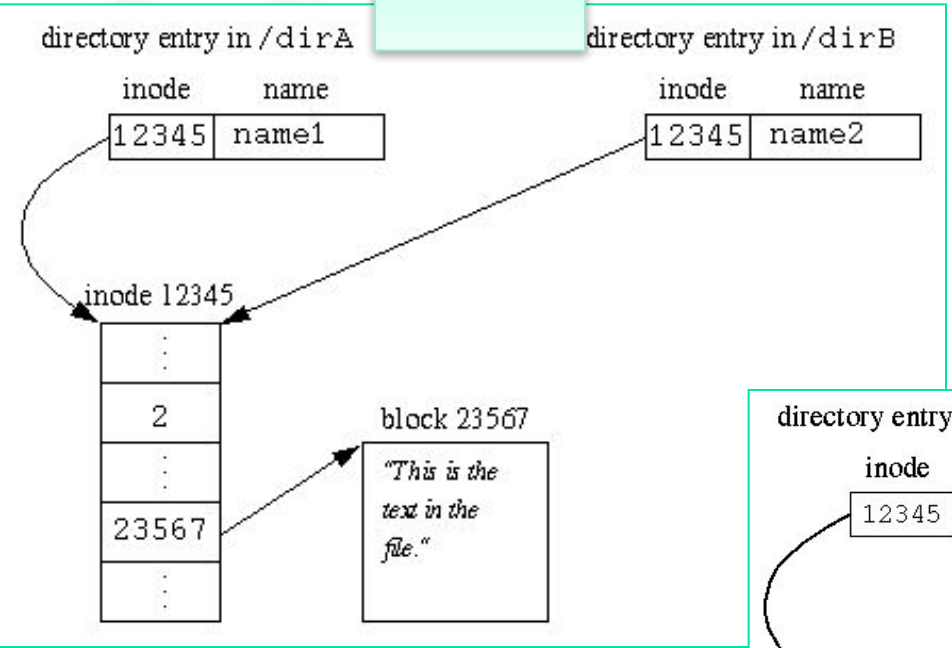
Arquivos Compartilhados (3)

- 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

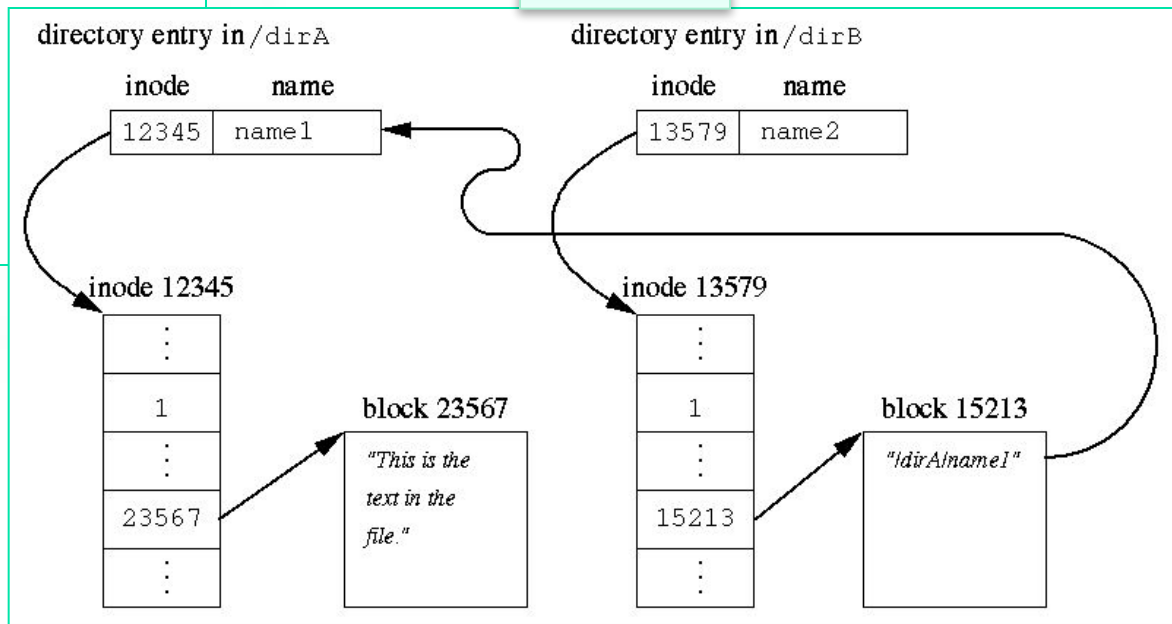


Arquivos Compartilhados (4)

Hard Link



Soft Link



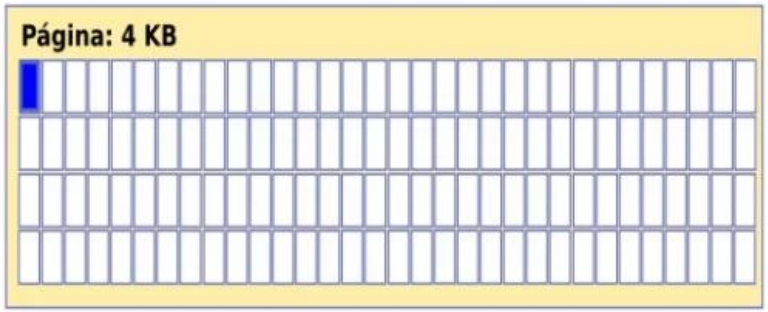
Gerenciamento de Espaço em Disco (HDs) (1)

- 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 em discos magnéticos

Tmp médio de seek + latência rotacional + tempo de leitura

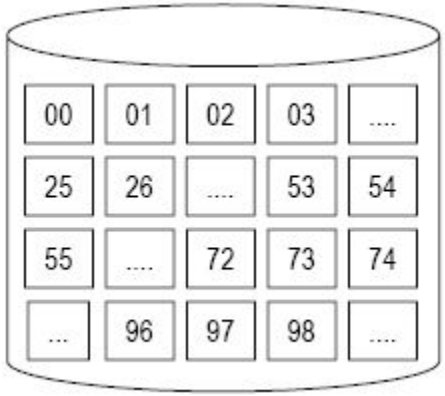
- E no SSD?



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.)