



Sistemas Operacionais

Arquivos

Carlos Ferraz (cagf@cin.ufpe.br)

Jorge Cavalcanti Fonsêca (jcbf@cin.ufpe.br)

Copyright



Copy Right

Carlos Ferraz – Cin/UFPE



Armazenamento da Informação a Longo Prazo

1. Deve ser possível **armazenar** uma quantidade muito grande de informação
2. A informação deve sobreviver ao término do **processo** que a usa – persistência
3. Múltiplos **processos** devem ser capazes de acessar a informação concorrentemente – **compartilhamento**



Armazenamento da Informação a Longo Prazo

- De maneira geral, duas operações são necessárias:
 - Ler e escrever em determinado bloco do disco (sequências lineares de blocos).
- Porém...
 - Como encontrar a informação?
 - Como impedir que um usuário tenha acesso a informações de outro usuário?
 - Como saber quais blocos estão livres?

Arquivos – abstração



S.O. (Conceitos)

- *“Juntas, as abstrações de processos (e threads), espaços de endereçamento e arquivos são os conceitos mais importantes relacionados aos sistemas operacionais.”*

- *“Se de fato você compreender esses três conceitos do começo ao fim, você estará no caminho certo para se tornar um especialista em sistemas operacionais.”*

Tanenbaum



Infra-estrutura de Software

Orientada a computação/processamento → processo

- ▶ Escalonamento e concorrência – *processador*
 - ▶ Memória virtual
 - ▶ Paginação
 - ▶ Segmentação
 - ▶ Captura de dados
 - ▶ Armazenamento persistente – sistema de arquivos – *disco*
 - ▶ Apresentação – saída – *periféricos*
 - ▶ Processamento distribuído – *rede*
- Diagrammatic annotations:
- A bracket groups "Memória virtual", "Paginação", and "Segmentação" with the label *memória*.
 - A bracket groups "Captura de dados" and "Armazenamento persistente" with the label *entrada – periféricos*.



Infra-estrutura de Software

Orientada a computação/processamento → processo

- ✓ Escalonamento e concorrência – *processador*
 - ✓ Memória virtual
 - ✓ Paginação
 - ✓ Segmentação
- } *memória*

- ▶ Captura de dados – entrada – *periféricos*
- ▶ Armazenamento persistente – sistema de arquivos – *disco*
- ▶ Apresentação – saída – *periféricos*
- ▶ Processamento distribuído – *rede*

E/S



Infra-estrutura de Software

Orientada a computação/processamento → processo

- ✓ Escalonamento e concorrência – *processador*
 - ✓ Memória virtual
 - ✓ Paginação
 - ✓ Segmentação
- } *memória*
- ▶ Captura de dados – entrada – *periféricos*
 - ▶ Armazenamento persistente – **sistema de arquivos** – *disco*
 - ▶ Apresentação – saída – *periféricos*
 - ▶ Processamento distribuído – *rede*



Introdução

- ▶ Os arquivos são gerenciados pelo sistema operacional e é mediante a implementação de **arquivos** que o sistema operacional **estrutura e organiza as informações**
- ▶ A parte do sistema responsável pela gerência é denominada **sistema de arquivo**, que é a parte mais visível do sistema operacional, pois é uma atividade freqüentemente realizada pelos usuários
- ▶ Deve ocorrer de maneira **uniforme**, independente dos diferentes dispositivos de armazenamento



Arquivo

- ▶ É um **conjunto de registros** definidos pelo sistema de arquivos e podem ser armazenados em diferentes dispositivos físicos
- ▶ É constituído de informações logicamente relacionadas, podendo representar programas ou dados
- ▶ São unidades lógicas de informação criadas por processos.
 - ▶ Uma espécie de espaço de endereçamento para modelar o disco



Idéias Básicas

- ▶ A representação da informação dentro de um computador é feita na forma de *números*
- ▶ No Linux ou no Windows, arquivos são concebidos e tratados como STREAMS de BYTES



Decifrando os números: o código

A : 65	a : 97	0 : 48
B : 66	b : 98	1 : 49
...
Z : 90	z : 122	9 : 57

Caracteres especiais :

0	: NULO
8	: BACKSPACE
9	: TAB
10	: mudança de linha
12	: mudança de página
13	: ENTER



Representação de um arquivo

Representação interna do conteúdo de um arquivo

65	66	67	10	49	50	51	10	68	69	10	97	98	10
----	----	----	----	----	----	----	----	----	----	----	----	----	----

Convertendo a representação interna de um arquivo em um arquivo texto:

A	B	C	10	1	2	3	10	D	E	10	a	b	10
---	---	---	----	---	---	---	----	---	---	----	---	---	----



*mudança
de linha*



*mudança
de linha*



*mudança
de linha*



*mudança
de linha*



Arquivo (Identificação)

- ▶ É identificado por meio de um **nome**, formado por uma seqüência de caracteres
- ▶ Em alguns sistemas operacionais, a identificação de um arquivo é composta por duas partes separadas por um ponto
 - ▶ a parte após o ponto é chamada *extensão* do arquivo e serve para identificar o conteúdo
- ▶ Ex. <nome_arquivo>.<extensão> → Aula.ppt



Arquivo (Identificação)

Extensão	Significado
.bak	Cópia de segurança
.c	Código-fonte de programa em C
.gif	Imagem no formato <i>Graphical Interchange Format</i>
.hlp	Arquivo de ajuda
.html	Documento em HTML
.jpg	Imagem codificada segundo padrões JPEG
.mp3	Música codificada no formato MPEG (camada 3)
.mpg	Filme codificado no padrão MPEG
.o	Arquivo objeto (gerado por compilador, ainda não ligado)
.pdf	Arquivo no formato PDF (<i>Portable Document File</i>)
.ps	Arquivo PostScript
.tex	Entrada para o programa de formatação TEX
.txt	Arquivo de texto
.zip	Arquivo compactado

Tabela 4.1 Algumas extensões comuns de arquivos.

Organização de Arquivos

- ▶ Consiste no modo como seus dados estão internamente armazenados
- ▶ A **forma mais simples** de organização de arquivos é através de uma *seqüência não-estruturada de bytes*, onde o sistema de arquivos não impõe qualquer estrutura lógica para os dados – a aplicação deve definir toda a organização

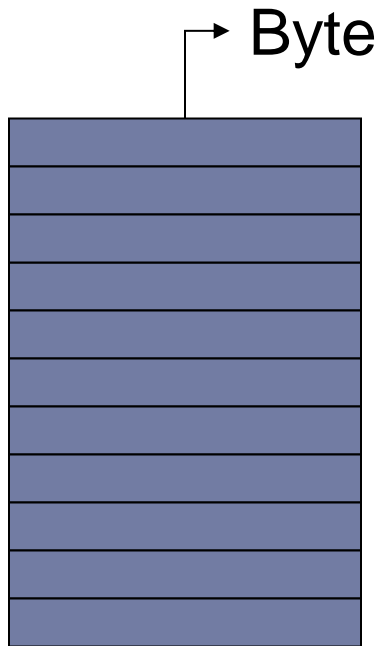


Organização de Arquivos

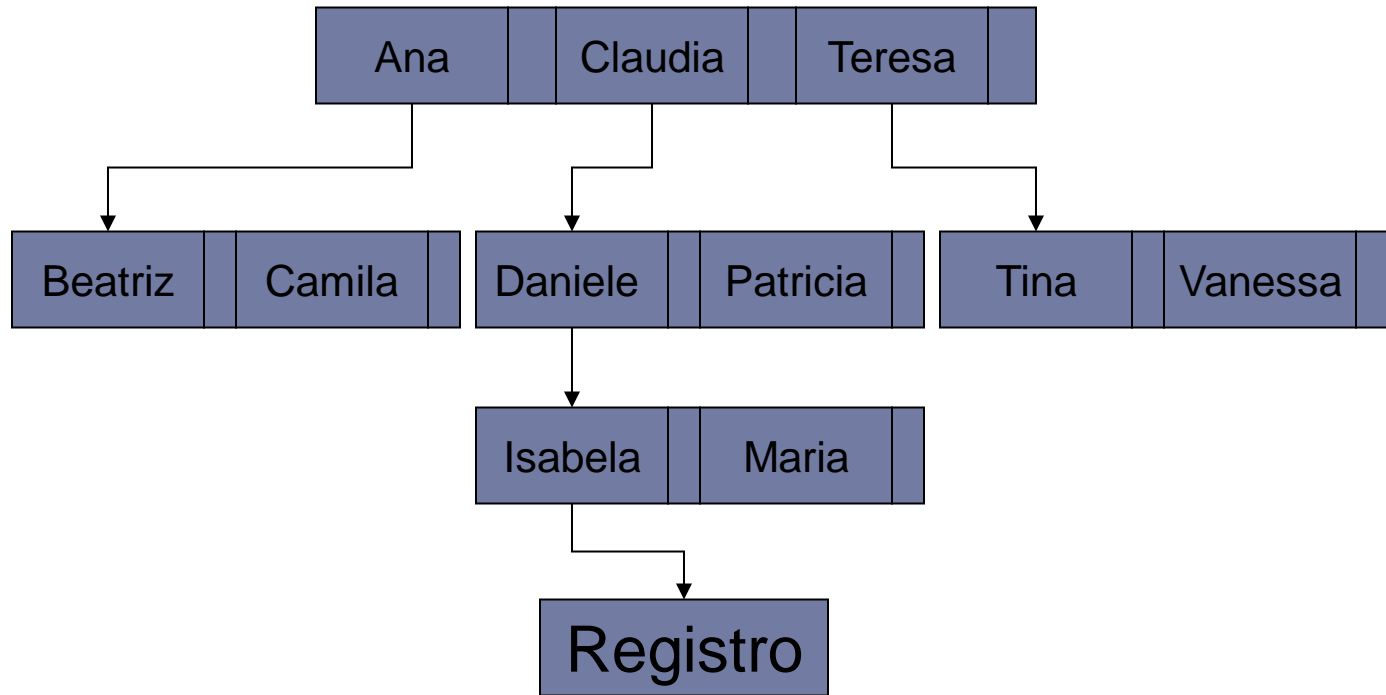
- ▶ A grande **vantagem** deste modelo é a *flexibilidade* para criar estruturas de dados, porém todo o controle de dados é de responsabilidade da aplicação
- ▶ Alguns sistemas operacionais estabelecem diferentes organizações de arquivos e cada um deve seguir um modelo suportado pelo sistema de arquivos
- ▶ As organizações mais conhecidas e implementadas são: **seqüencial, relativa e indexada**



Organização de Arquivos



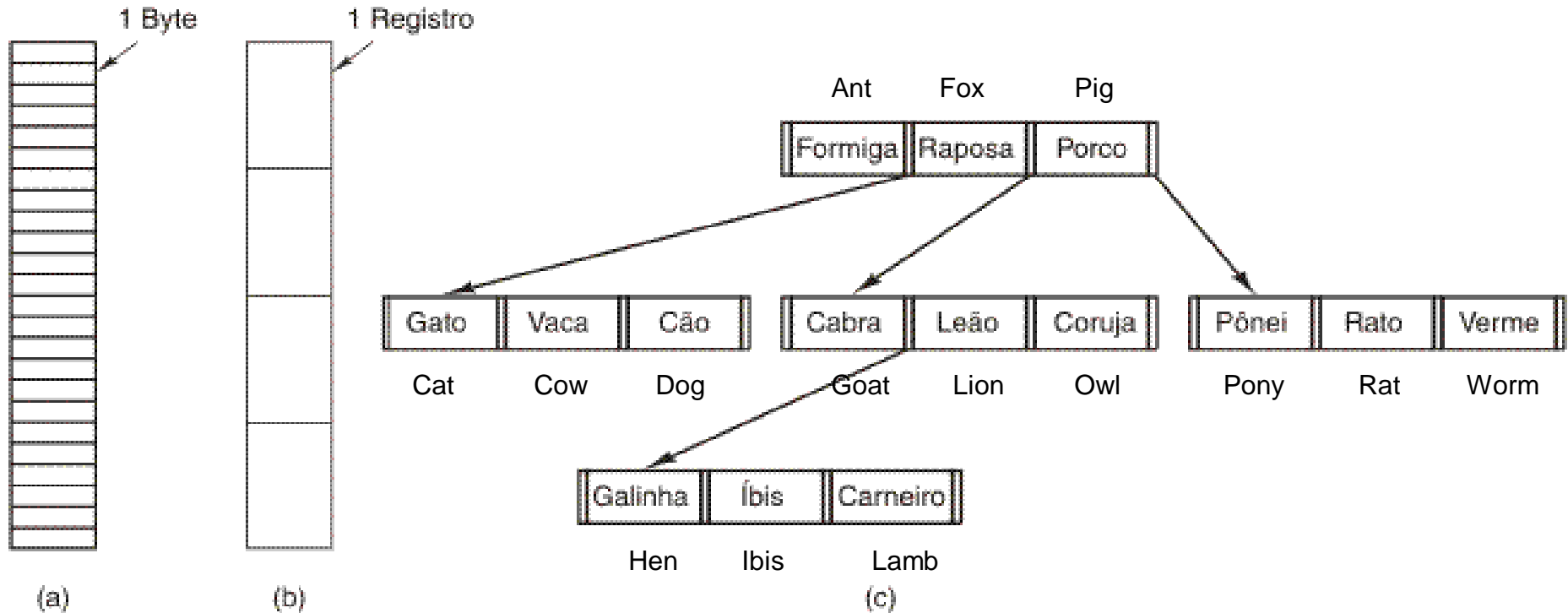
Organização não-estruturada



Organização Indexada



Estrutura de Arquivos

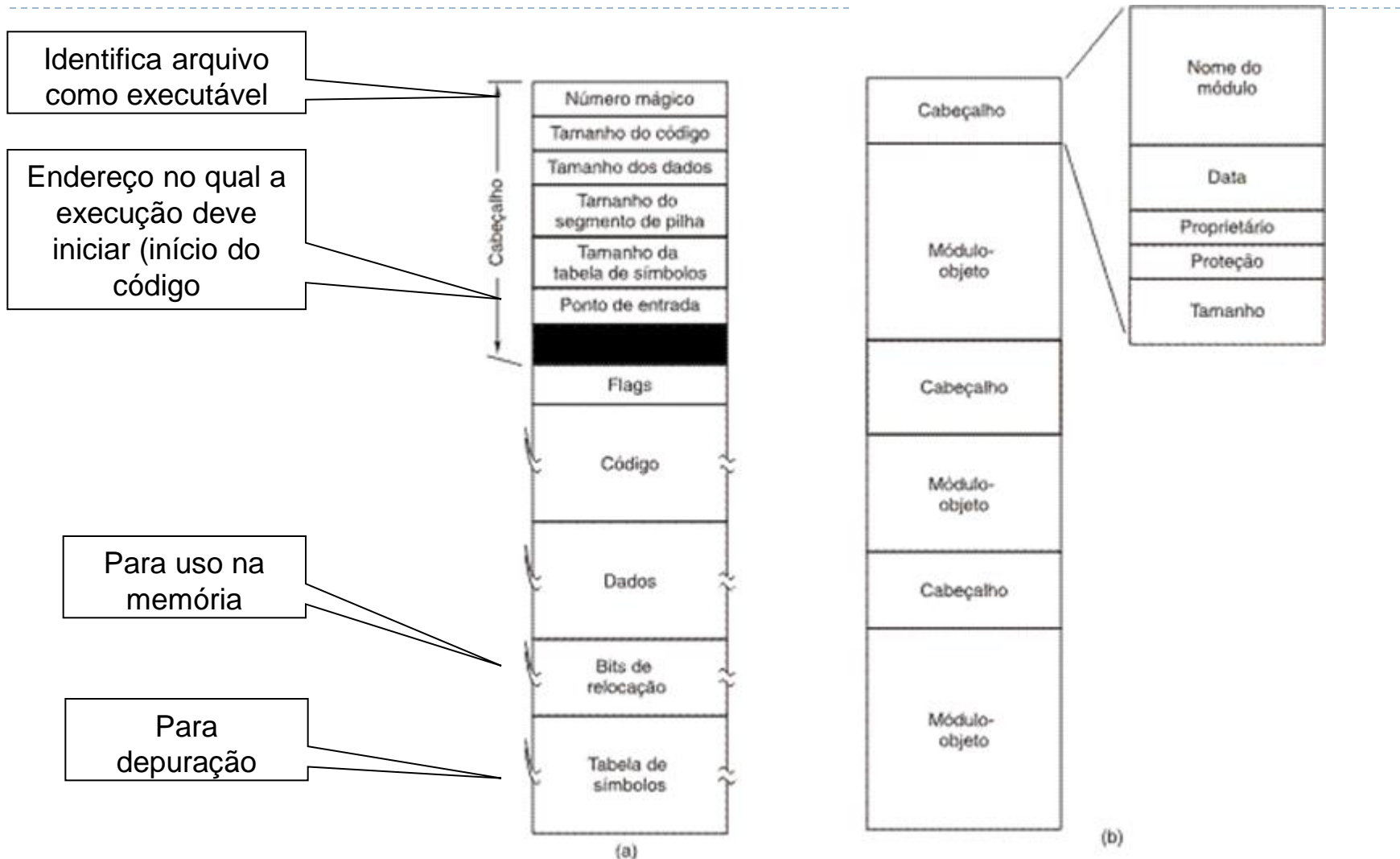


Três tipos de arquivos

- a) seqüência de bytes
- b) seqüência de registros
- c) árvore



Tipos de Arquivos



(a) Um arquivo executável (b) Um repositório (*archive*)

Métodos de Acesso

- ▶ Em função de como o arquivo está organizado, o sistema de arquivos pode recuperar registros de diferentes maneiras:
 - ▶ **Acesso seqüencial:** arquivos armazenados em fitas magnéticas
 - ▶ o acesso era restrito à leitura na ordem em que eram gravados, sendo a gravação de arquivos possível apenas no final (APPEND)
 - ▶ Não é possível “saltar” e ler fora da ordem



Métodos de Acesso

- ▶ **Acesso direto:** permite a leitura/gravação de um registro diretamente na sua posição
 - ▶ É realizado através do número de registro
 - ▶ Não existe restrição à ordem em que os registros são lidos ou gravados
 - ▶ Somente é possível quando é definido com registros de tamanho fixo

- ▶ Pode-se combinar o acesso sequencial com o direto:
 - Acessar diretamente um arquivo e em seguida em forma sequencial



Métodos de Acesso

- ▶ **Acesso indexado** ou **por chave**: o arquivo deve possuir uma área de índice onde existam ponteiros para os diversos registros e a partir desta informação realiza-se um acesso direto

- ▶ Arquivo cujo bytes podem ser lidos em qualquer ordem são chamados de **arquivos de acesso aleatório** (necessários para muitas aplicações).
 - ▶ Ex. Banco de Dados
 - ▶ **Read and Seek**



Acesso Direto



Operações de Entrada / Saída

- ▶ O sistema de arquivos oferece um conjunto de **system calls** que permite às aplicações realizar operações de E/S, como tradução de nomes em endereços, leitura e gravação de dados e criação/eliminação de arquivos
- ▶ As **system calls** de E/S têm como função oferecer uma interface simples e uniforme entre a aplicação e os diversos dispositivos

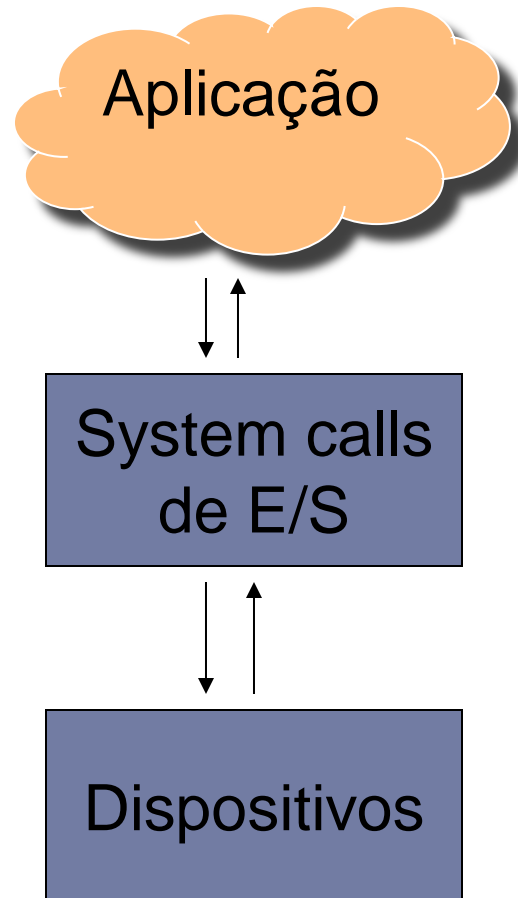


S.O. como uma máquina estendida

- ▶ *I/O de disco*
 - ▶ *Movimentação do braço do disco, formatação das trilhas, inicialização, sinalização, reinicialização e recalibração do controlador do disco.*
- ▶ *Read:*
 - ▶ *13 parâmetros:*
 - ▶ *Endereço do bloco de dados a ser lido*
 - ▶ *Numero de setores por trilha*
 - ▶ *Modo de gravação usado no meio fisico*
 - ▶ *Espaço livre entre setores*
 - ▶ *Retorno:*
 - *23 campos de status e de erros*
 - ▶ *Disco fléxivel:*
 - *Motor está ligado? (Ligado muito tempo gerá desgaste no disco)*
- ▶ *Abstração: Read block from file*



Operações de Entrada / Saída



Operações de Entrada/Saída

Comando	Descrição
<i>create</i>	Criação de um arquivo
<i>open</i>	Abertura de um arquivo
<i>read</i>	Leitura de dados de um arquivo
<i>write</i>	Gravação de dados de um arquivo
<i>close</i>	Fechamento de um arquivo
<i>rename</i>	Alteração de nome de um arquivo
<i>erase</i>	Eliminação de um arquivo



Atributos

▶ **Informações de controle**

- ▶ variam dependendo do sistema de arquivos
- ▶ alguns como **tamanho**, **criador**, **proteção** e **data** estão presentes em quase todos
- ▶ Alguns atributos especificados na criação do arquivo não podem ser mudados e outros são modificados pelo próprio sistema operacional
 - ▶ E ainda existem alguns que podem ser alterados pelo usuário tais como **proteção**, **tamanho** e **senha**.



Atributos de Arquivos

Atributos	Descrição
Tamanho	Especifica o tamanho do arquivo
Proteção	Código de proteção de acesso
Dono	Identifica o criador do arquivo
Criação	Data e hora da criação do arquivo
Backup	Data e hora do último backup realizado
Organização	Indica a organização lógica dos registros
Senha	Senha necessária para acessar o arquivo



Atributos

Atributo	Significado
Proteção	Quem tem acesso ao arquivo e de que modo
Senha	Necessidade de senha para acesso ao arquivo
Criador	ID do criador do arquivo
Proprietário	Proprietário atual
Flag de somente leitura	0 para leitura/escrita; 1 para somente leitura
Flag de oculto	0 para normal; 1 para não exibir o arquivo
Flag de sistema	0 para arquivos normais; 1 para arquivos do sistema
Flag de arquivamento	0 para arquivos com backup; 1 para arquivos sem backup
Flag de ASCII/binário	0 para arquivos ASCII; 1 para arquivos binários
Flag de acesso aleatório	0 para acesso somente sequencial; 1 para acesso aleatório
Flag de temporário	0 para normal; 1 para apagar o arquivo ao sair do processo
Flag de travamento	0 para destravados; diferente de 0 para travados
Tamanho do registro	Número de bytes em um registro
Posição da chave	Posição da chave em cada registro
Tamanho do campo-chave	Número de bytes no campo-chave
Momento de criação	Data e hora de criação do arquivo
Momento do último acesso	Data e hora do último acesso do arquivo
Momento da última alteração	Data e hora da última modificação do arquivo
Tamanho atual	Número de bytes no arquivo
Tamanho máximo	Número máximo de bytes no arquivo

Tabela 4.2 Alguns atributos possíveis de arquivos.

Diretórios

- ▶ Modo como o sistema **organiza os diferentes arquivos contidos num disco**
- ▶ É a estrutura de dados que contém entradas associadas aos arquivos onde estão informações como **localização física, nome, organização** e demais atributos
- ▶ Quando um arquivo é aberto, o sistema operacional procura a sua entrada na estrutura de diretórios, armazenando as informações do arquivo em uma tabela mantida na memória principal
 - ▶ Esta tabela contém todos os arquivos abertos, sendo fundamental para aumentar o desempenho das operações com arquivos

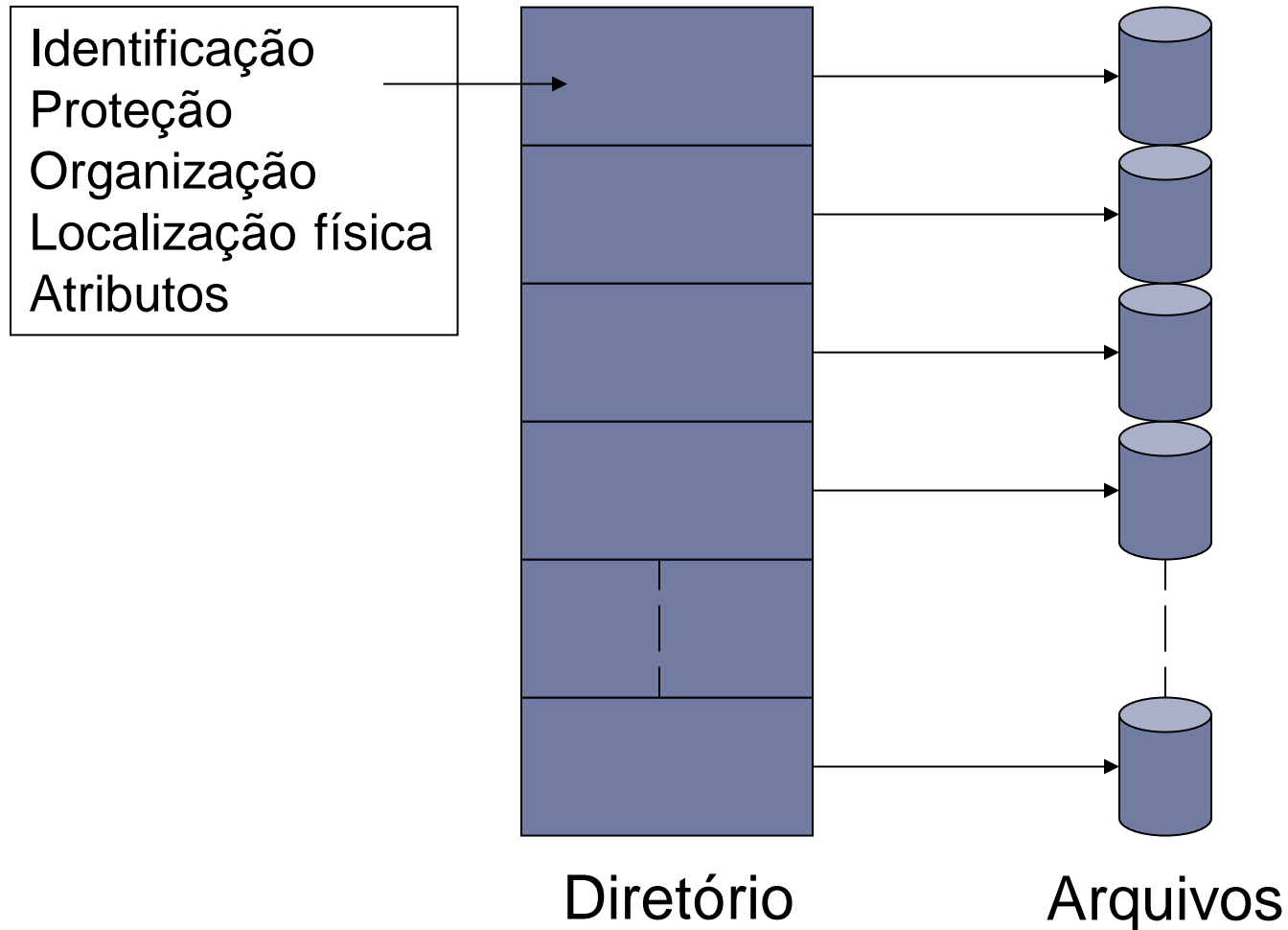


Diretórios

- ▶ **Nível Único (*single-level directory*): implementação mais simples**
 - ▶ existe apenas um único diretório contendo todos os arquivos do disco
 - ▶ bastante limitado já que não permite que usuários criem arquivos com o mesmo nome, o que ocasionaria um conflito no acesso aos arquivos



Estrutura de diretórios de nível único

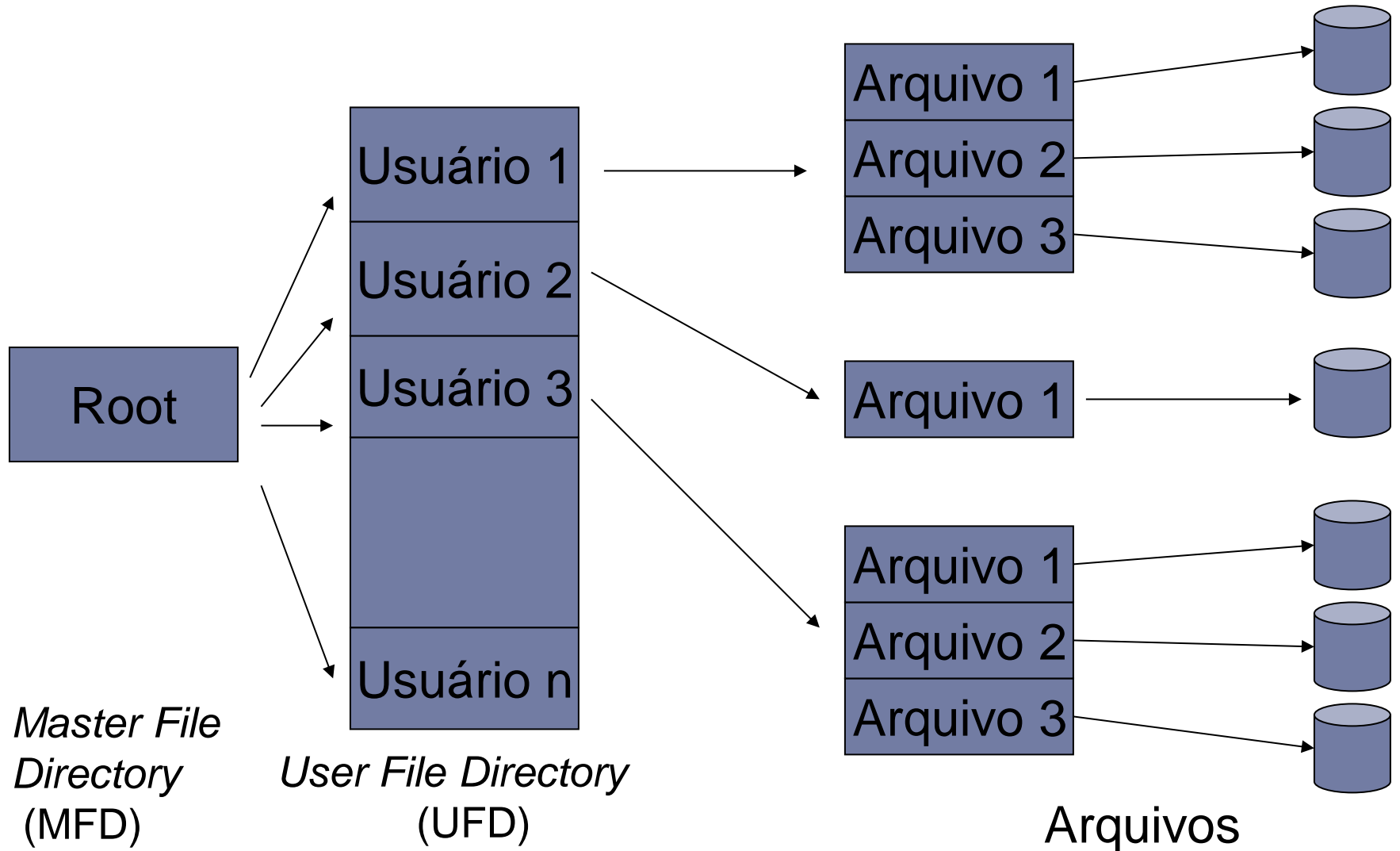


Diretórios

- ▶ *User File Directory* (UFD): para cada usuário existe um diretório particular e assim poderia criar arquivos com qualquer nome
- ▶ Deve haver um nível de diretório adicional para controle chamado de *Master File Directory* (MFD) que é indexado pelo nome do usuário e cada entrada aponta para o diretório pessoal
- ▶ É análoga a uma estrutura de dados em árvore onde o MFD é a raiz, os galhos são a UFD e os arquivos são as folhas
- ▶ Quando se referencia a um arquivo é necessário especificar seu nome e seu diretório – *path* (caminho)



Estrutura de diretórios com dois níveis

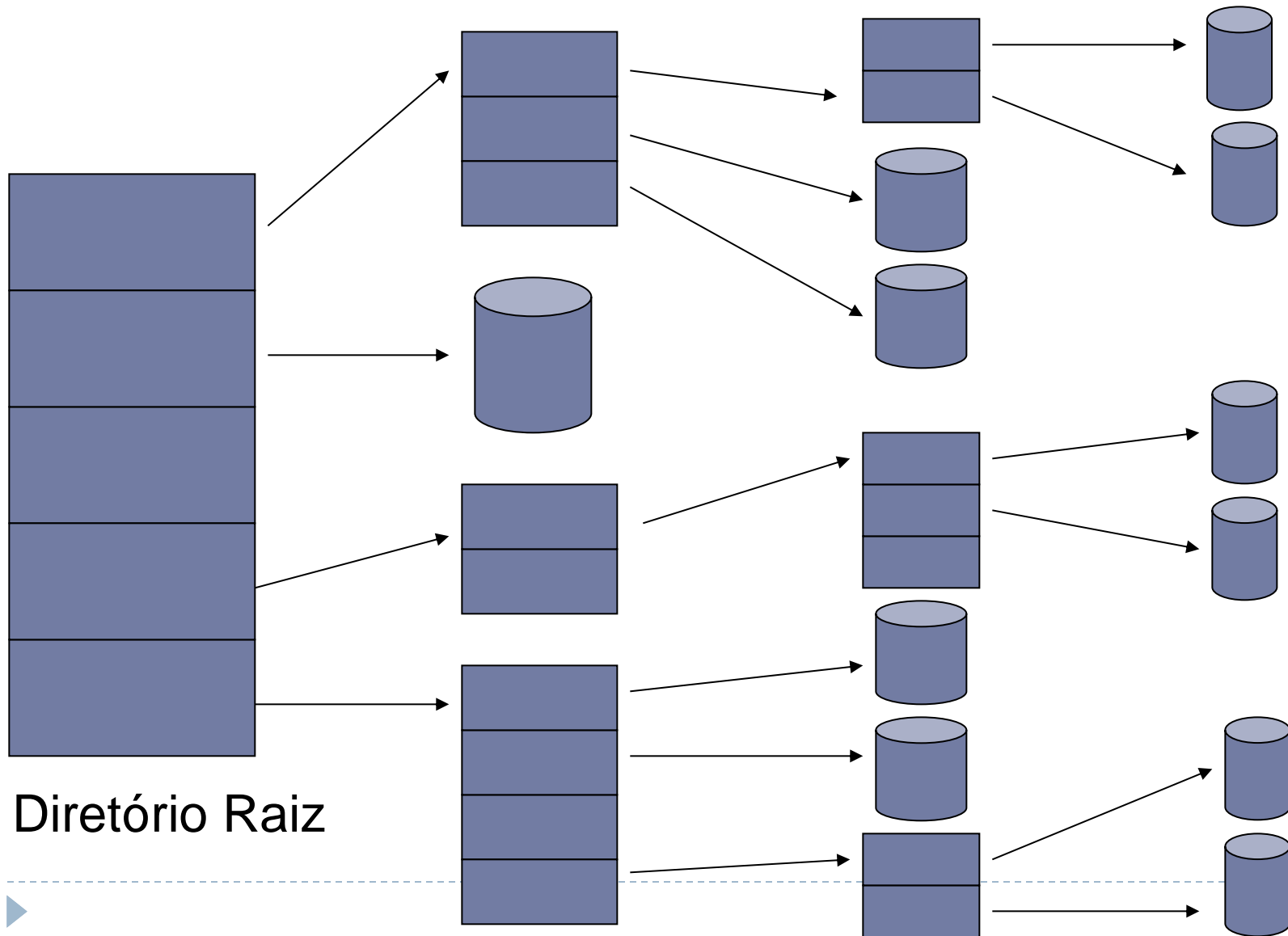


Diretórios

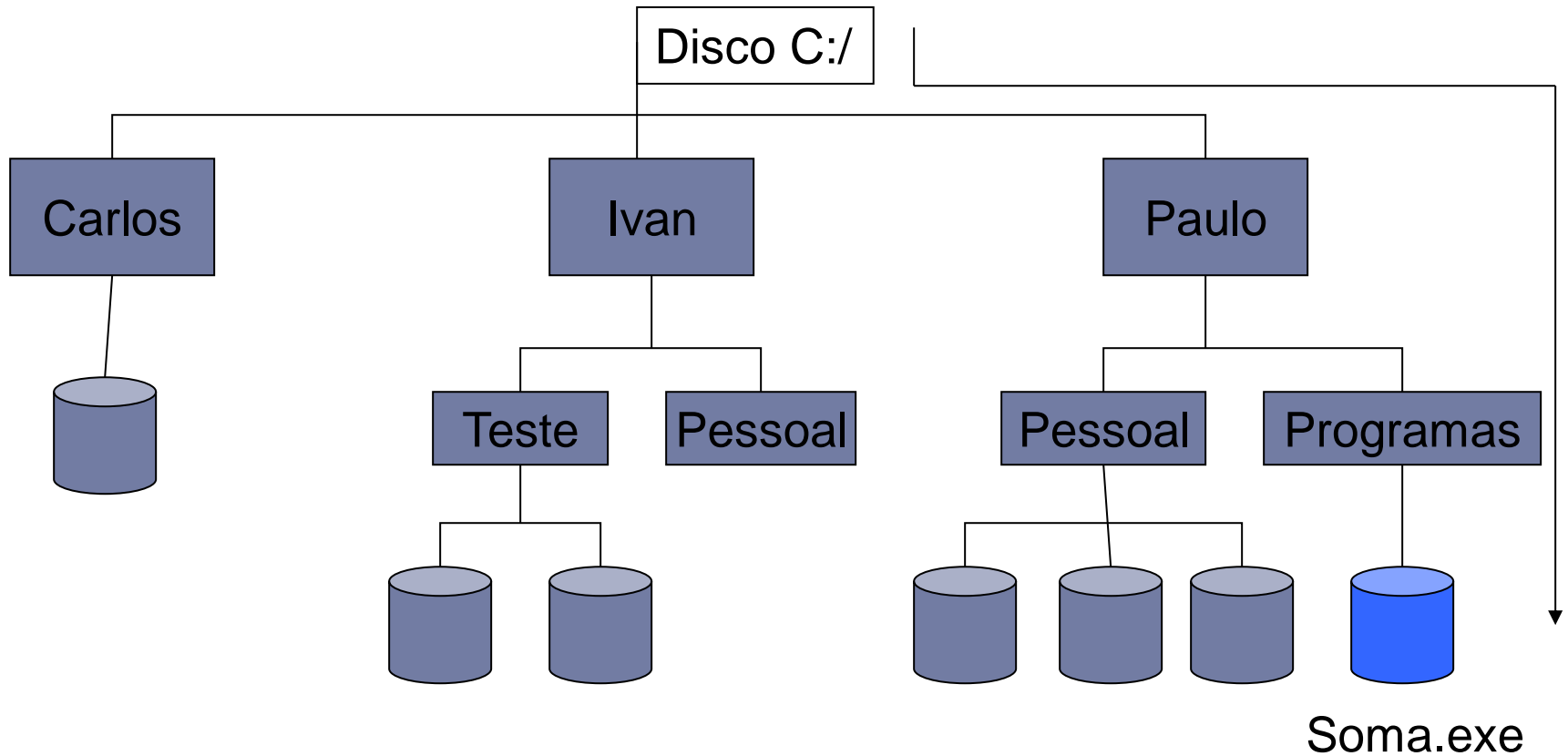
- ▶ Estrutura de diretórios em *Árvore (Tree Structured Directory)*
 - ▶ Adotado pela maioria dos sistemas operacionais e é logicamente melhor organizado
- ▶ É possível criar quantos diretórios quiser, podendo um diretório conter arquivos e outros diretórios (chamados subdiretórios)
- ▶ Cada arquivo possui um *path* único que descreve todos os diretórios da raiz (MFD) até o diretório onde o arquivo está ligado
 - ▶ na maioria dos sistemas, os diretórios são tratados como arquivos tendo atributos e identificação



Estrutura de diretórios em árvore



Path de um Arquivo



Path relativo ou absoluto ?





Sistemas Operacionais

Arquivos

Carlos Ferraz (cagf@cin.ufpe.br)

Jorge Cavalcanti Fonsêca (jcbf@cin.ufpe.br)