

# Implementação de Arquivos

Marcelo Johann

## Lembrando...

- Os dados no disco são organizados em arquivos e diretórios
  - Facilidades de accountability, de organização, de procura...
- O Sis. Op. mantém, para cada processo, uma tabela de arquivos abertos (TDAA).
  - Ponteiros para descritores de arquivos.
- Disponibiliza ao usuário chamadas de sistema para escrever/ler o arquivo
  - Read, write, ...

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 2

## Plano da aula de hoje

- Implementação de arquivos
  - Como gerenciar os blocos que constituem o arquivo?
  - Listas, índices (FAT), i-nodes...
- Exemplos de sistemas de arquivos:
  - S5fs/UNIX
  - FAT
  - NTFS

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 3

## Do ponto de vista do usuário ao ponto de vista do implementador....

- Problema principal: saber quais blocos do disco estão sendo usados por um dado arquivo.
  - É preciso de mecanismos de armazenamento dos blocos.
- Também é preciso manter as informações de cada arquivo de uma forma consistente no sistema todo.
  - Organização do sistema de arquivos.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 4

## Alocação contígua

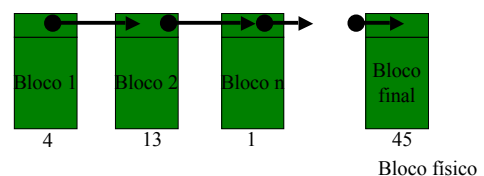
- Um arquivo que usa K blocos vai ser implementado com K blocos contíguos no disco.
  - Obs: "contíguos" no sentido do Sis. Op.
- **Vantagens:** simplicidade e desempenho.
  - Basta armazenar o tamanho do arquivo e um ponteiro sobre o 1o bloco.
  - Simples de carregar de uma vez vários blocos do arquivo.
- **Problemas:**
  - Fragmentação externa;
  - Necessidade de conhecer o tamanho (máximo) do arquivo.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 5

## Alocação por listas encadeadas

- Os blocos formam uma lista encadeada.



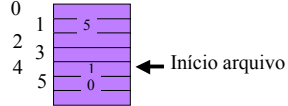
- Sem fragmentação externa;
- Basta armazenar um ponteiro sobre o 1o bloco;
- O tamanho do arquivo pode crescer a vontade.
- Problema: acesso randômico complicado!
  - E o ponteiro ocupa espaço no bloco.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 6

## Lista com índice

- Pode-se usar um índice: uma tabela de endereços, que ficará na memória.



- **Vantagens:**
  - Sem perda de espaço (não tem ponteiros);
  - Busca rápida, pois tudo está na memória (acesso randômico simples!).
- **Desvantagem:** toda a tabela deve caber na memória...
  - Exemplo: 500 MBytes, bloco de 1 KB, 500.000 entradas de 4 bytes, 2 MBytes de tabela!

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 7

## Lista com índice - FAT

- É este esquema que foi usado em MS-DOS.
- Chamado FAT (File Allocation Table).
  - Cada entrada na FAT aponta para a próxima entrada.
  - FAT-xxx usa xxx bits para endereçar os blocos
    - FAT-12, FAT-16 ( $2^{16} \times 64 \text{ KB} = 4 \text{ GB}$ )
    - FAT-32 a partir de Windows-95+
- Tamanho proporcional à capacidade do disco.
- Para aliviar a memória, os blocos podem ser aumentados (e.g. 32 KB) em discos grandes.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 8

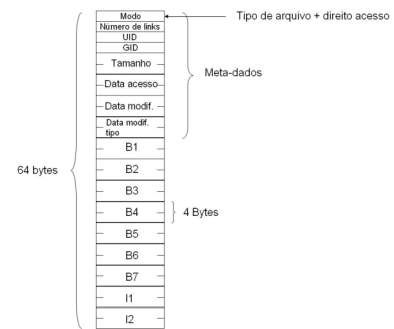
## I-node

- Solução para diminuir o tamanho da tabela: passar a usar uma tabela multi-nível.
  - O **i-node** é a solução empregada no Unix/Linux.
  - I-node = Index Node
- O i-node aponta para:
  - atributos
  - Blocos (usados para pequenos arquivos) – por exemplo, 10.
  - Tabelas de blocos (indireção simples)
  - Tabelas de tabelas de blocos
  - Tabelas de .... De tabelas de blocos (indireção triplíce)
- Muito eficiente para pequenos arquivos, flexível para grandes arquivos.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 9

## O I-node MINIX



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 10

## Quantos blocos são acessíveis por um i-nodo?

- 10 blocos diretamente acessados no i-nodo;
- 11a entrada aponta para um bloco que contém endereços de blocos
  - Se um bloco contiver 1 KB e
  - se um endereço de bloco usar 4 Bytes,
  - então são 256 blocos acessíveis através de um bloco.
- 12a entrada aponta para um bloco de bloco:
  - São  $256 \times 256$  (65536) blocos acessíveis pela 12a entrada.
- 13a entrada:
  - São  $256 \times 256 \times 256$  blocos acessíveis pela 13a entrada (16 Milhões).

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 11

## Atributos do I-nodo

- O i-node inclui vários atributos:
  - Tipo de arquivo
    - Diretório, link, E/S, ...
  - Direitos de acesso
    - Leitura, escrita, execução
  - Tamanho do arquivo em bytes
    - Não em blocos!
  - Hora de acesso / criação

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 12

## Sistema de arquivos

- Os arquivos (e diretórios) são organizados em sistema de arquivos.
  - FAT, NTFS, extfs, ext2fs, ext3fs, Berkeley FFS, minix, iso9660...
- Além de descritores de arquivos, o sistema de arquivos mantém **meta-dados** sobre todos os arquivos
  - Tamanho,
  - Log,
  - Acessos,
  - Blocos disponíveis...
- O Sis. Op. deve usar uma estrutura de dados para organizar os dados.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 13

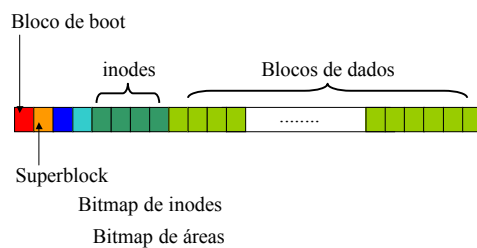
## Sistemas de arquivos Unix-like

- Existe um **superblock** que mantém os meta-dados.
  - Lido ao carregar o sistema de arquivos
  - Guardado na memória
  - Contém basicamente o tamanho das estruturas de dados.
- Conteúdo:
  - Tamanho (em blocos) do sistema de arquivos
  - Tamanho (em blocos) da lista de i-nodos
  - Número de blocos disponíveis e de i-nodos.
  - Lista de blocos e i-nodos disponíveis.
- Mantém listas parciais na RAM, para não sobrecarregar o superbloco.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 14

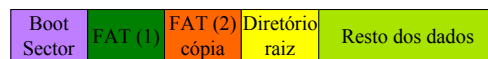
## Implementação do filesystem Unix



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 15

## O sistema de arquivos FAT (MSDOS)



- **Setor de boot**: o setor que contém a primeira imagem a ser carregada ao boot.
- **FAT (1) e FAT (2)**: tabela de alocação dos arquivos, espelhada.
  - (obs: a estrutura de diretório contém os ponteiros para os blocos iniciais dos arquivos)
- **Diretório raiz**: ponto de entrada
- **Resto**: espaço para outros diretórios e arquivos

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 16

## Sistema NTFS (Windows 2000+)

- **Objetivos principais**:
  - Confiabilidade e tolerância a falhas.
    - Garante a manutenção dos metadados.
    - logging
  - Eficiência
    - Links, compressão, arquivos esparsos...
- Organiza os discos em volumes (partição lógica);
- O cluster (bloco) varia ao formatar o volume
  - de 1 até 2<sup>n</sup> setores do disco
  - 512 Bytes, ... Alguns Kbytes.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 17

## NTFS, MFT e metadados

- **MFT = Master File Table**
  - É um arquivo especial
  - Tabela de registros de 1 KB
  - 1 registro por arquivo no volume
    - Inclui 1 registro para o arquivo que implementa a própria MFT.
- Além da MFT, NTFS define uma série de arquivos de metadados.
  - Exemplo: meta-arquivo de log.
- O primeiro registro da MFT contém o setor de boot, usado para carregar o sistema de arquivos na memória.
  - Espelhado no meio da MFT, caso haja um problema.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 18

## Registros da MFT

- Registros de metadados:
  - Diretório raiz (I)
    - Este registro contém a lista dos arquivos / diretórios encontrados na raiz;
  - Alocação dos blocos do volume
    - Bitmap
  - Blocos danificados
  - ...
- Registro de arquivo:
  - Sequência de pares (nome atributo, valor)
  - Alguns atributos: nome arquivo, informações de acesso, dados...
  - Registro residente: para pequenos arquivos, todos os atributos cabem no registro da MFT (1 KB)
  - Registro estendido: blocos fora da MFT são usados para armazenar parte dos "grandes" atributos (e.g. dados).

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 19

## Os registros da MFT

0	MFT	} Reservados aos Meta-dados predefinidos
1	Espelho MFT	
2	Arquivo de log	
3	Arquivo do volume	
4	Tabela de def. atributos	
5	Raiz	
6	Bitmap de clusters livres	
7	Setor de boot	
8	Setores estragados	
9	Clusters estragados	
10	Segurança	
11	Dir. meta-dados estendidos	
12	Não usado...	
16	Arquivo/diretório usuário	

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 20

## Seguindo...

### Implementação de Diretórios

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 21

## Plano

- Implementação de diretórios:
  - Do ponto de vista do usuário:
    - Organização dos diretórios;
    - Links e ciclos na organização.
  - Como implementar a estrutura de dados para gerenciar diretórios?
    - Estabelecer um relacionamento entre nome e blocos, entre meta-dados e nome.
    - Listas, tabelas de hash, B-trees...
  - Casos de estudo:
    - MS-DOS, Unix, NTFS.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 22

## Diretório & Arquivos

- O arquivo é a unidade básica de dados (do ponto de vista do usuário)
- O diretório é a unidade básica para organizar os arquivos.
  - Em geral, implementado como sendo um arquivo "especial".
  - Chamado "pasta" em alguns sistemas...
- O diretório deve associar, para cada arquivo/diretório nele contido:
  - Um nome definido pelo usuário;
  - Informações de contabilidade;
  - Um ponteiro sobre o arquivo.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 23

## Operações com diretórios

- Criação
  - Deve-se acrescentar uma entrada na estrutura de dados "diretório" para incluir o novo diretório/arquivo.
  - Possivelmente, deve-se re-organizar as entradas.
- Remoção
  - Deve-se tirar uma entrada;
    - Obs: problema potencial ao remover um diretório inteiro! Fazer o que se não está vazio?
  - Possivelmente, deve-se re-organizar as entradas.
- Acesso/listagem
  - Deve retornar a lista das entradas;
  - Para tanto, precisa ser "aberto".

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 24

## Diretório de um nível

- Todos os arquivos estão em **um diretório** só, no sistema todo.
- Ausência total de controle de acesso...
  - Por exemplo, arquivos executáveis e/ou de sistema estão misturado com os dados dos usuários.
- Falta de eficiência para procurar os arquivos.
- Conflito entre os nomes dos arquivos
  - Todos os usuários devem evitar usar nomes parecidos!
- Usado em sistemas de arquivos básicos & antigos
  - Floppies.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 25

## Diretório de 2 níveis

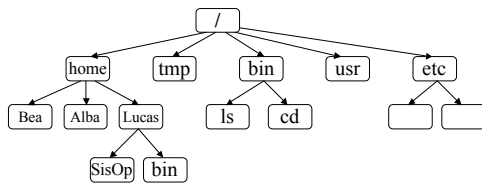
- Um primeiro nível distingue (por exemplo) contas de usuários;
  - Serve também para isolar diretórios do sistema.
- Cada usuário tem todos seus arquivos em um único diretório, pessoal.
- Pode-se compartilhar arquivos
  - É preciso prefixar o nome do arquivo pelo diretório onde se encontra.
- Pode-se compartilhar um diretório inteiro
  - Bin/ com os executáveis do sistema.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 26

## Árvore de diretórios

- Generalização do mecanismo com 2 níveis.



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 27

## Caminho (path)

- O caminho é a seqüência de diretórios a serem percorridos até chegar a um arquivo.
- Caminho **absoluto**: a partir da raiz
  - Começa com '/' (ou '\', em MS-DOS/Windows)
  - /home/nmaillard/bin/gcc
- Caminho **relativo**:
  - Diretório corrente ('.') = diretório onde o usuário está trabalhando quando dispara o processo.
    - Deve ser incluído na struct procl
  - O caminho relativo começa a partir do diretório corrente
    - bin/gcc
    - ../bea/bin/gcc
- Os diretórios '.' e '..' designam o diretório corrente e pai.
- Existe uma variável de ambiente **PATH** que indica os caminhos default onde procurar um arquivo executável.
  - Um processo mantém uma variável para seu diretório corrente

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 28

## DAG de diretórios & Links (atalhos)

- O **link** (atalho) possibilita o compartilhamento de um arquivo e/ou diretório.
  - Não há cópia física dos dados!
    - Possibilita a manutenção simples da coerência
  - Problema: pode criar laços!!
    - Problema para a procura.
    - Problema potencial na hora de deletar arquivos.
- **Soft link**:
  - Implementado através de um ponteiro sobre uma entrada física
  - Possibilita detectar ciclos, por isso lícito com diretórios
- **Hard link**
  - Implementado através de uma cópia física de uma entrada.
  - Impossibilita detecção de ciclos, proibido com diretórios.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 29

## Implementação de diretório

- O diretório provê o acesso aos arquivos e outros diretórios nele contidos.
  - Deve mapear o nome simbólico com a estrutura de dados que implementa o arquivo
    - Arquivo todo, endereço do 1o bloco, inode...
  - Também pode haver mapeamento para um outro diretório.
- O diretório será implementado como uma **seqüência de entradas**, cada qual descrevendo um dos diretórios/arquivos nele contido.
  - Várias soluções para implementar a seqüência

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 30

## Possibilidades de implementação

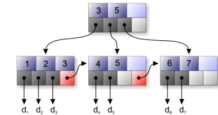
- Lista linear de entradas
  - Pesquisa linear... :o(
    - Cada inserção necessita uma procura para evitar nomes repetidos!
  - Simples :o)
  - Uma forma de otimizar ordenar a lista
    - Sobrecusto à inserção
- Tabela de Hash
  - Lista linear + índice
  - Deve-se gerenciar as colisões!

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 31

## Implementação com Árvores B (B-trees)

- Estrutura de dados para agilizar as procuras.
  - Altura  $\log(n)$ , com  $n$  entradas.
  - Usado para dicionários
- Baseado em chaves, usadas para definir:
  - O número de filhos de cada nó
    - Um filho / intervalo de valores da chave
  - Todas as folhas estão na mesma altura



- Pesquisa parecida à árvore binária
  - Compara-se as chaves para orientar a procura.
- Inserção de chave mais complicada!
  - Divide-se um nó folha, e migra-se a chave para cima.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 32

## Exemplo de implementação de entrada

- Implementa poucas informações
  - Nome + tipo (extensão): foo.txt
  - Identificador do usuário dono do diretório/arquivo
- Exemplo simples: CP/M (1986)
  - 1 diretório único
  - Endereços dos Blocos dos arquivos armazenados também na estrutura do diretório
  - Uma entrada do diretório é da forma seguinte (32 Bytes):

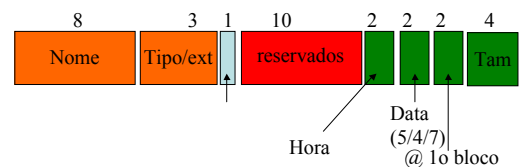


INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 33

## Implementação no MS-DOS

- Árvore de diretórios!
- Cada entrada tem comprimento 32 Bytes
  - Nome, extensão...
  - Ponteiro para o primeiro bloco na FAT.
  - Os outros blocos do arquivo se encontram graças à FAT.
  - Se o bloco apontado na FAT contiver a mesma estrutura, aninham-se os diretórios.

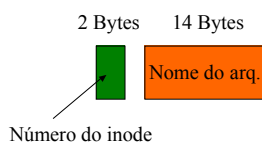


INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 34

## Implementação no Unix

- Muito simples, devido ao uso de inodes.
  - As informações sobre o arquivo/diretório estão armazenadas no inode (atributos):
    - Nome, extensão, data/hora de criação...



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 35

## Procura por um arquivo

- Procura-se o arquivo /alguma/coisa/foo.txt
- Localiza a raiz / (vide entrada no sistema de arquivos!)
- É um diretório (inode, por exemplo).
  - Vai ser consultado para achar o que tem dentro.
  - Procura-se o diretório 'alguma'
    - Vai achar o inode correspondendo
    - Vai carregar o 1o bloco do disco, que implementa o diretório.
- É um diretório... Se repete o procedimento
  - Acha o inode associado a 'coisa'
  - Carrega-o do disco
- Afinal, chega-se a procurar 'foo.txt'
  - Obtém-se o inode apropriado... E os blocos de dados.
- Obs: pode ter **erro** se algum nome está errado!

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2010/2

Aula 23 : Slide 36

Próxima aula...

- **Gerenciamento de espaço livre.**