



Universidade Tecnológica Federal do Paraná

Departamento Acadêmico de Computação

Bacharelado em Ciência da Computação

Sistemas Operacionais

Sistemas de Arquivos Implementação

Prof. Rodrigo Campiolo

25/11/21

Introdução

- Sistemas de Arquivos (SA)
 - Arquitetura de Sistemas de Arquivos
 - Implementação de arquivos e diretórios
 - Gerenciamento de espaço
 - Exemplos de Sistemas de Arquivos

SA - Arquitetura

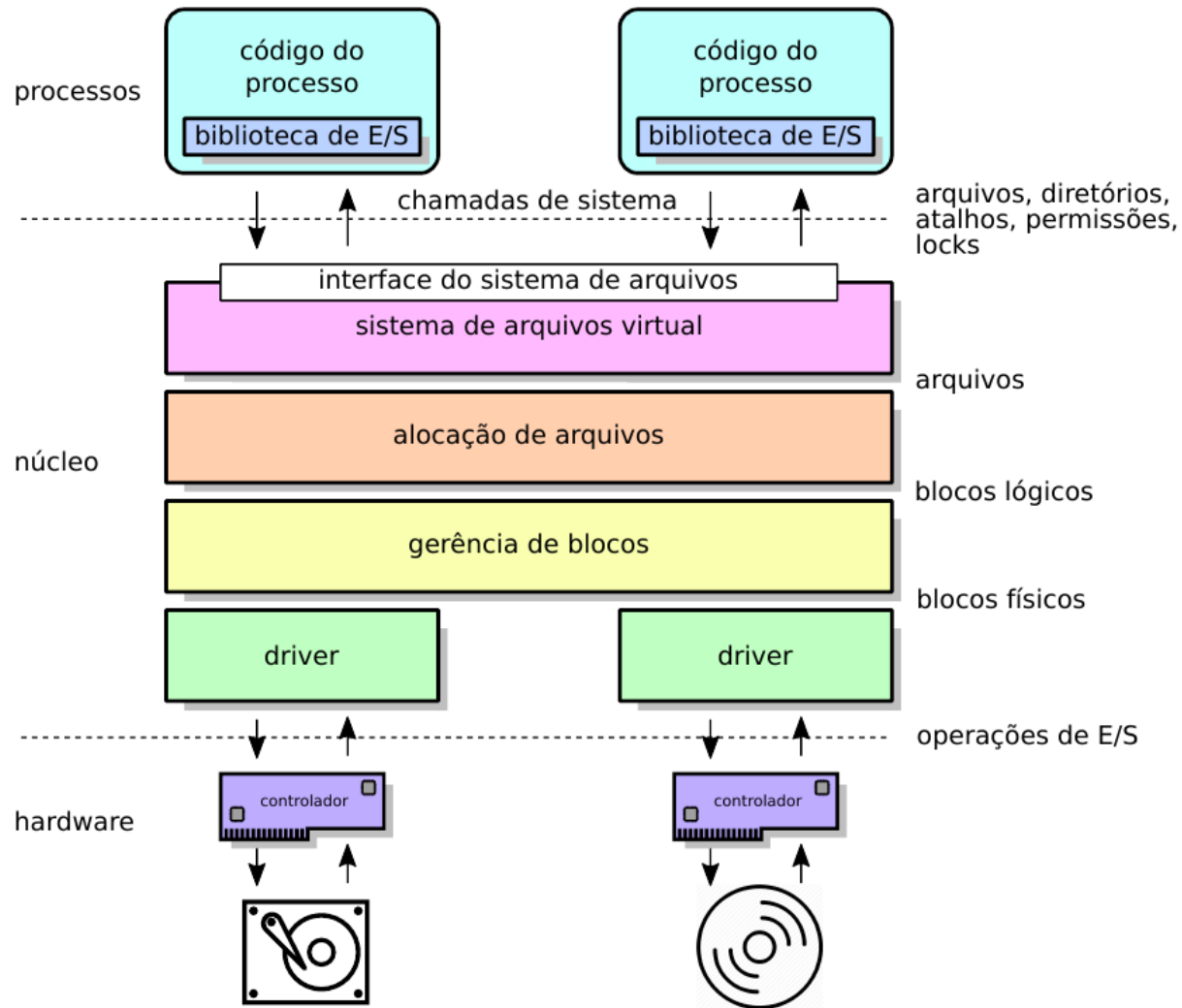


Figura: Camadas da implementação de gerência de arquivos. (Maziero)

Sistemas de Arquivos

- Discos e Partições
 - Acesso e armazenamento em blocos.
 - Discos podem ser divididos em partições.
 - A tabela de partições armazena as informações de controle da partição.
 - A área inicial do disco armazena o código de inicialização do SO e a tabela de partições (*MBR, GPT*).
 - No início de cada partição há também blocos reservados para descrição da partição e inicialização do SO.

Sistemas de Arquivos

MBR

Master Boot Record						Extended Partition		
Partition table								
Master Boot Code	1st Partition Table Entry	2nd Partition Table Entry	3rd Partition Table Entry	4th Partition Table Entry	0x55 AA	Primary Partition (C:)	Primary Partition (E:)	Primary Partition (F:)
						Logical Drive (G:)	Logical Drive (H:)	Logical Drive n

MBR:

- 64 bytes
- 4 primárias (16 bytes)
- partições estendidas
- partições 2TB (4 bytes)

GPT

Protective MBR						Primary GUID Partition Entry Array				Backup GUID Partition Entry Array			
Master Boot Code	1st Partition Table Entry	2nd Partition Table Entry	3rd Partition Table Entry	4th Partition Table Entry	0x55 AA	Primary GUID Partition Table Header	GUID Partition Entry 1	GUID Partition Entry 2	GUID Partition Entry <i>n</i>	GUID Partition Entry 128	Primary Partition (C:)	Primary Partition (E:)	Primary Partition <i>n</i>
											GUID Partition Entry 1	GUID Partition Entry 2	GUID Partition Entry <i>n</i>
											GUID Partition Entry 128	Backup GUID Partition Table Header	

GPT:

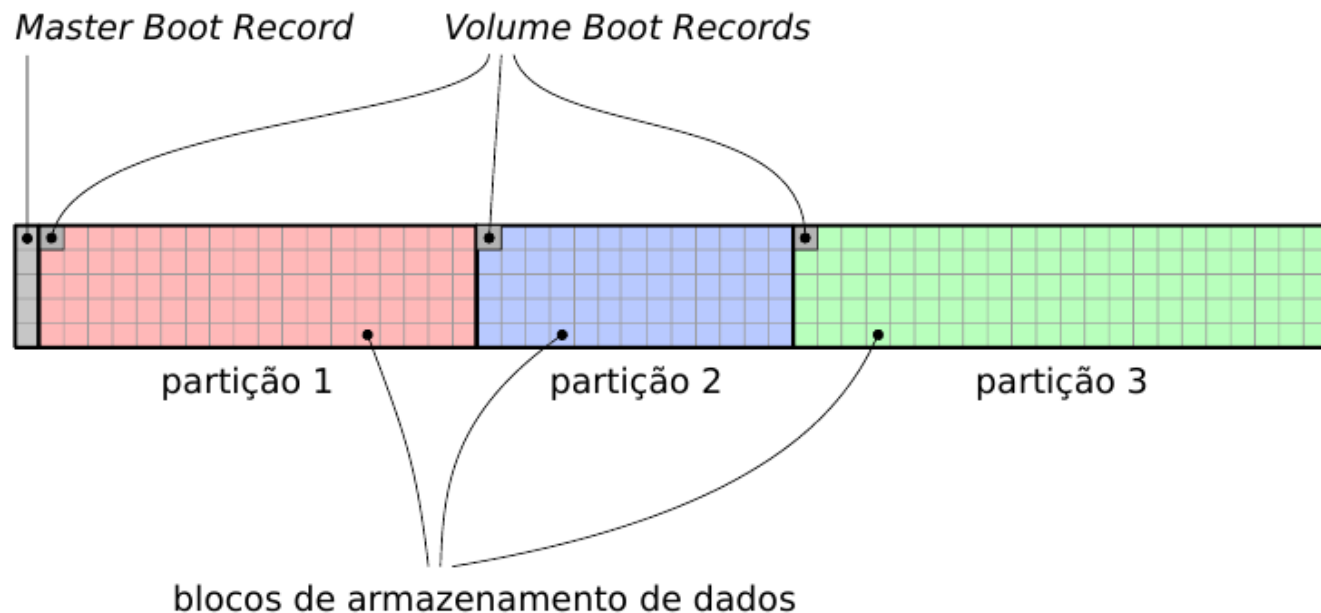
- 16 KB
- 128 primárias (128 bytes)
- partições ~9ZB (8 bytes)
- redundância

Figura: Master Boot Record x GUID Partition Table

Fonte: <http://www.mustbegeek.com/difference-between-mbr-and-gpt/>

Sistemas de Arquivos

- Discos e Partições
 - SA independentes por partição.
 - Formatação do sistema de arquivos.
 - Partições inicializáveis e **volume**.



Sistemas de Arquivos

■ Partições e Sistemas de Arquivos

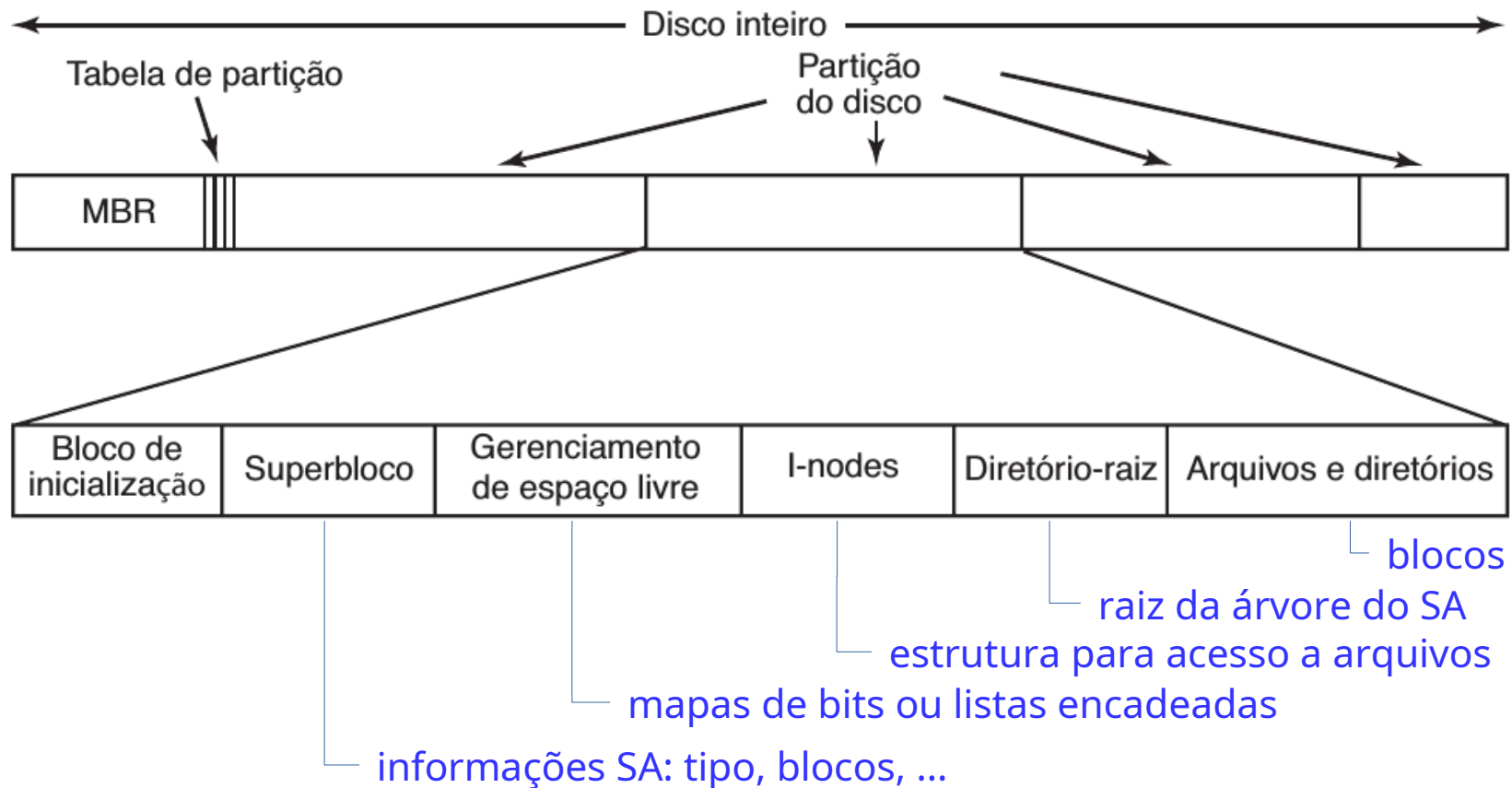


Figura: Uma possível configuração para um sistema de arquivos (Tanenbaum)

Sistemas de Arquivos

- Gerência de Blocos
 - Tipos de blocos
 - Blocos físicos (512 ou 4096 bytes)
 - Blocos lógicos (4 KiB a 64 KiB)
 - Tamanho dos blocos
 - Pequenos (desempenho)
 - Grandes (fragmentação interna)

Sistemas de Arquivos

- Gerência de Blocos
 - Otimização de leitura
 - **Read-through**: consulta cache para localizar o bloco.
 - **Read-ahead**: lê mais blocos que solicitado e armazena na cache.
 - Otimização de escrita
 - **Write-through**: escrita síncrona.
 - **Write-back**: dados são armazenados em cache para futura escrita.

Sistemas de Arquivos

- Gerência de Espaço Livre
 - Mapa de bits
 - Manter um mapa de bits em uma área reservada do volume.
 - Cada bit representa um bloco livre ou ocupado.
 - Simples de implementar e compacto.
 - Disco 1TiB, blocos 4KiB, mapa de bits = 32MiB

Sistemas de Arquivos

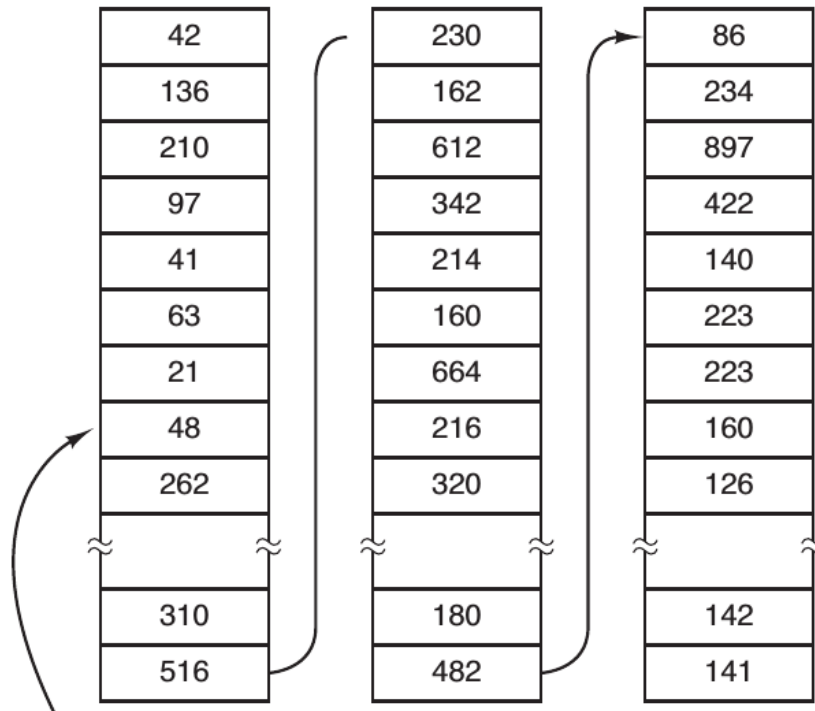
- Gerência de Espaço Livre
 - Lista encadeada
 - Armazenar ponteiros para blocos livres.
 - O último ponteiro apontaria para outro bloco de ponteiros.
 - Obter um grande número de blocos livres a cada acesso ao disco.
 - Blocos 1KiB, número blocos 32 bits, 1 bloco contém 255 blocos livres.
 - Uma alternativa seria uma **tabela de grupos de blocos livres** com entradas [bloco, tamanho].

Sistemas de Arquivos

■ Gerência de Espaço Livre

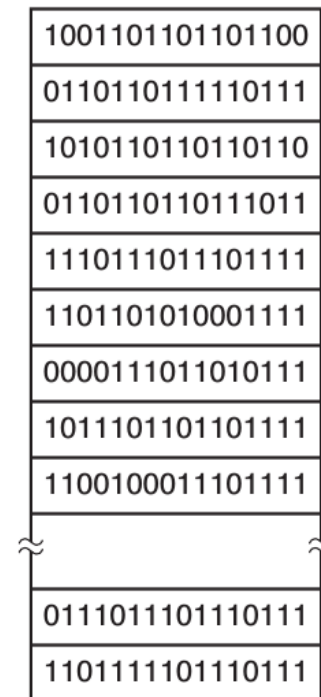
(a) Armazenamento da lista de blocos livres em uma lista encadeada. (b) Um mapa de bits.

Blocos livres de disco: 16, 17, 18



Um bloco de disco de 1 KB pode conter 256 números de blocos de disco de 32 bits

(a)



Um mapa de bits

(b)

SA – Implementação de Arquivos

- Para manipular arquivos, precisa-se de estruturas de dados.
- O **Descritor de Arquivos** mantém informações de um arquivo.
- Duas estruturas são usadas para manter informações dos descritores de arquivos abertos:
 - Tabela de Descritores de Arquivos Abertos (TDAA) – *Global File Table*
 - Tabela de Arquivos Abertos por Processo (TAAP) – *File Table*

SA – Implementação de Arquivos

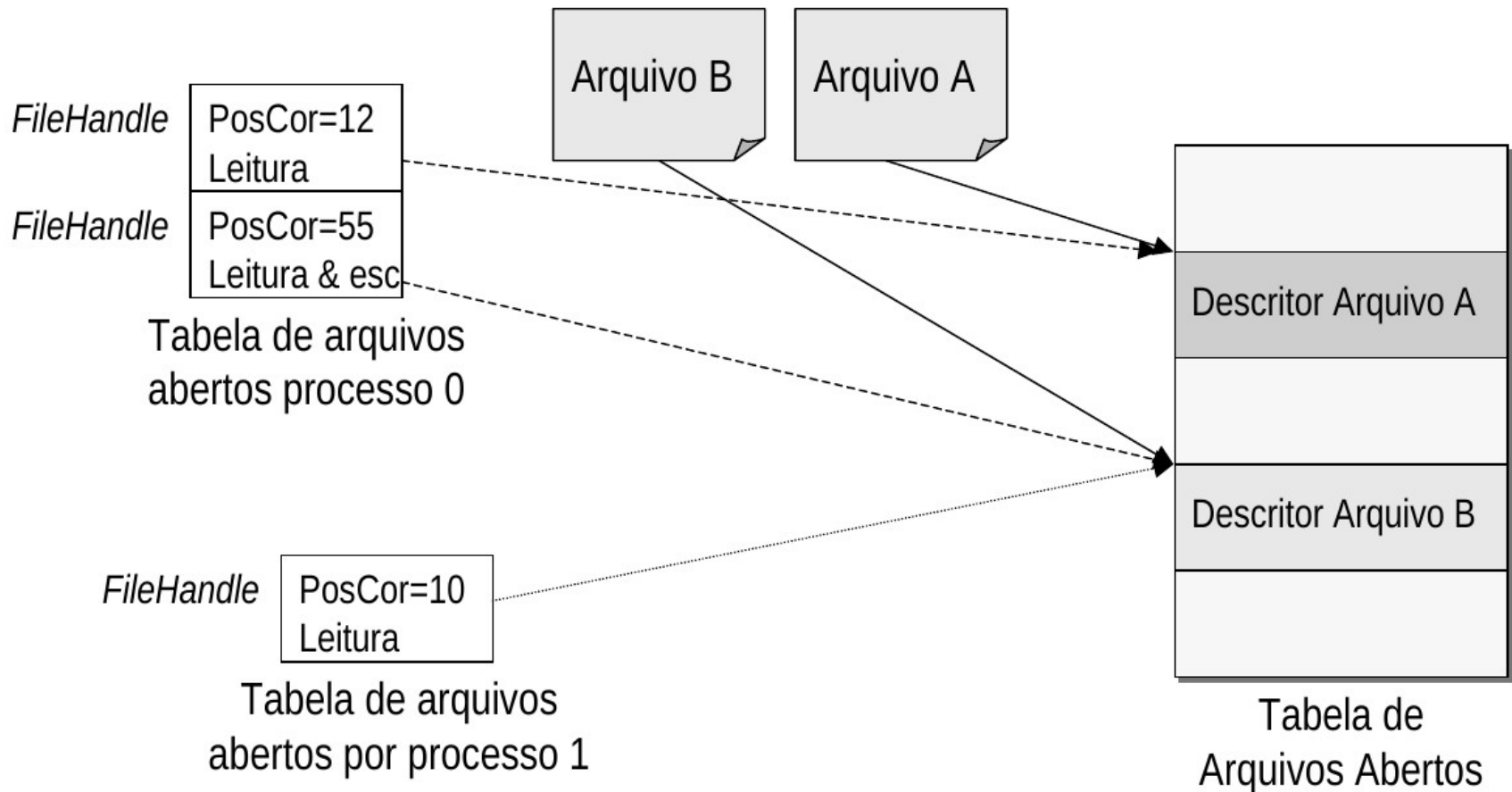


Figura: TAAP (processos) e TDAA (núcleo) (Oliveira).

SA – Alocação de Arquivos

- Espaço de armazenamento dividido em blocos lógicos fixos.
- **Alocação de Arquivos:** alocar os **metadados** e conteúdos dos arquivos dentro dos blocos lógicos.
- Desempenho é essencial:
 - **Rapidez:** acessos sequenciais e aleatórios.
 - **Robustez:** quanto a erros, blocos defeituosos, dados corrompidos.
 - **Flexibilidade:** criação, alteração e exclusão de arquivos.

SA – Alocação de Arquivos

- Bloco de Controle de Arquivo
 - *File Control Block* (FCB)
 - Estrutura para armazenar os metadados e referências para os blocos do arquivo.
- Implementação
 - Entrada no diretório.
 - Estrutura independente:
 - *Master File Table* (MFT)
 - i-nodes

SA – Alocação de Arquivos

- Alocação Contígua
 - Arquivos armazenados em um conjunto de blocos contíguos no disco.
 - Vantagens
 - simples de implementar [bloco inicial e tamanho]; alto desempenho para leitura.
 - Desvantagens
 - escrita; fragmentação externa.
 - Exemplo:
 - Sistema ISO 9660 para CD-ROM.

SA – Alocação de Arquivos

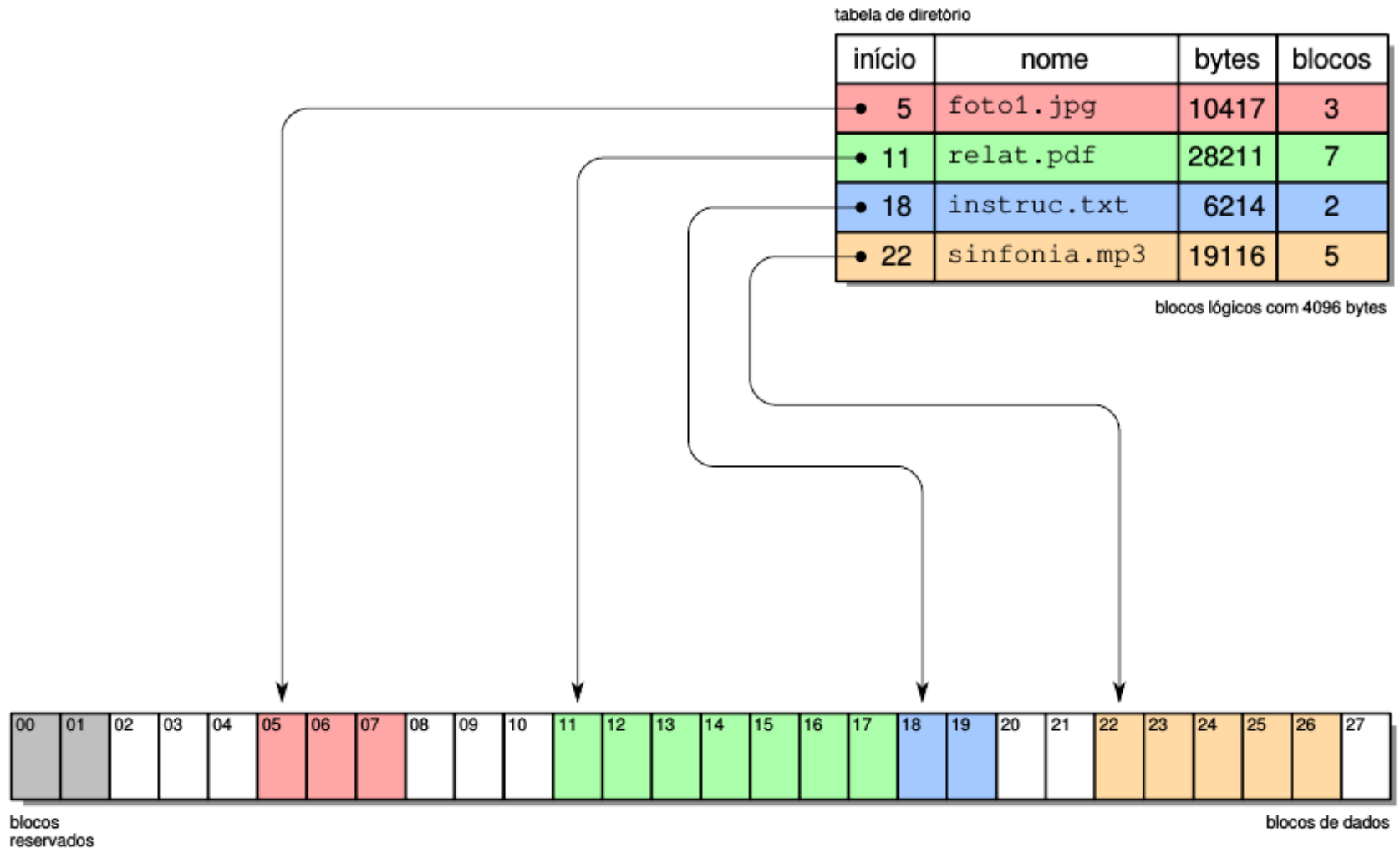


Figura: Estratégia de alocação contígua (Maziero).

SA – Alocação de Arquivos

- Alocação Encadeada Simples
 - Bloco do arquivo contém dados e ponteiro para o próximo bloco.
 - Vantagens
 - Elimina fragmentação externa; não conhecimento prévio do tamanho do arquivo.
 - Desvantagens
 - Acesso aleatório.
 - Problema de robustez: bloco defeituoso.
 - Tamanho de dados dos blocos diminuído (ponteiro).

SA – Alocação de Arquivos

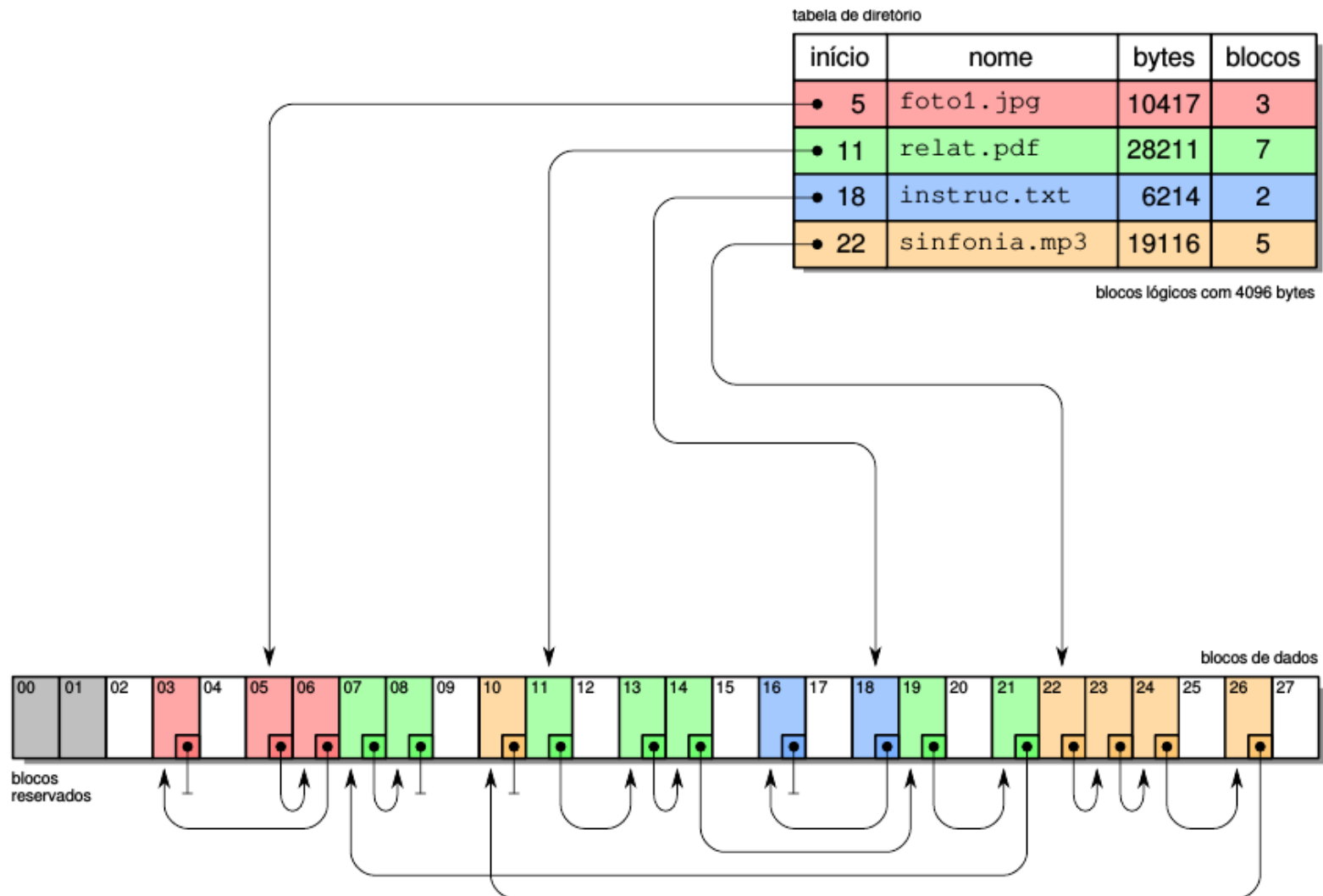


Figura: Estratégia de alocação encadeada simples (Maziero).

SA – Alocação de Arquivos

- Alocação Encadeada FAT
 - Os ponteiros para os blocos de arquivos são mantidos em uma única tabela denominada **Tabela de Alocação de Arquivos – *File Allocation Table* (FAT)**
 - A tabela é armazenada em blocos reservados na partição e carregada inteira na memória principal.
 - Cada entrada corresponde a um bloco lógico e contém ponteiro para o próximo bloco do arquivo. Pode conter *tags* para blocos livres, defeituosos e reservados.
 - FAT geralmente possui uma cópia de segurança.
 - Usada no MS-DOS, Windows e em sistema de arquivos para dispositivos de armazenamento portáteis.

SA – Alocação de Arquivos

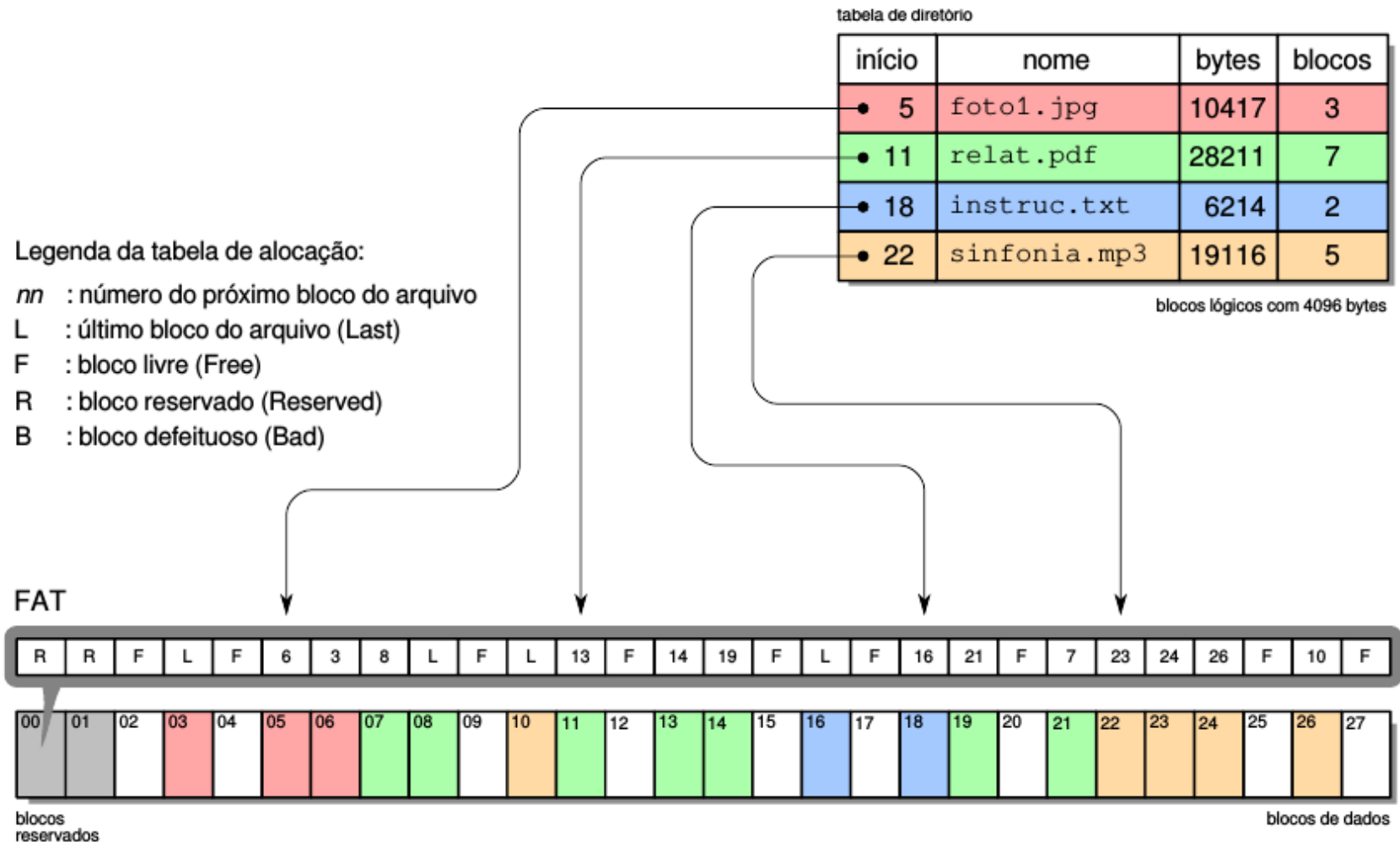


Figura: Estratégia de alocação encadeada com FAT (Maziero).

SA – Alocação de Arquivos

- Alocação Indexada Simples
 - Associar a cada arquivo uma estrutura de dados que armazena os atributos e os índices de blocos de dados do arquivo.
 - A estrutura é denominada de **i-node** (*index-node*).
 - Uma tabela de *i-nodes* em uma área reservada armazena todos os *i-nodes*.
 - Desvantagens
 - Tamanho fixo dos *i-nodes* e da tabela: limitam tamanho de um arquivo e número máximo de arquivos.

SA – Alocação de Arquivos

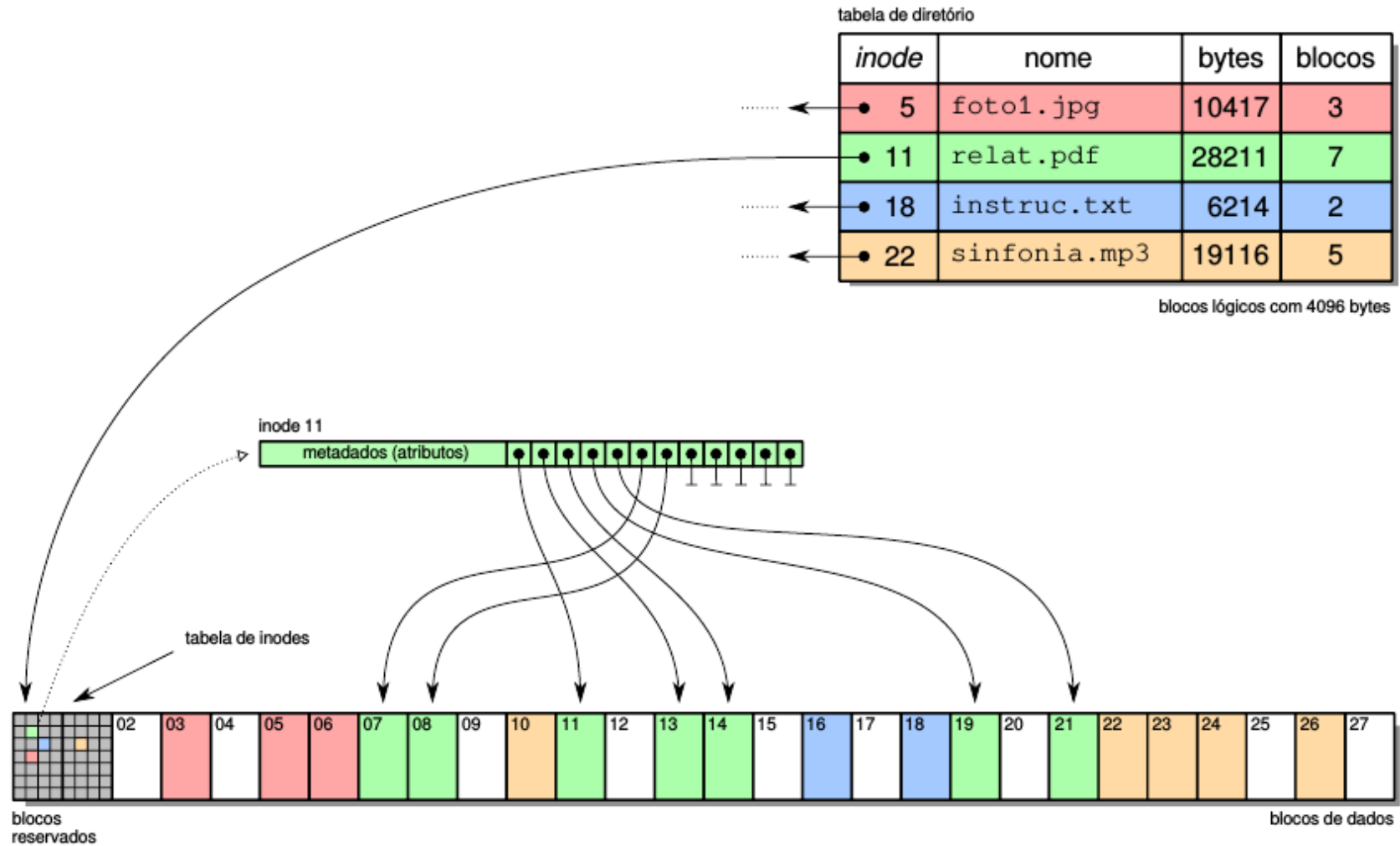


Figura: Estratégia de alocação indexada simples (Maziero).

SA – Alocação de Arquivos

- Alocação Indexada Multinível
 - Usar entradas dos *i-nodes* como ponteiros indiretos, isto é, para novos *blocos de índices*.
 - No ext2/3/4 (usado no Linux) temos:
 - 12 ponteiros diretos (dados)
 - 1 ponteiro indireto
 - 1 ponteiro duplamente indireto
 - 1 ponteiro triplamente indireto
 - Considere o ext3 com blocos de 4KiB e ponteiros de 4bytes, logo cada bloco contém 1024 ponteiros. Qual o tamanho máximo de um arquivo?

Figura: Alocação em Nível (Maziero)

Figura: Alocação indexada multi-Nível (Maziero).

SA – Implementação de Diretórios

- Implementado como um arquivo que o conteúdo é uma relação de entradas (arquivos, diretórios, atalhos, entre outros).
- Cada entrada deve possuir ao menos: nome do arquivo, tipo, localização física (*i-node*, bloco inicial).
- Os atributos podem estar na entrada ou na estrutura (*i-nodes*).
- A lista ou índice do diretório pode ser implementada como uma lista simples ou com estruturas de dados mais complexas (*hash*, árvores).

SA – Implementação de Diretórios

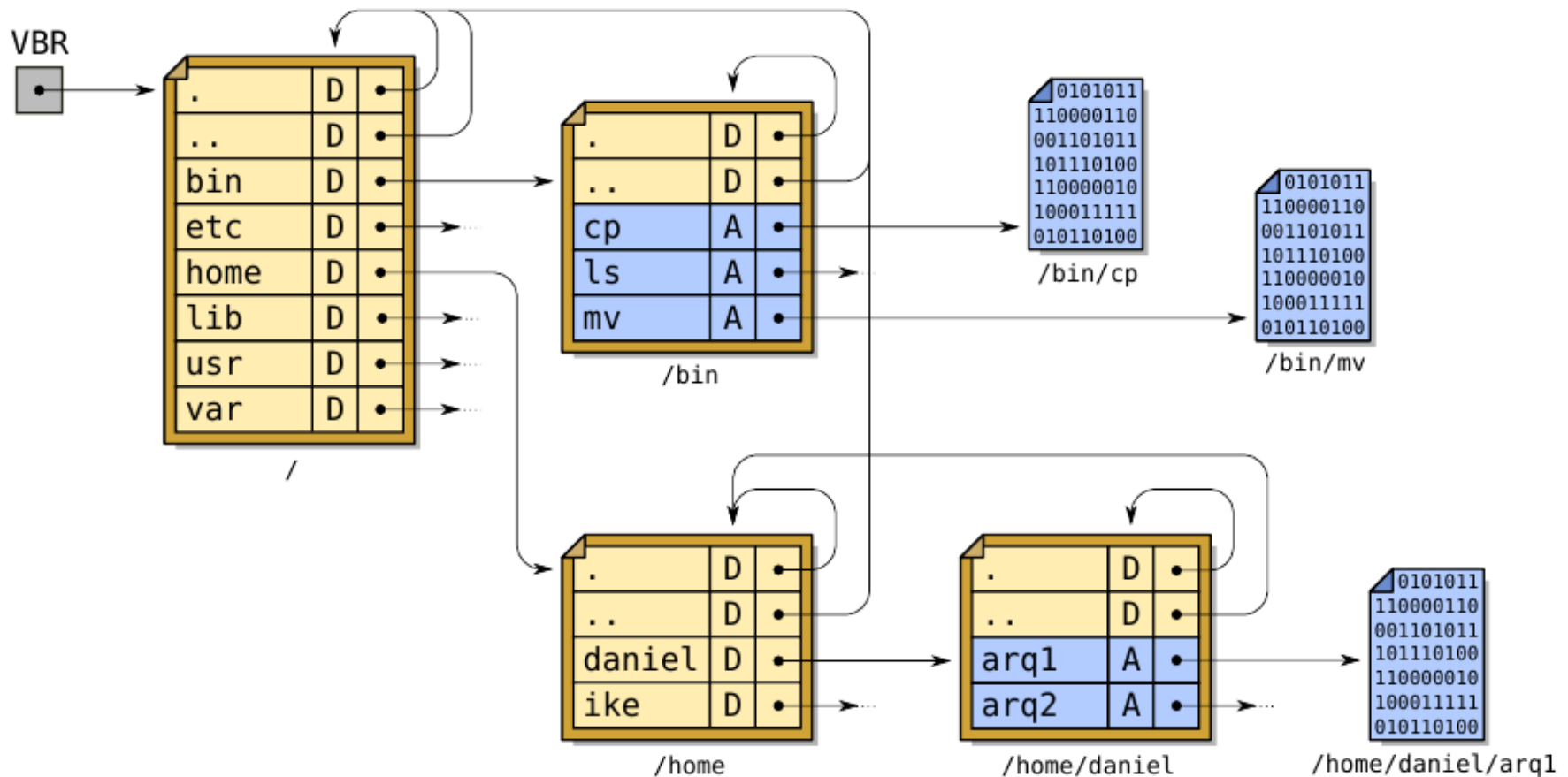


Figura: Implementação de uma estrutura de diretórios (Maziero).

SA – Implementação de Diretórios

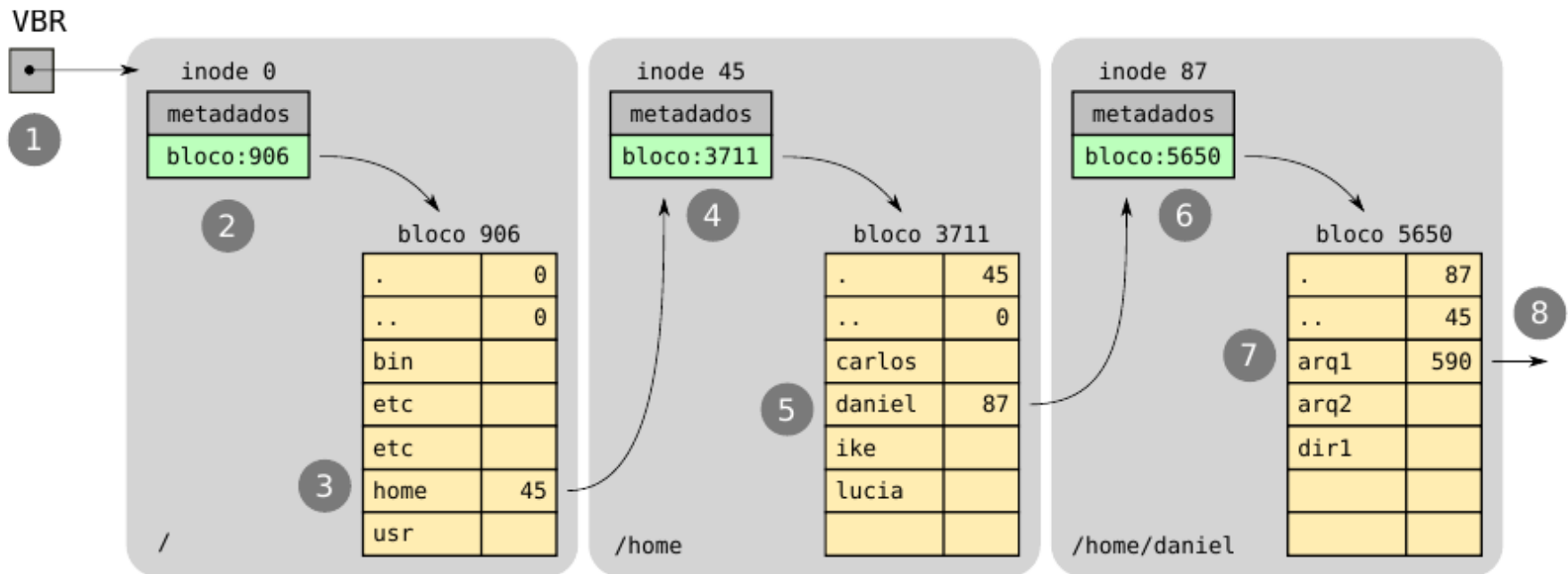


Figura: Resolução para o arquivo `/home/daniel/arq1` (Maziero).

1. localizar i-node no dir. raiz
2. ler i-node `"/"` (permissão e localização)
3. ler conteúdo i-node `"/"` para localizar `"/home"`
4. ler i-node `"/home"` (permissão e localização)

5. ler i-node `"/home"` para localizar `"/home/daniel"`
6. ler i-node `"/home/daniel"` (permissão e localização)
7. ler i-node `"/home/daniel"` para localizar `"/home/daniel/arq1"`.
8. Devolver o i-node do arquivo.

SA – Exemplos

- FAT
 - FAT (1977), FAT12 (1980), FAT16 (1984), FAT32 (1996), exFAT (2006)

FAT16



FAT32



Diretório raiz

SA – Exemplos

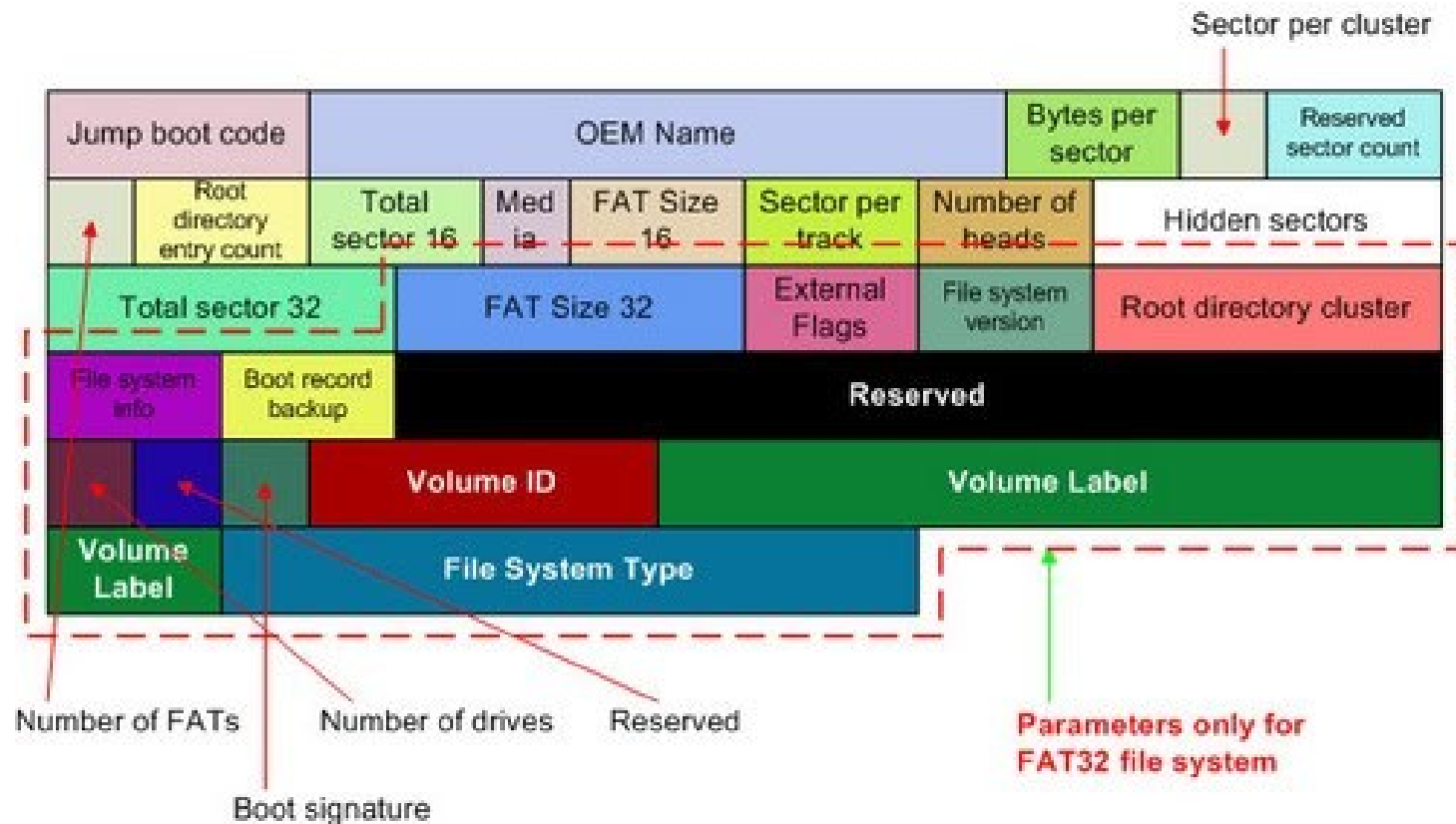


Figura: FAT32 - Estrutura do *Boot Record*.

Fonte: <https://sites.google.com/site/iprinceps/Home/embedded-system-and-operating-systems/understanding-file-system>

SA – Exemplos

Directory table entry (32B)

Filename (8B)
Extension (3B)
Attributes (1B)
Reserved (1B)
Create time (3B)
Create date (2B)
Last access date (2B)
First cluster # (MSB, 2B)
Last mod. time (2B)
Last mod. date (2B)
First cluster # (LSB, 2B)
File size (4B)

File allocation table

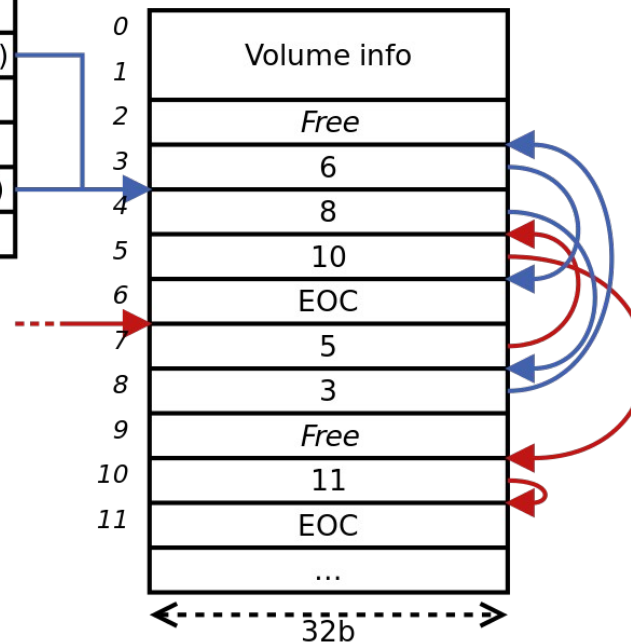


Figura: Estrutura da FAT e de uma entrada na tabela de diretórios.

Fonte: https://commons.wikimedia.org/wiki/File:Fat32_structure.svg

SA – Exemplos

- NTFS
 - *New Technology File System*
 - Julho de 1993, Windows NT 3.1
 - Diretórios, arquivos e metadados de arquivos são armazenados como metadados na MFT.
 - Provê escalabilidade, sistema de journaling, cotas, encriptação, compressão, ...
 - Arquivos pequenos podem ser armazenados diretamente na MFT.

SA – Exemplos

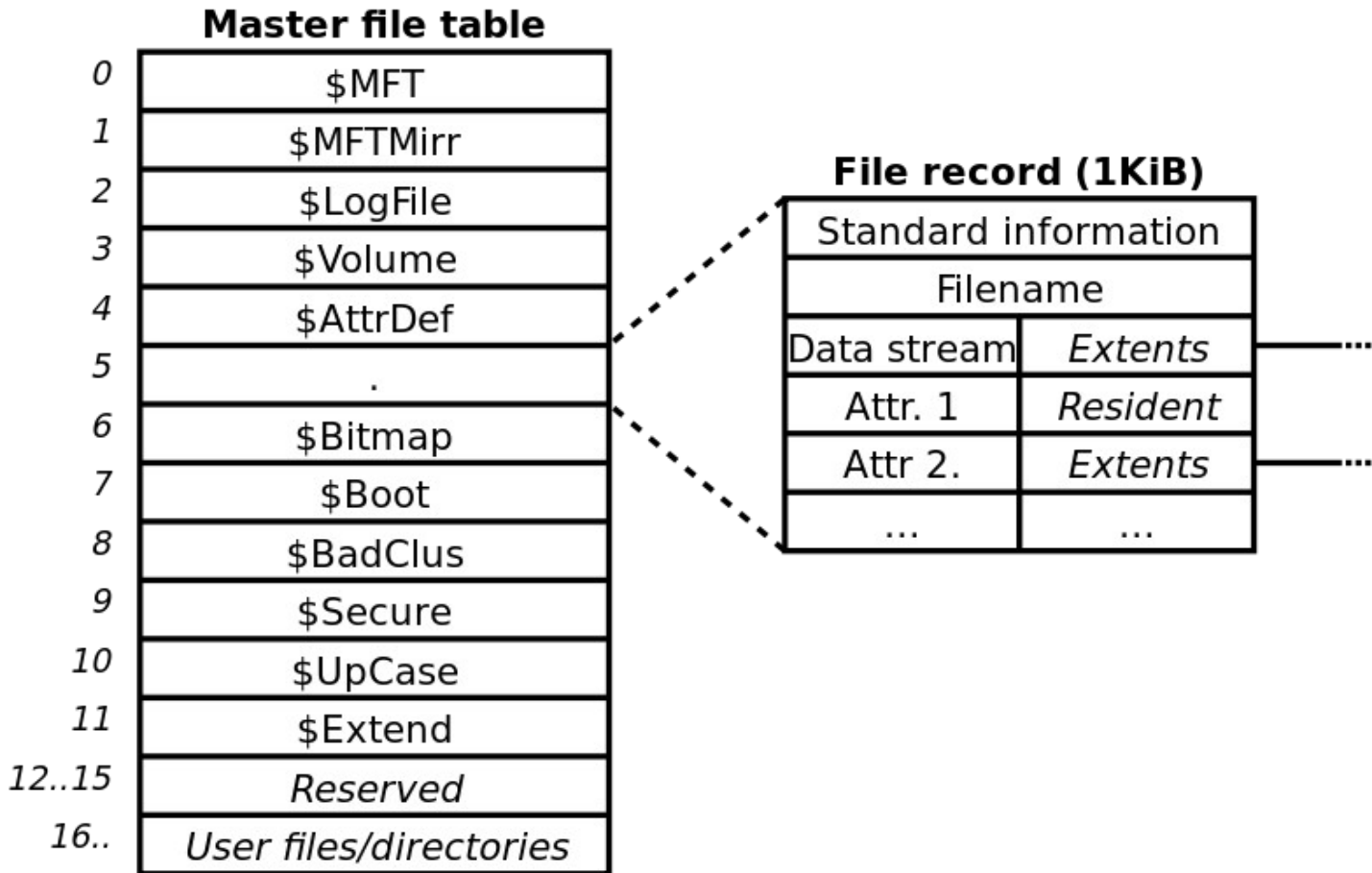


Figura: Estrutura do NTFS e da entrada na tabela de diretórios.

https://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Ntfs_mft.svg/800px-Ntfs_mft.svg.png

SA – Exemplos

- EXT4
 - ***Extended File System*** que faz uso da estrutura *inode*.
 - Versões: EXT (1992), EXT2 (1993), EXT3 (2001) e EXT4 (2006).
 - Algumas características:
 - Suporte para arquivos grandes (aumento de desempenho e diminuição fragmentação).
 - Sistema com *journaling* e com *checksum* para o *journal*.
 - Checagem rápida do SA.
 - Uso de *extents* (sequências de blocos).

SA – Exemplos

- EXT4
 - Dividido em grupos de blocos
 - Blocos: 4KiB
 - Grupos de blocos: 32K blocos
 - Tamanho do grupos: 128 MiB

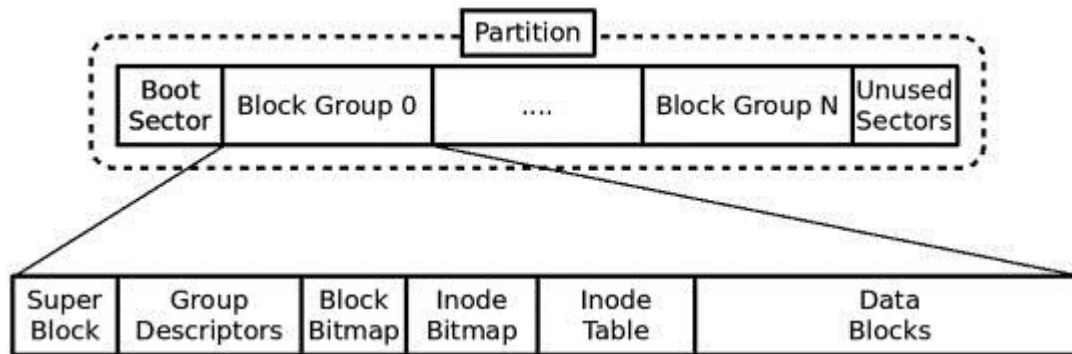


Figura: EXT4 – Estrutura, grupos de blocos e detalhes do grupo 0.

https://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Ntfs_mft.svg/800px-Ntfs_mft.svg.png

SA - Ferramentas Linux

- Informações de sistema de arquivos
 - `tune2fs -l /dev/sda1`
 - `fdisk -l`
 - `lsblk`
 - `parted -l`
- Listar inodes de arquivos
 - `ls -li`
- Informações de arquivo.
 - `stat filename.abc`

Atividades

Referências

- Maziero