

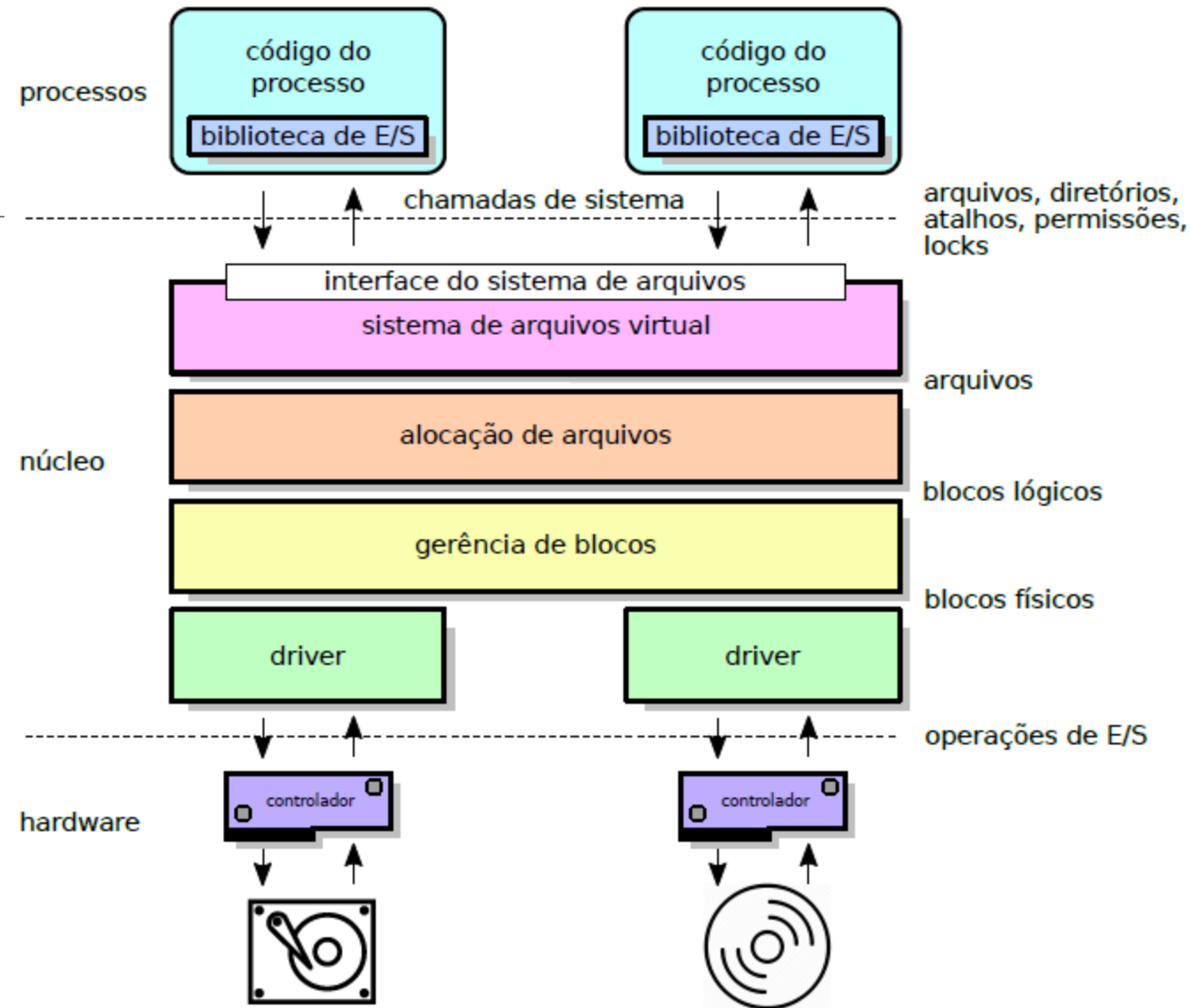
Sistemas de Arquivos

PROFESSOR: Jheymesson A. Cavalcanti

DISCIPLINA: SISTEMAS OPERACIONAIS II

1. Introdução

Os principais elementos que realizam a implementação de arquivos no sistema operacional estão organizados em camadas:



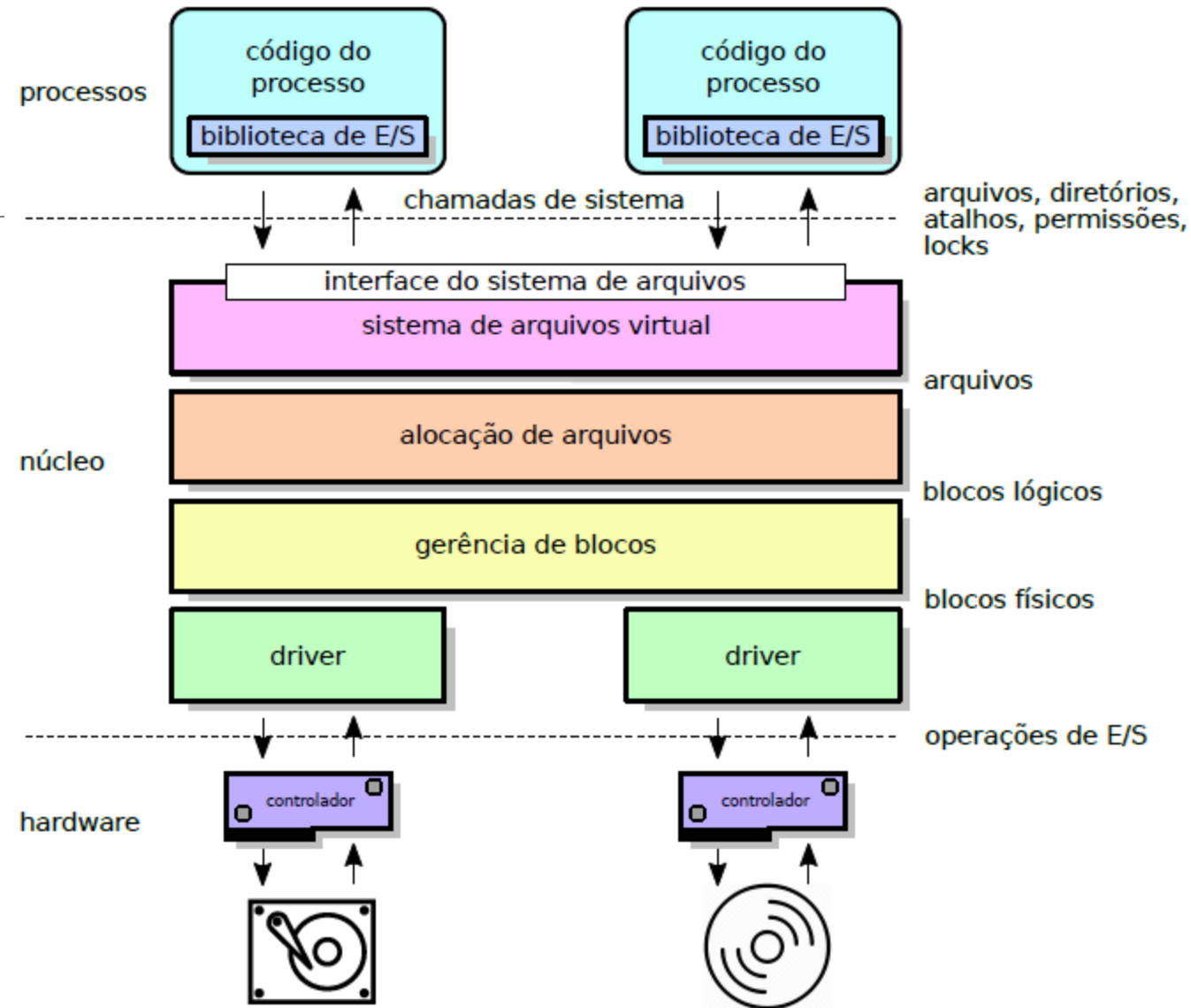
1. Introdução

Dispositivos:

- São os responsáveis pelo **armazenamento de dados**;
- Exemplos: discos rígidos e bancos de memória flash.

Controladores:

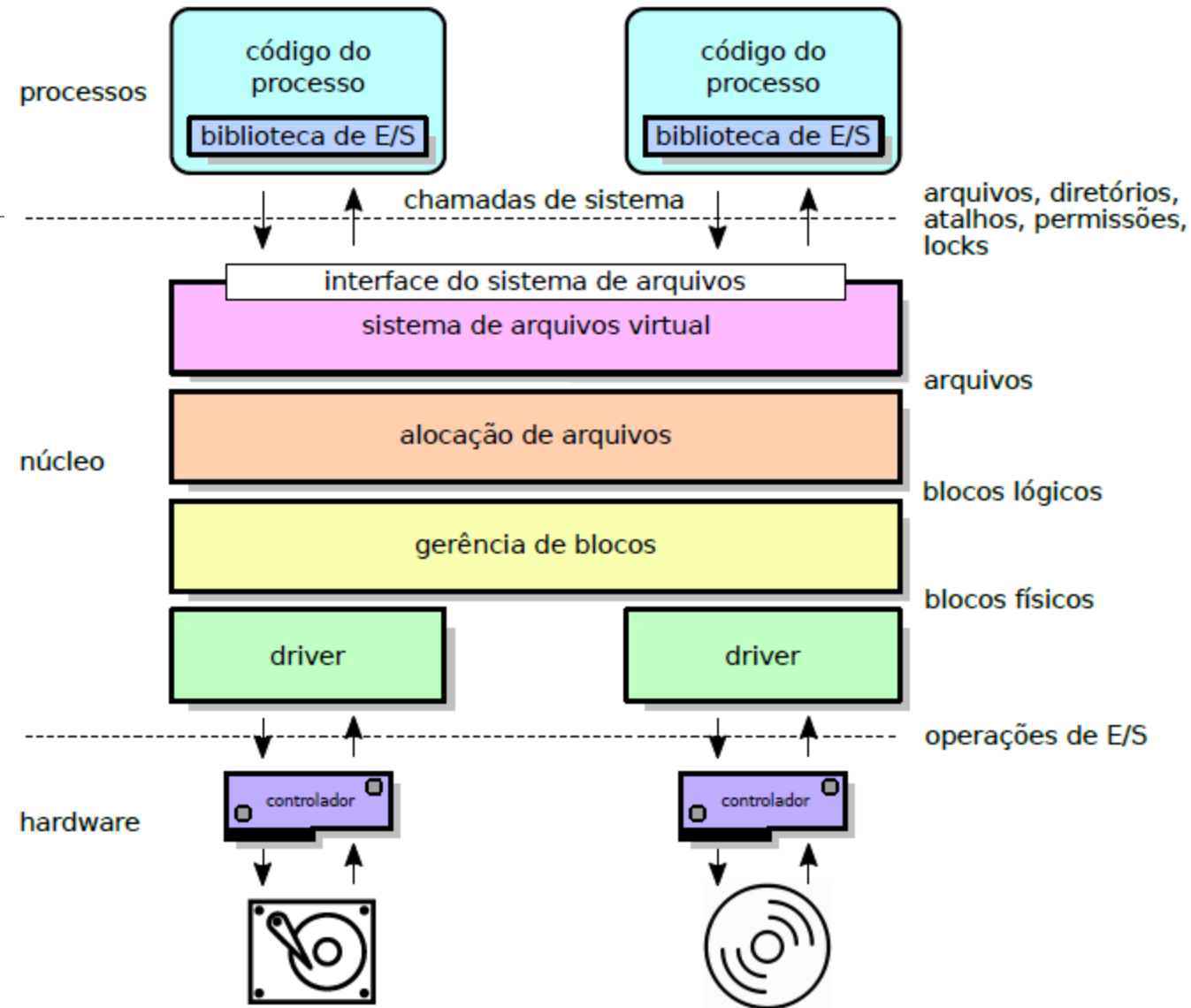
- São os **circuitos eletrônicos dedicados ao controle dos dispositivos físicos**;
- Eles são acessados através de portas de entrada/saída, interrupções e canais de acesso direto à memória (DMA).



1. Introdução

Drivers:

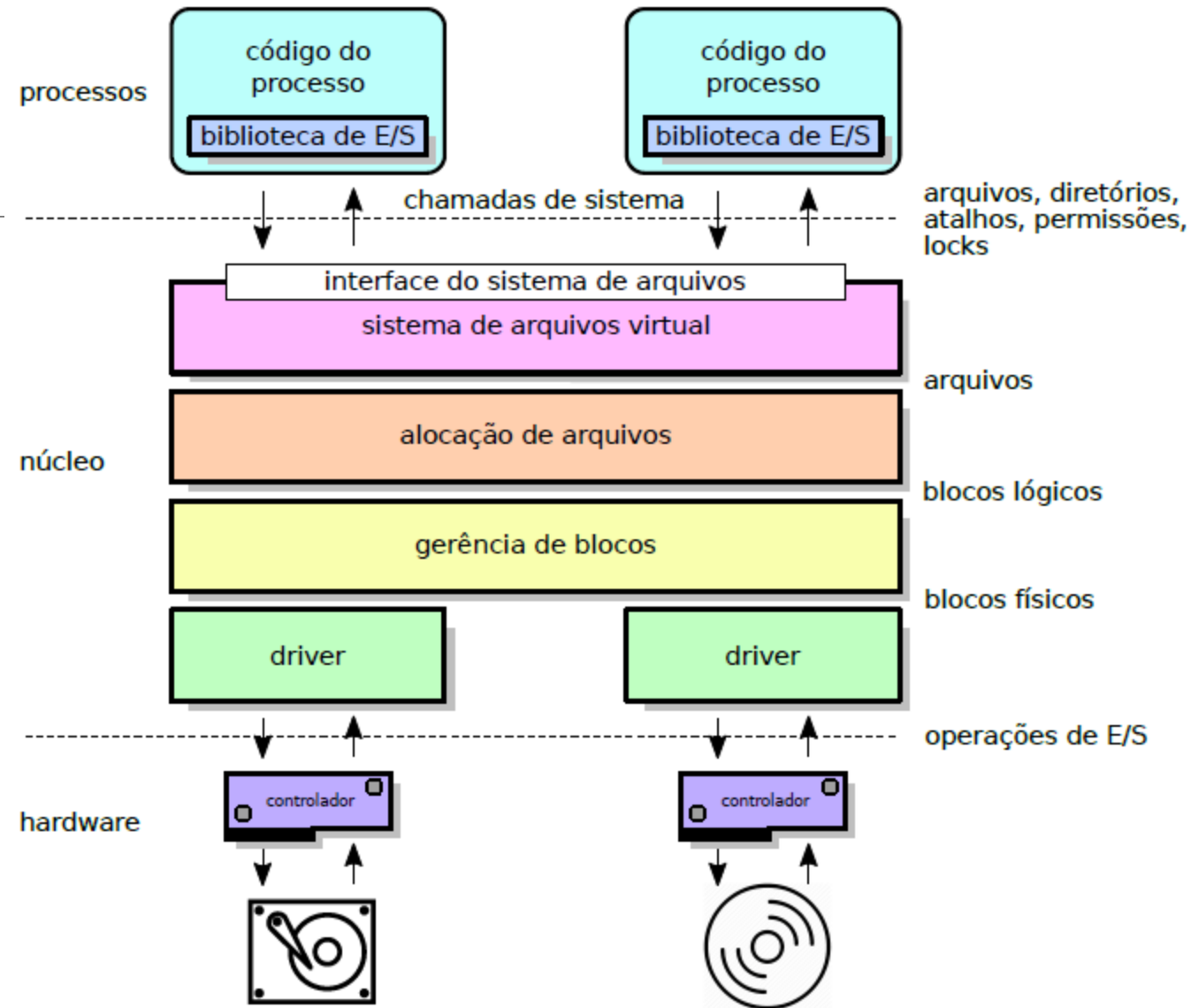
- **Interagem com os controladores de dispositivos** para configurá-los e realizar as transferências de dados entre o sistema operacional e os dispositivos;
- **Os drivers ocultam as diferenças entre controladores** e oferecem às camadas superiores do núcleo uma **interface padronizada** para acesso aos dispositivos de armazenamento.



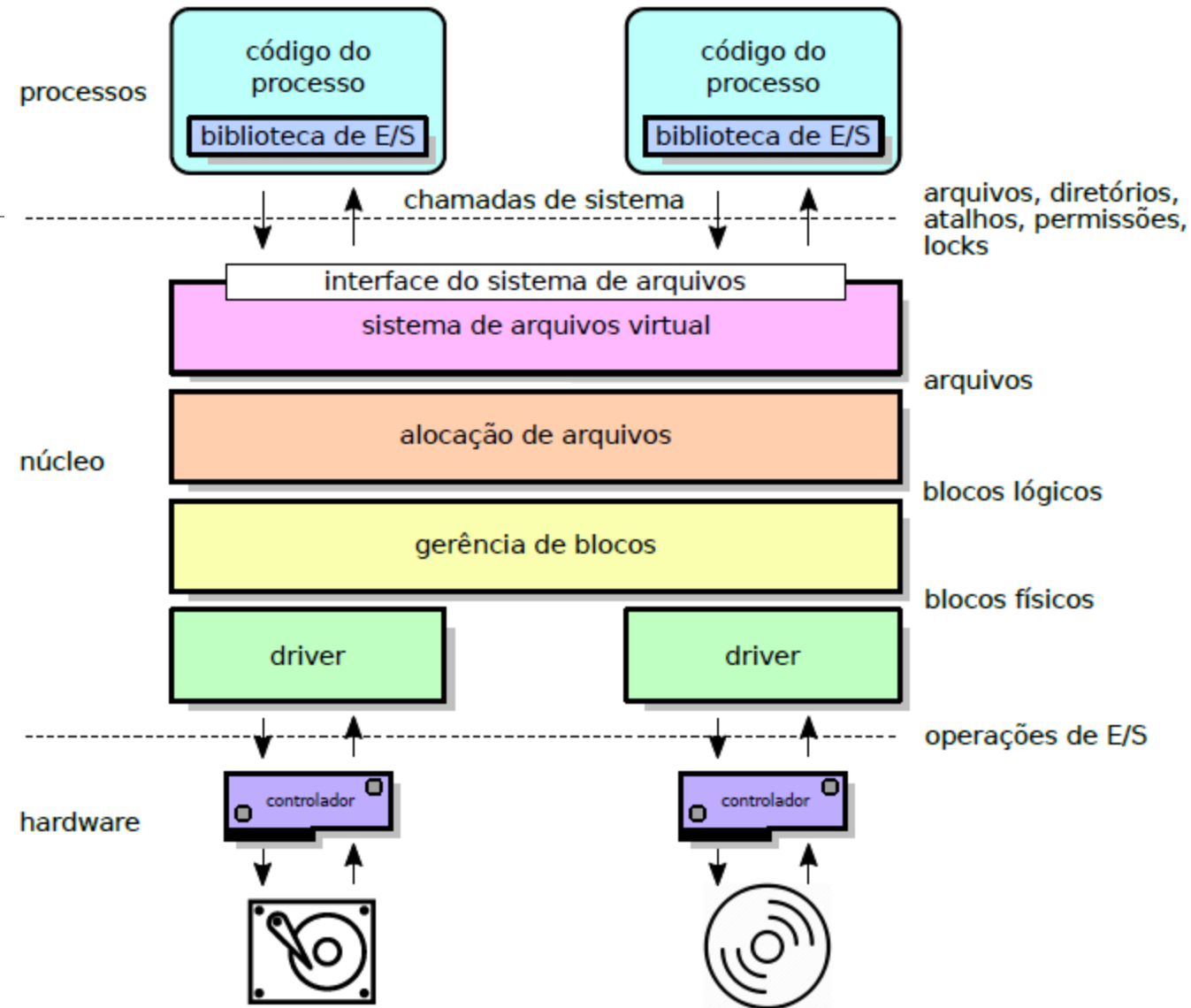
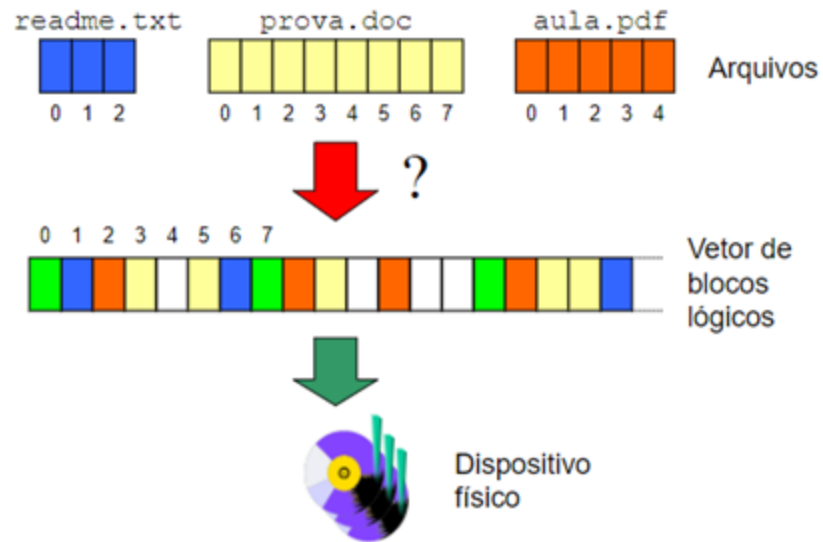
1. Introdução

Gerência de blocos:

- **Gerencia o fluxo de blocos de dados** entre as **camadas superiores** e os **dispositivos de armazenamento**;
- As funções mais importantes desta camada são efetuar o **mapeamento de blocos lógicos nos blocos físicos** do dispositivo e o **caching de blocos**.



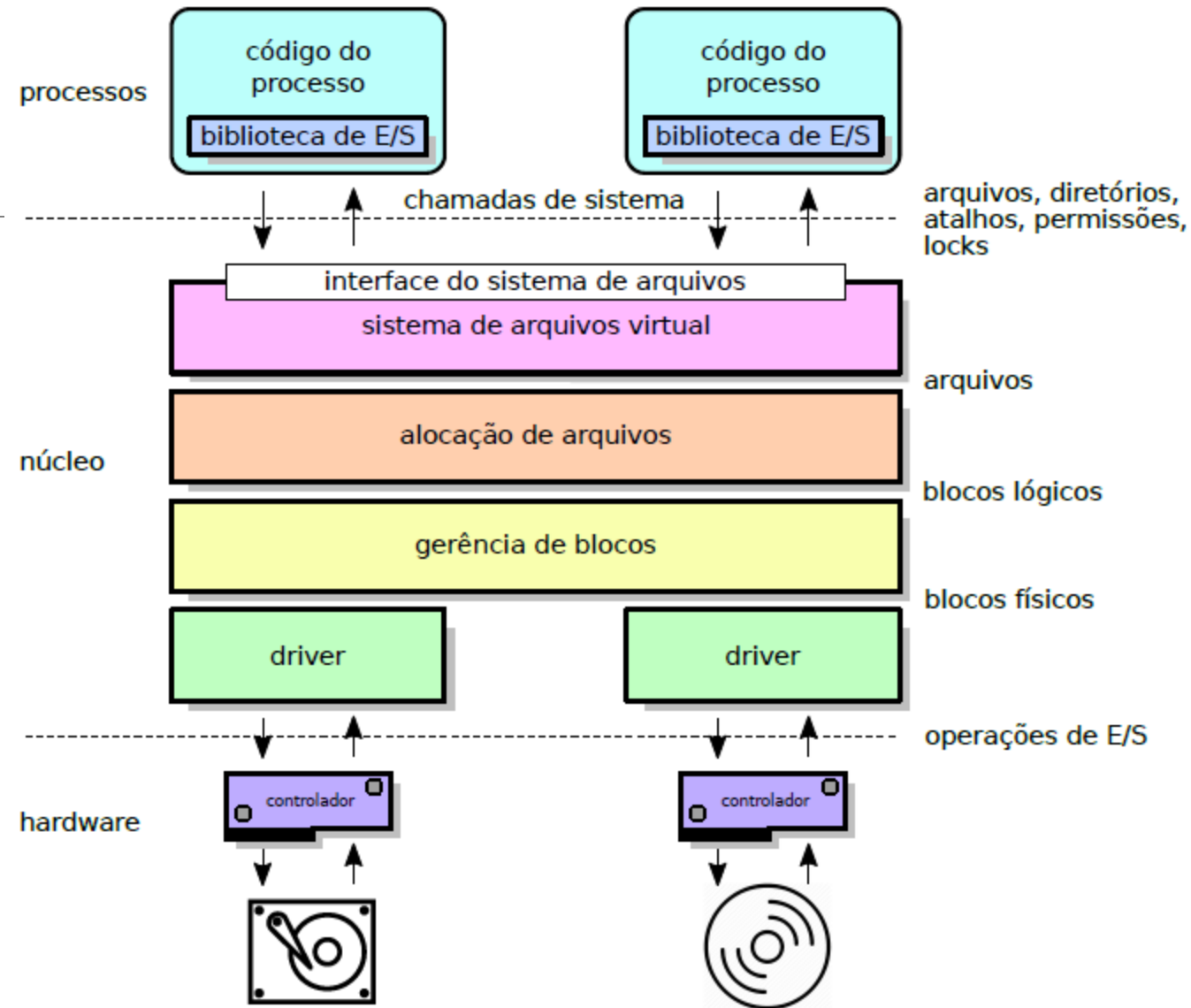
1. Introdução



1. Introdução

Alocação de arquivos:

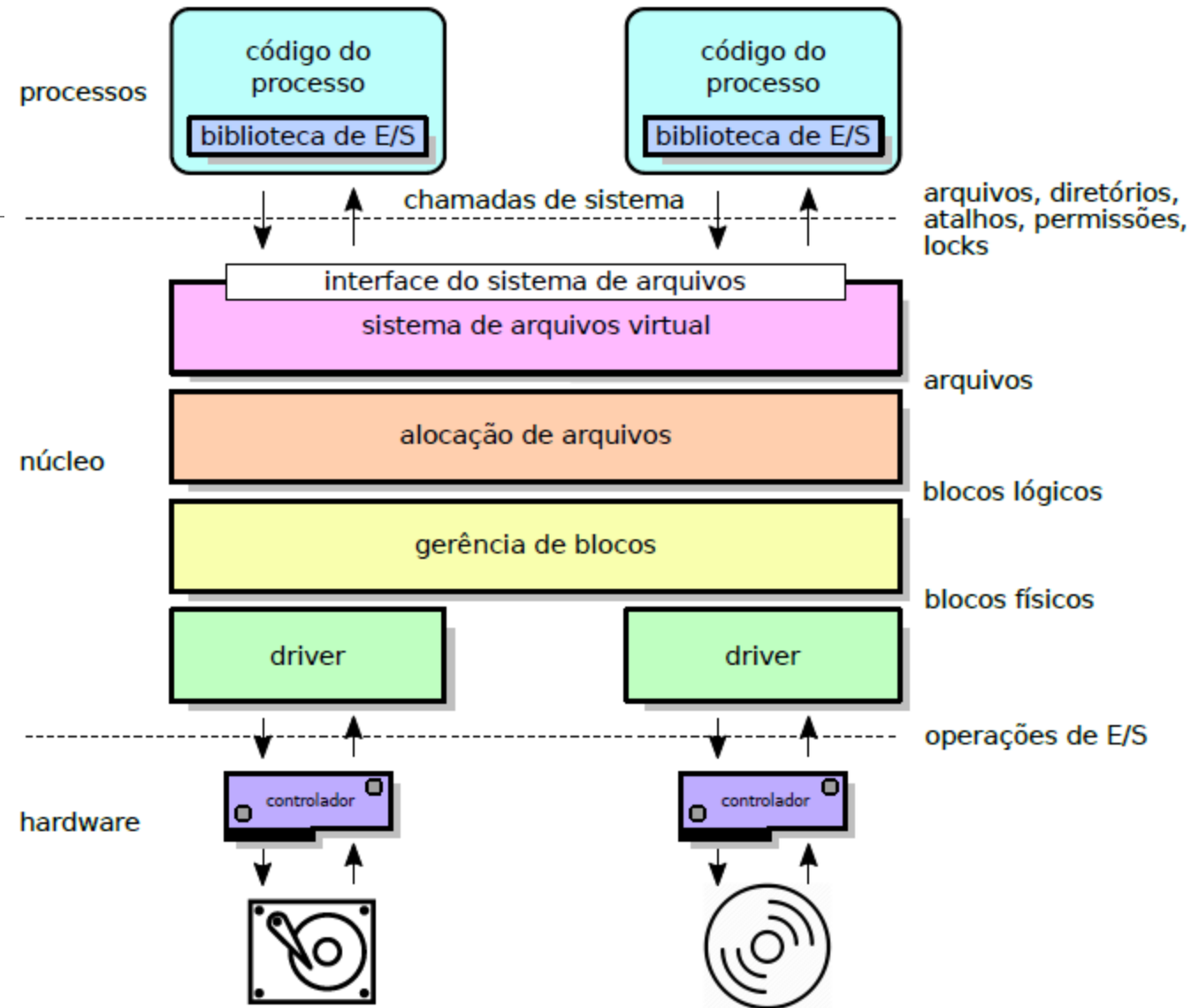
- Realiza a alocação dos arquivos sobre os blocos lógicos oferecidos pela camada de gerência de blocos;
- Cada arquivo é visto como uma sequência de blocos lógicos que deve ser armazenada nos blocos dos dispositivos de forma eficiente, robusta e flexível.



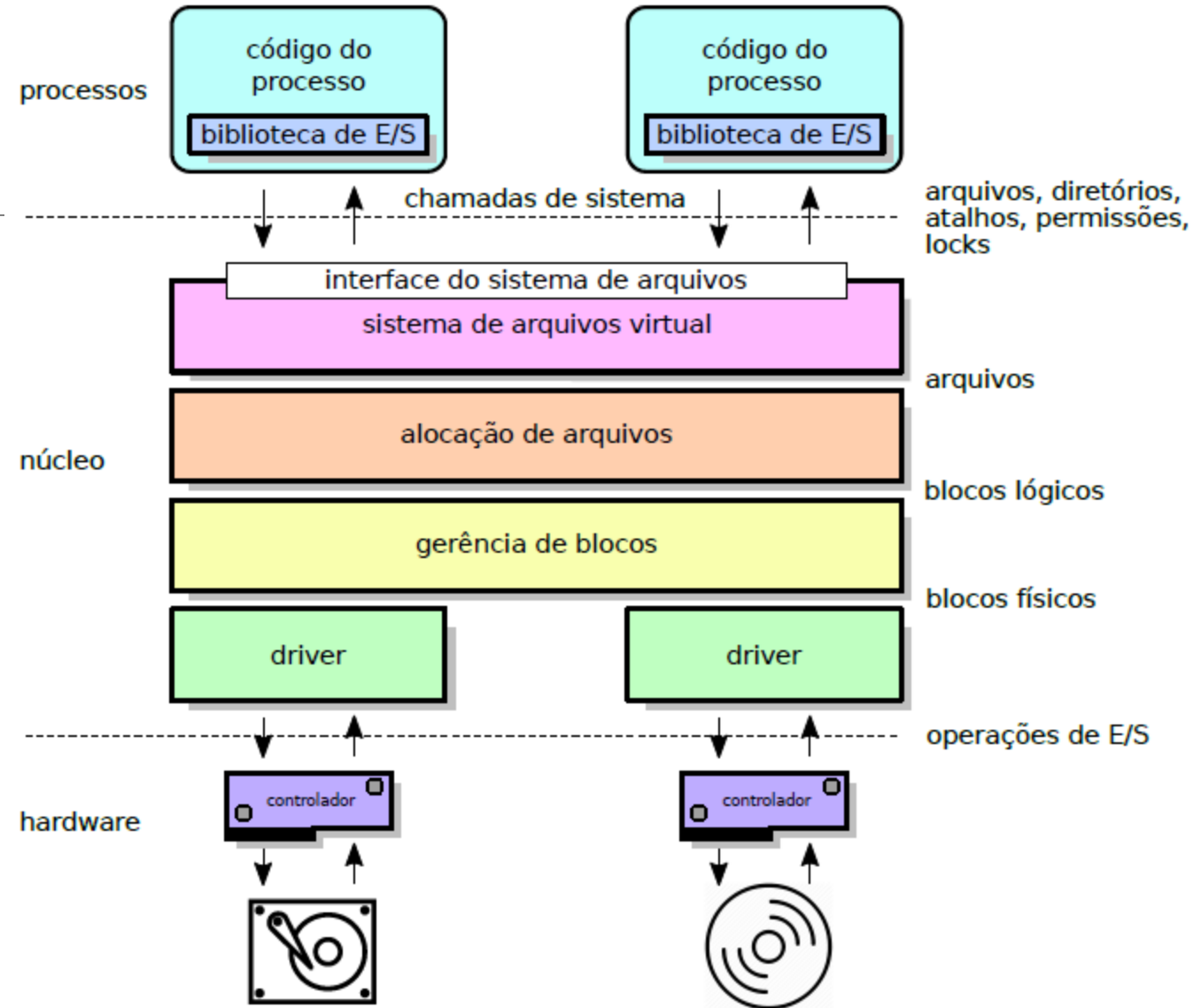
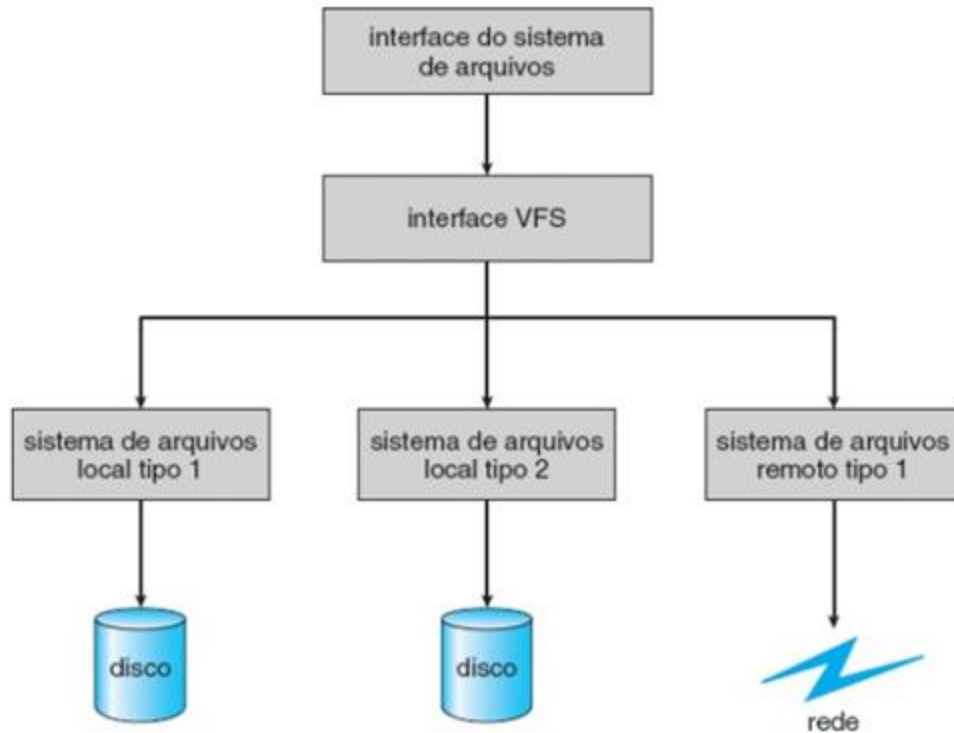
1. Introdução

Sistema de arquivos virtual (VFS):

- Constrói as abstrações de diretórios e atalhos, além de gerenciar as permissões associadas aos arquivos e as travas de acesso compartilhado;
- Outra responsabilidade importante desta camada é manter o registro de cada arquivo aberto pelos processos;
- Este registro envolve a posição da última operação no arquivo, o modo de abertura usado e o número de processos que estão usando o arquivo.



1. Introdução



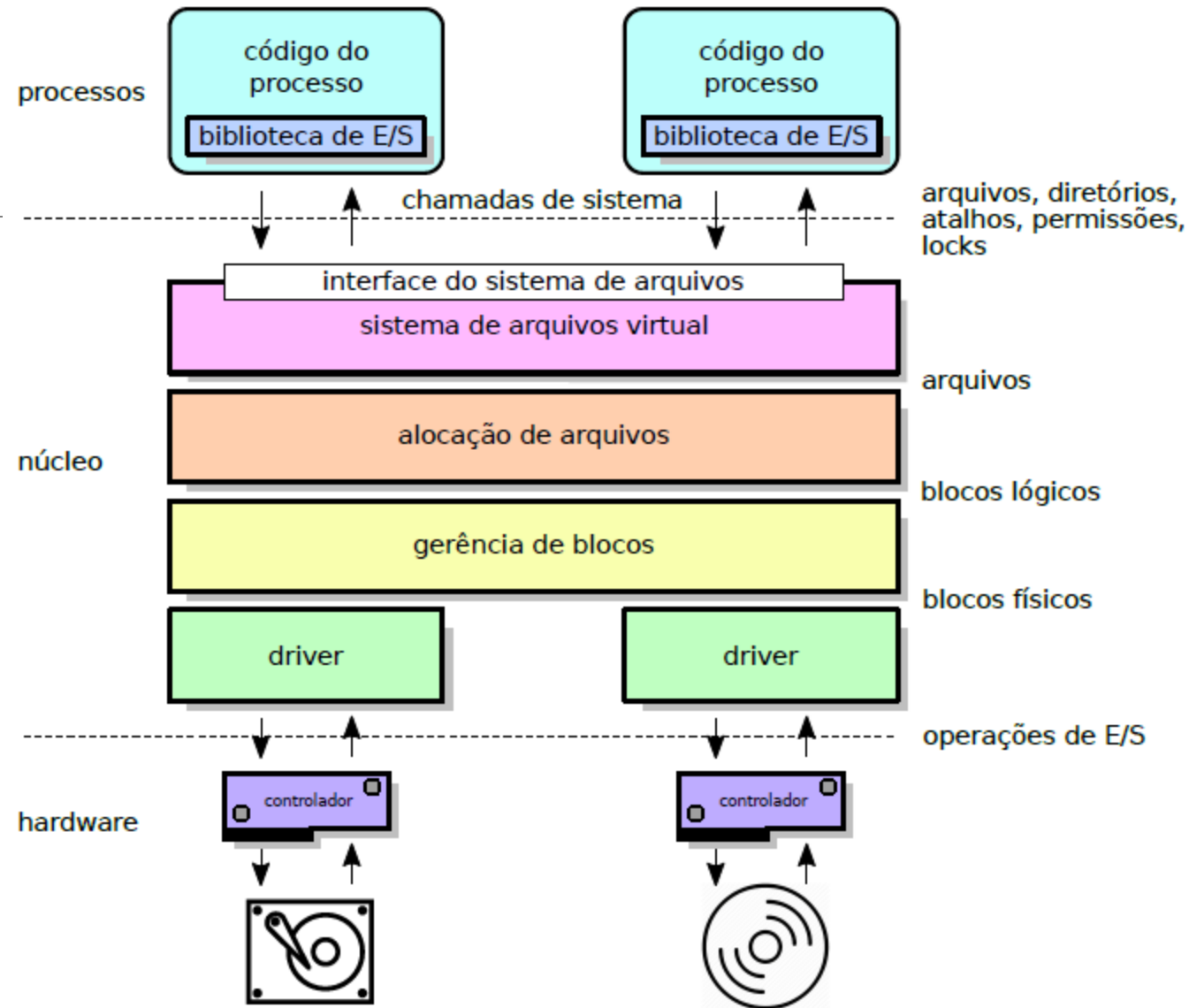
1. Introdução

Interface do sistema de arquivos:

- Conjunto de chamadas de sistema oferecidas aos processos do espaço de usuários para a criação e manipulação de arquivos.

Bibliotecas de entrada/saída:

- Usam as chamadas de sistema da interface do núcleo para construir funções padronizadas de acesso a arquivos para cada linguagem de programação.



1. Introdução

Na implementação de um sistema de arquivos, considera-se que cada arquivo possui **dados** e **metadados**;

Os **dados** de um arquivo são o seu **conteúdo** em si;

Ex:

- Música;
- Planilha;
- Etc.

Os **metadados** do arquivo são seus **atributos** (nome, datas, permissões de acesso, etc.) e todas as informações de controle necessárias para localizar e manter seu conteúdo no disco.

Obs:

- Também são considerados metadados as informações necessárias à gestão do sistema de arquivos, como os mapas de blocos livres, etc.

2. Espaços de armazenamento

Um computador normalmente possui um ou mais dispositivos para armazenar arquivos, que podem ser:

- Discos rígidos;
- Discos ópticos (CD-ROM, DVD-ROM);
- Discos de estado sólido (baseados em memória flash, como pendrives USB);
- Etc.

2. Espaços de armazenamento

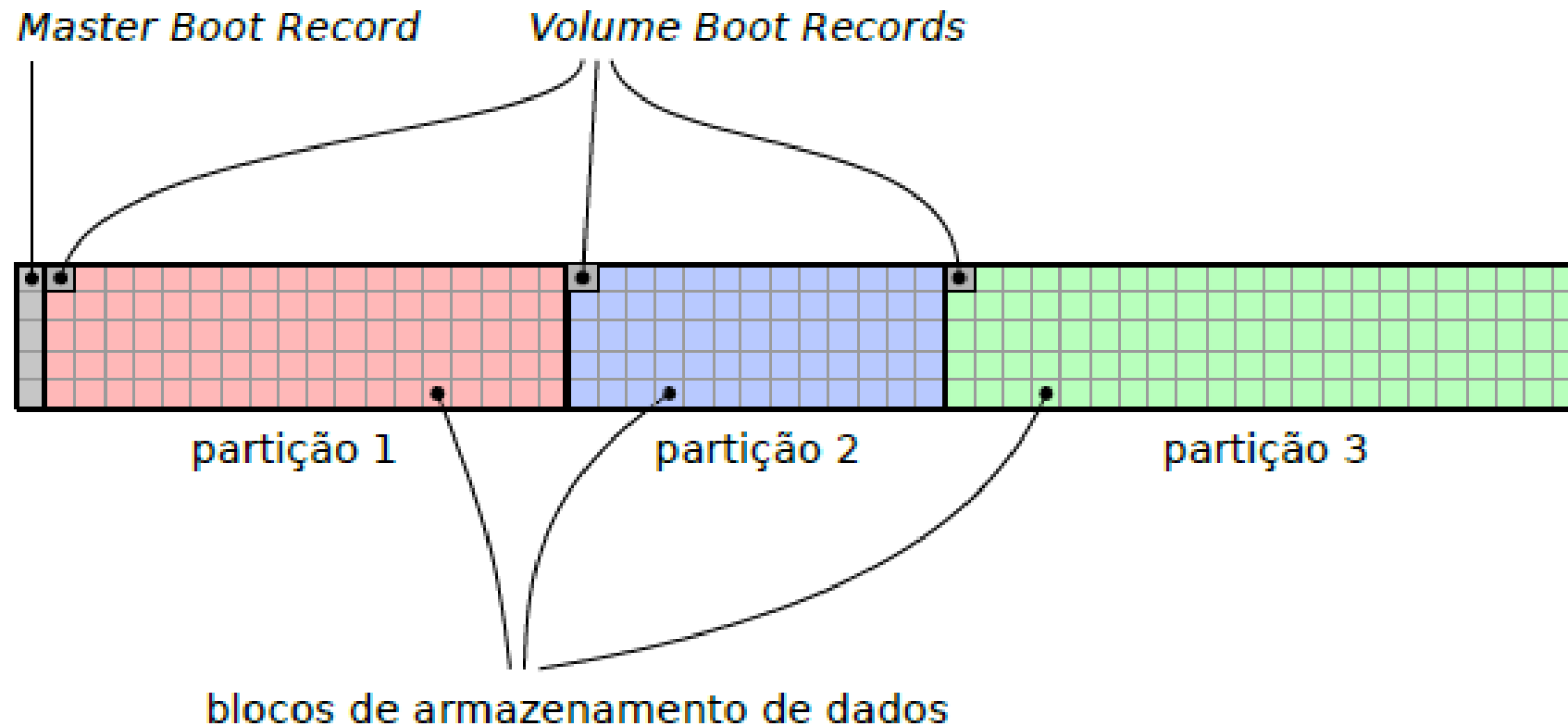
Em linhas gerais, um **disco é visto pelo sistema operacional como um grande vetor de blocos de dados de tamanho fixo, numerados sequencialmente;**

As operações de leitura e escrita de dados nesses dispositivos são feitas bloco a bloco;

Por essa razão, esses dispositivos são chamados dispositivos de blocos (*block devices* ou *block-oriented devices*);

O espaço de armazenamento de cada dispositivo é dividido em uma pequena área de configuração reservada, no início do disco, e uma ou mais partições, que podem ser vistas como espaços independentes.

2. Espaços de armazenamento



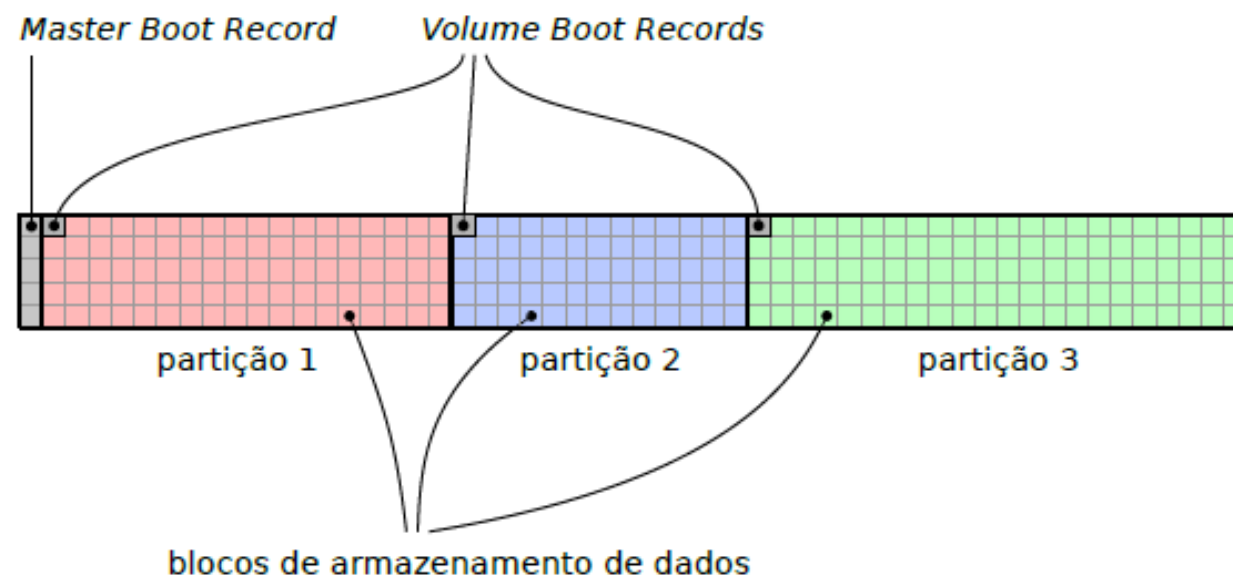
2. Espaços de armazenamento

A **área de configuração** contém uma tabela de partições com informações sobre o particionamento do dispositivo:

Ex:

- Número do bloco inicial;
- Quantidade de blocos
- Etc.

Além disso, essa área contém também um pequeno código executável usado no processo de inicialização do sistema operacional (boot), por isso ela é usualmente chamada de boot sector ou MBR (*Master Boot Record*, em PCs).



2. Espaços de armazenamento

No início de cada partição do disco há também um ou mais blocos reservados;

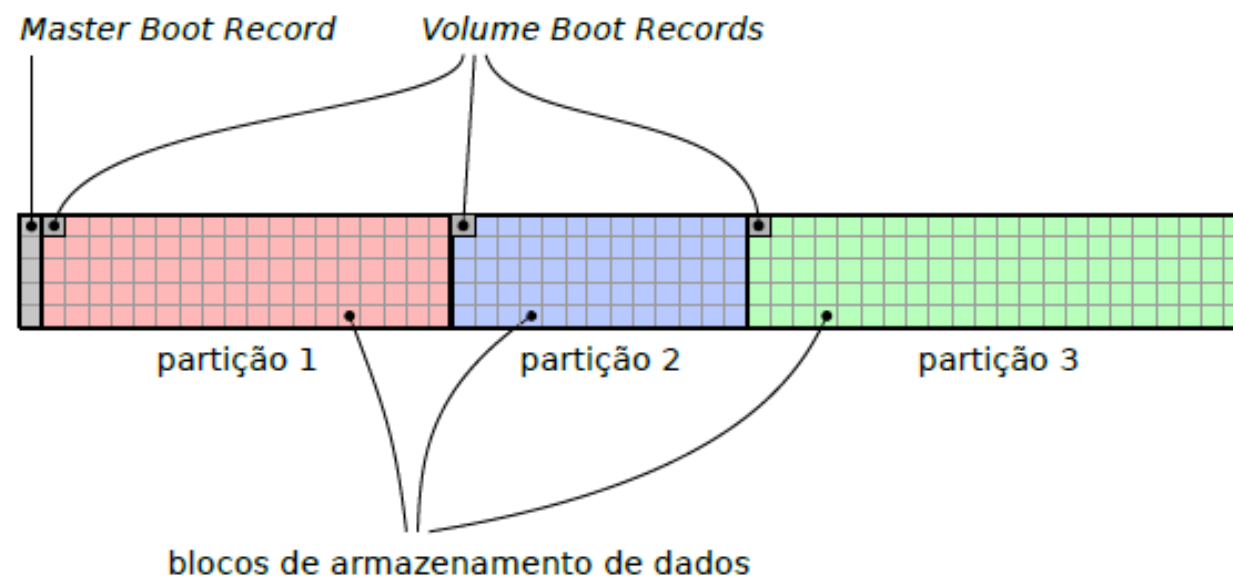
Eles são usados para a descrição do conteúdo daquela partição e para armazenar o código de lançamento do sistema operacional, se aquela for uma partição inicializável (*bootable partition*);

Essa área reservada é denominada bloco de inicialização ou VBR - *Volume Boot Record*;

O restante dos blocos da partição está disponível para o armazenamento de arquivos;

Obs:

- Existem diversos formatos possíveis para os blocos de inicialização de disco e de partição, além da estrutura da própria tabela de partição.



2. Espaços de armazenamento

Um termo frequentemente utilizado em sistemas de arquivos é o **volume**;

Volume é um espaço de armazenamento de dados, do ponto de vista do sistema operacional;

Em sua forma mais simples, cada volume corresponde a uma partição, mas configurações mais complexas são possíveis;

Exemplo:

- O subsistema LVM (Logical Volume Manager) do Linux permite construir volumes lógicos combinando vários discos físicos, como nos sistemas RAID.

3. Montagem de volumes

Para que o sistema operacional possa acessar os arquivos armazenados em um volume, ele deve:

1. Ler os dados presentes em seu bloco de inicialização (que **descrevem o tipo de sistema de arquivos do volume**);
2. Criar as **estruturas em memória** que representam esse volume dentro do núcleo do SO;
3. **Definir um identificador** para o volume, de forma que os processos possam acessar seus arquivos.

Esse procedimento é denominado **montagem do volume**;

O procedimento oposto, a **desmontagem**, consiste em fechar todos os arquivos abertos no volume e remover as estruturas de memória usadas para gerenciá-lo.

3. Montagem de volumes

Apesar de ser realizada para todos os volumes, a **montagem** é um procedimento particularmente frequente no caso de **mídias removíveis**, como:

- CD-ROMs;
- DVD-ROMs
- Pendrives USB;
- Etc.

Neste caso, a **desmontagem** do volume inclui também ejetar a mídia (CD, DVD) ou avisar ao usuário que ela pode ser removida (discos USB).

3. Montagem de volumes

Ao **montar um volume**, deve-se fornecer aos processos e usuários uma referência para seu acesso, denominada **ponto de montagem** (mounting point);

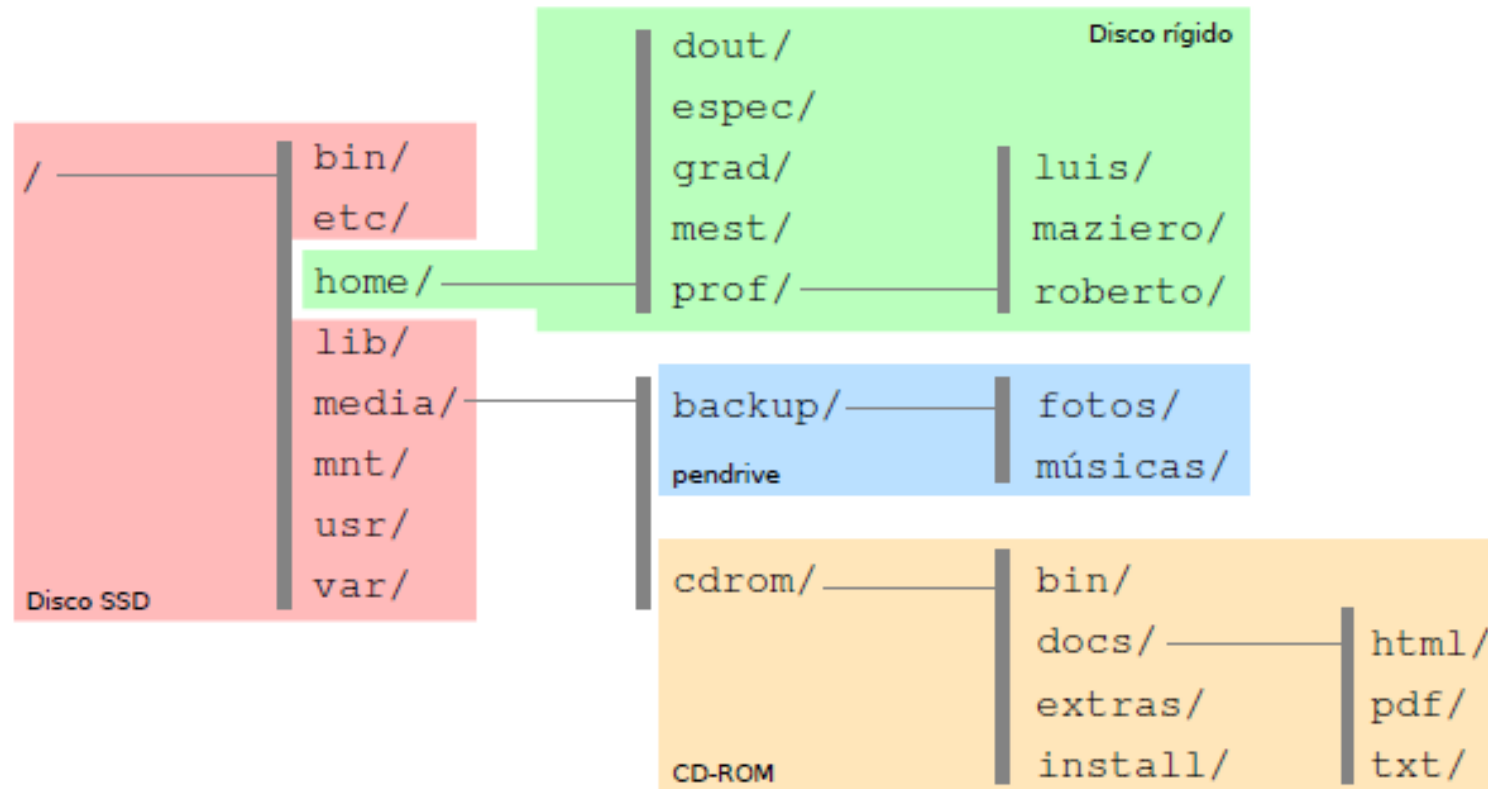
Sistemas UNIX normalmente definem os **pontos de montagem** de volumes como posições dentro da **árvore principal do sistema de arquivos**;

Dessa forma, há um volume principal, montado durante a inicialização do sistema operacional;

O próprio sistema operacional normalmente reside no volume principal, que define a estrutura básica da árvore de diretórios;

Os **volumes secundários são montados como subdiretórios** na árvore do volume principal, através do comando *mount*.

3. Montagem de volumes



3. Montagem de volumes

Em sistemas de arquivos como no DOS e no **Windows**, é comum definir cada **volume** montado como um disco lógico distinto, chamado simplesmente de **disco** ou drive e **identificado por uma letra** (“A:”, “C:”, “D:”, etc.);

O **sistema de arquivos NTFS do Windows** também permite a montagem de volumes como subdiretórios da árvore principal, da mesma forma que o UNIX.

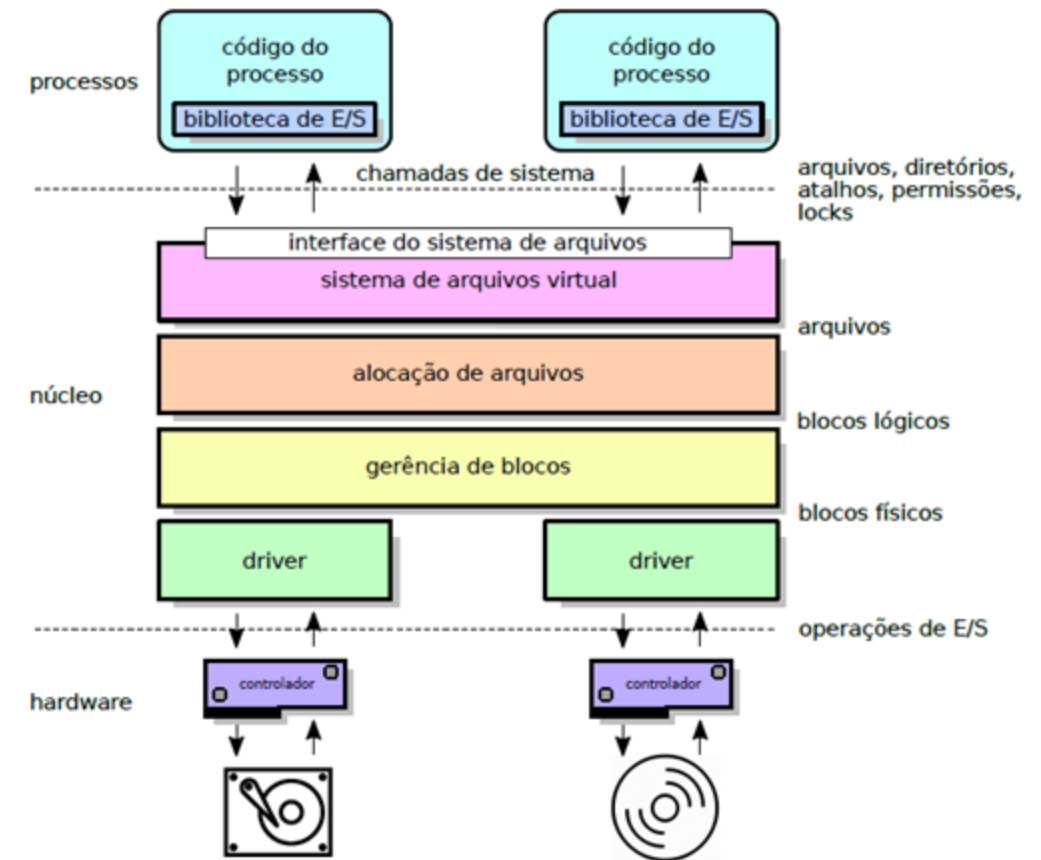
4. Gestão de blocos

A função primordial da camada de gestão de blocos é **interagir com os drivers** de dispositivos para realizar as **operações de leitura e escrita de blocos** de dados;

Neste nível do subsistema ainda não existe a noção de arquivo;

Portanto, todas as operações nesta camada dizem respeito a blocos de dados;

Além da interação com os drivers, esta camada também é responsável pelo **mapeamento entre blocos físicos e blocos lógicos** e pelo **mecanismo de caching de blocos**.



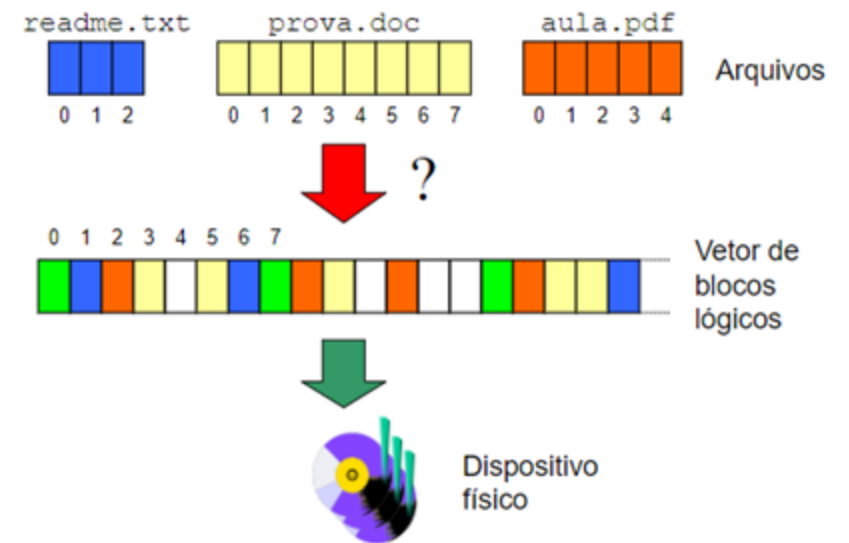
4. Gestão de blocos

Um **disco rígido** pode ser visto como um **conjunto de blocos de tamanho fixo** (geralmente de 512 ou 4.096 bytes);

Os **blocos do disco rígido** são normalmente denominados **blocos físicos**;

Os sistemas operacionais costumam agrupar os blocos físicos em blocos lógicos ou clusters, que são grupos de $2n$ blocos físicos consecutivos;

O objetivo é simplificar a gerência da imensa quantidade de blocos físicos e melhorar o desempenho das operações de leitura/escrita.



4. Gestão de blocos

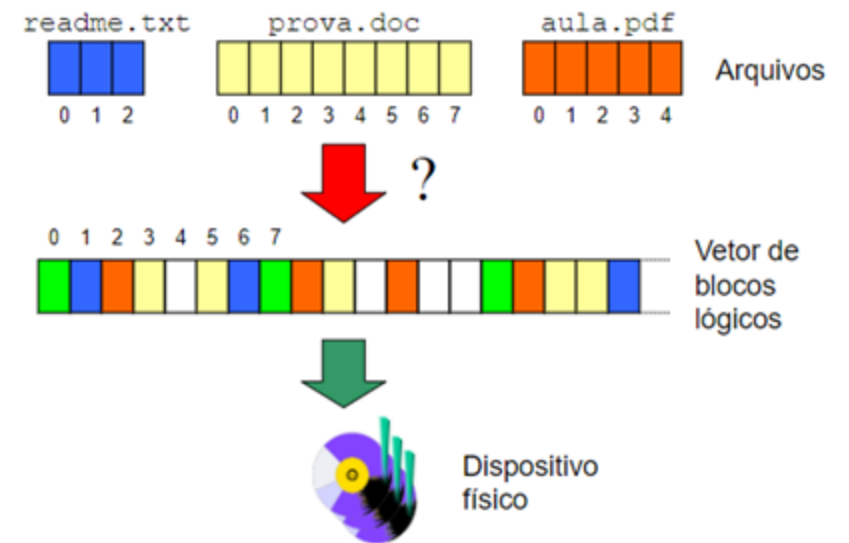
A maior parte das **operações** e **estruturas** de dados definidas nos discos pelos sistemas operacionais são baseadas em **blocos lógicos**;

Os blocos lógicos também definem a unidade mínima de alocação de arquivos e diretórios: cada arquivo ou diretório ocupa um ou mais blocos lógicos para seu armazenamento;

O número de blocos físicos em cada bloco lógico é fixo e definido pelo sistema operacional ao formatar a partição, em função de vários parâmetros, como:

- Tamanho da partição;
- Sistema de arquivos usado;
- Tamanho das páginas de memória RAM.

Blocos lógicos com tamanhos de 4 KB a 64 KBytes são frequentemente usados.



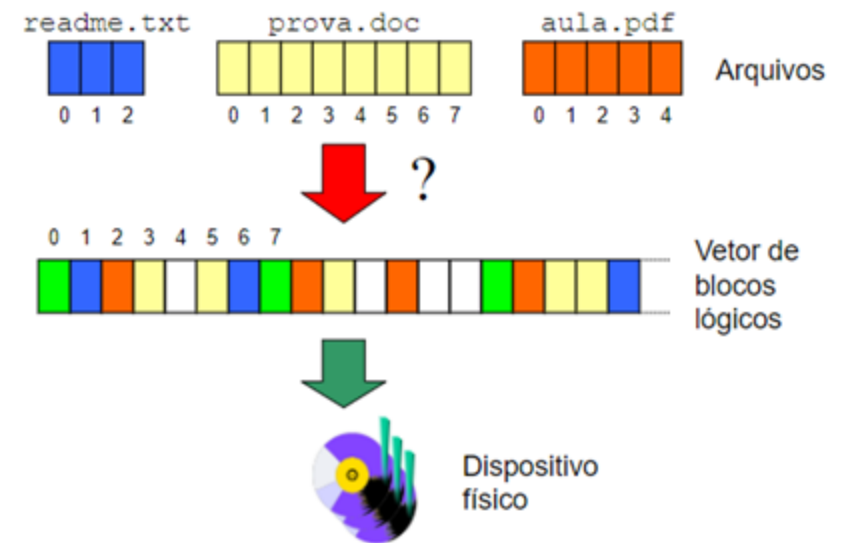
4. Gestão de blocos

Blocos lógicos maiores (32 KB ou 64 KB) levam a uma **menor quantidade de blocos lógicos** a gerenciar pelo **SO** em cada disco;

Isso implica em mais **eficiência de entrada/saída**, pois mais dados são transferidos em cada operação;

Entretanto, blocos grandes podem **gerar muita fragmentação interna**, causando perda de espaço útil;

Uma forma de tratar esse problema é escolher um tamanho de bloco lógico adequado ao tamanho médio dos arquivos a armazenar no disco, no momento de sua formatação.



5. Alocação de arquivos

O **problema da alocação** de arquivos consiste em dispor (**alocar**) o **conteúdo e os metadados** dos arquivos dentro desses **blocos**;

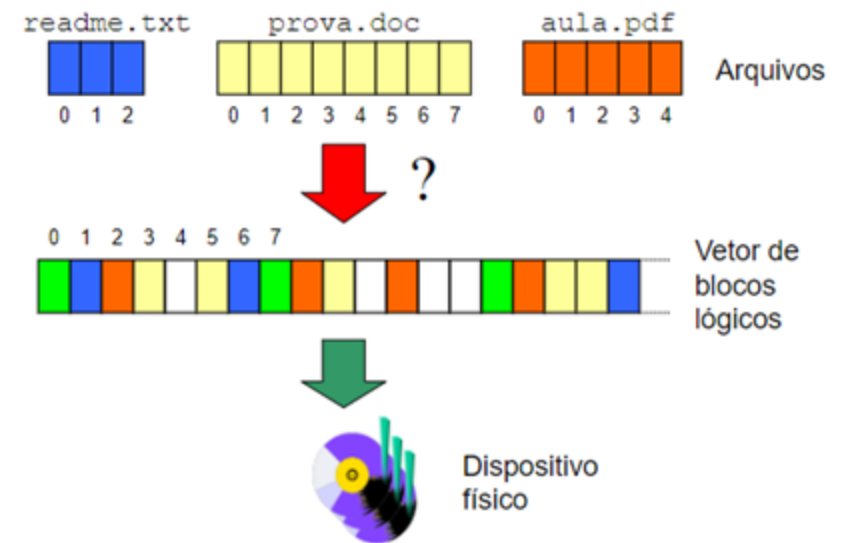
Como os blocos são pequenos, um arquivo pode precisar de muitos blocos para ser armazenado no disco;

Exemplo:

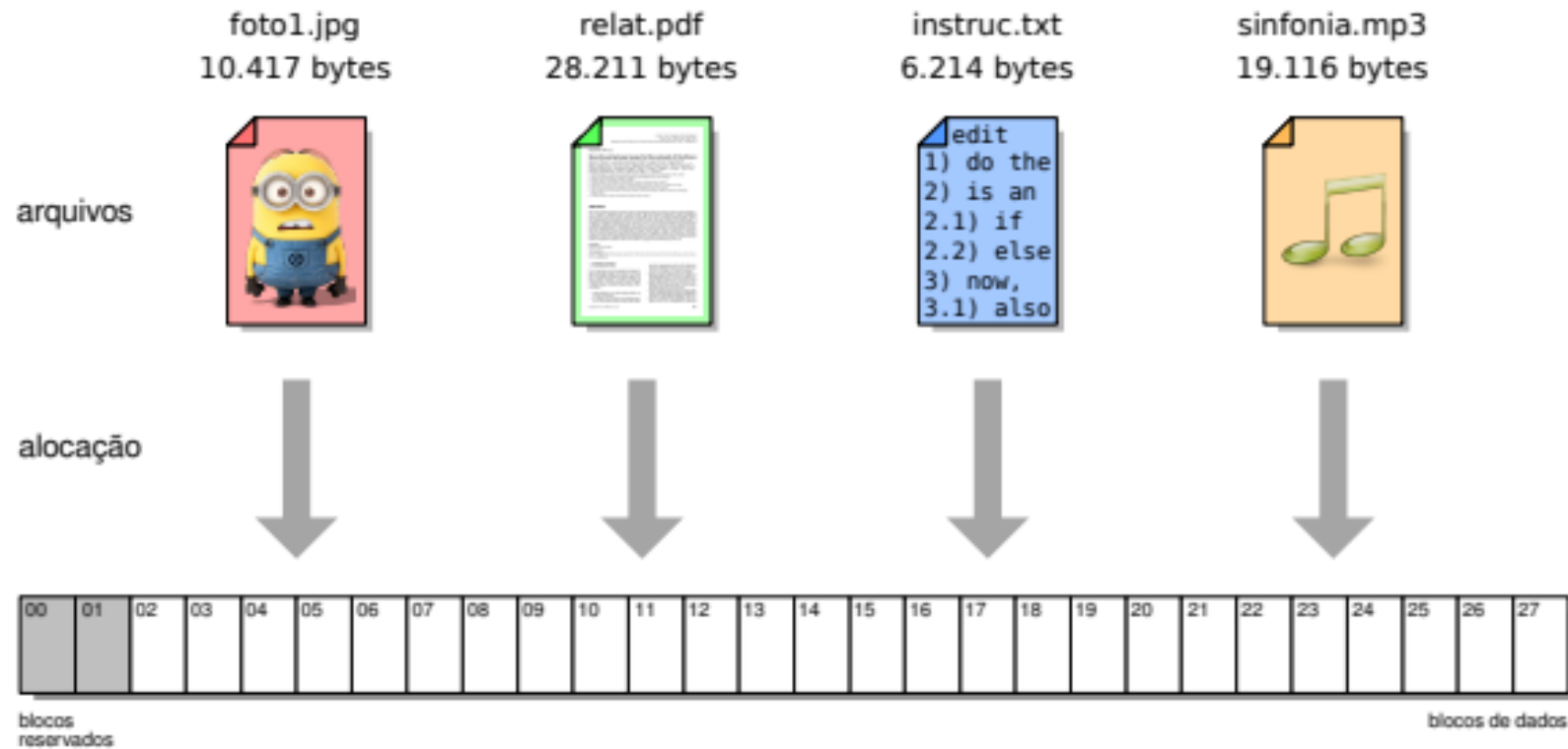
- Um arquivo de filme em formato MP4 com 1 GB de tamanho ocuparia 262.144 blocos de 4 KBytes.

O conteúdo do arquivo deve estar disposto nestes blocos de forma a permitir um acesso rápido, flexível e confiável;

Por isso, a forma de alocação dos arquivos nos blocos do disco tem um impacto importante sobre o desempenho e a robustez do sistema de arquivos.



5. Alocação de arquivos



5. Alocação de arquivos

Há três estratégias básicas de alocação de arquivos nos blocos lógicos do disco.

São as alocações:

- Contígua;
- Encadeada (Simples e FAT);
- Indexada.

6. Gestão do espaço livre

Além de manter informações sobre que blocos são usados por cada arquivo no disco, a camada de alocação de arquivos deve manter um **registro atualizado de quais blocos estão livres**, ou seja, **não estão ocupados por nenhum arquivo ou metadado**;

Isto é importante para identificar rapidamente os blocos no momento de criar um novo arquivo ou aumentar um arquivo existente;

Algumas técnicas de gerência de blocos são sugeridas na literatura, como:

- Mapa de bits;
- Lista de blocos livres;
- Tabela de grupos de blocos livres;
- Etc.

6. Gestão do espaço livre

Na abordagem de mapa de bits, um pequeno conjunto de blocos na área reservada do volume é usado para manter um mapa de bits;

Nesse mapa, cada bit representa um bloco lógico da partição, que pode estar livre (0) ou ocupado (1);

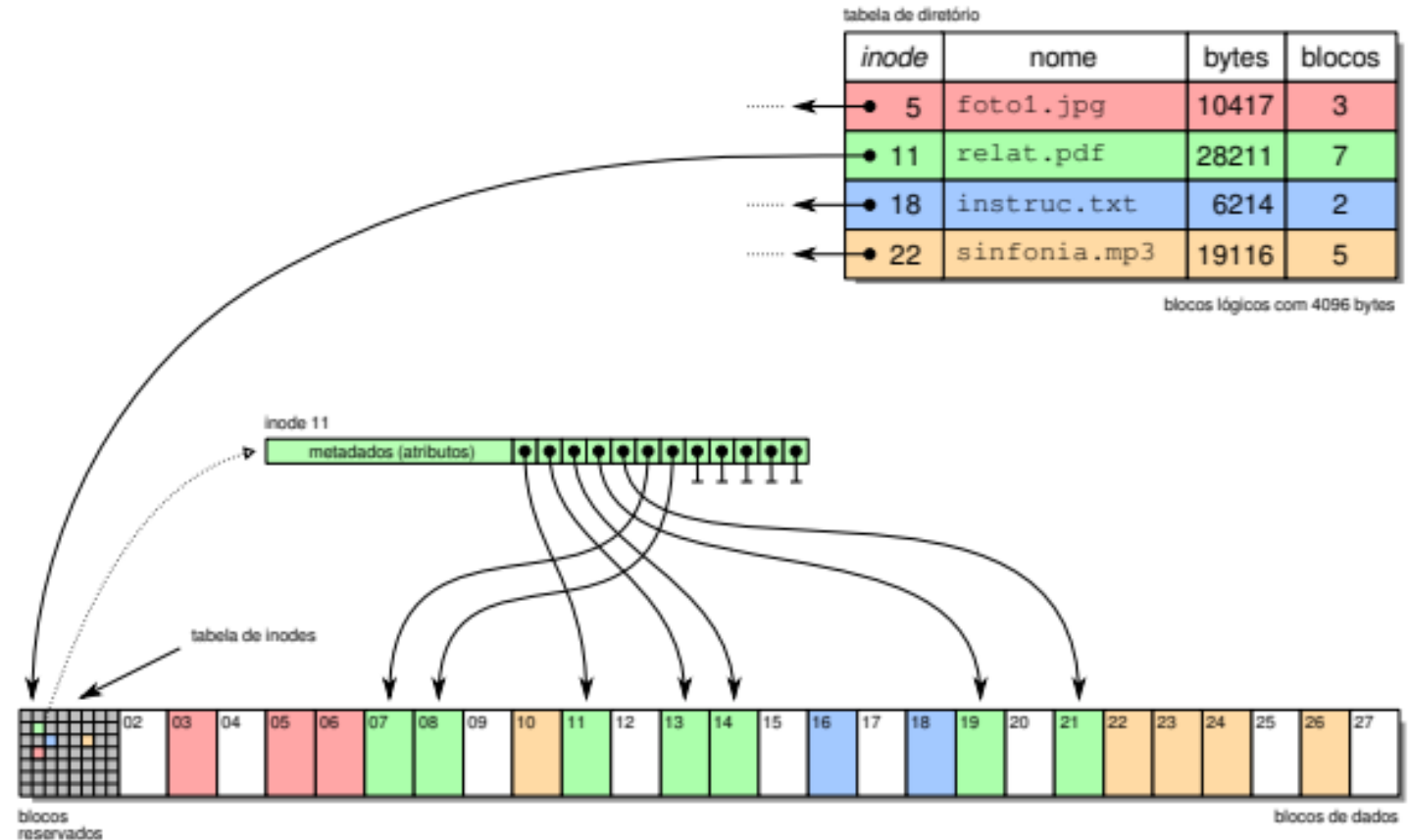
O mapa de bits tem como vantagens ser simples de implementar e ser bem compacto;

Exemplo:

- Em um disco de 500 GBytes com blocos lógicos de 4.096 bytes, seriam necessários 131.072.000 bits no mapa, o que representa 16.384.000 bytes, ocupando 4.000 blocos (ou seja, 0,003% do total de blocos lógicos do disco).

6. Gestão do espaço livre

No exemplo, o mapa de bits de blocos livres seria: 1101 0111 1011 0110 1011 0111 1010
(considerando 0 para bloco livre e incluindo os blocos reservados no início da partição).



6. Gestão do espaço livre

Na abordagem de **lista de blocos livres**, cada bloco livre contém um ponteiro para o próximo bloco livre do disco, de forma similar à alocação encadeada de arquivos;

Essa abordagem é ineficiente, por exigir um acesso a disco para cada bloco livre requisitado.

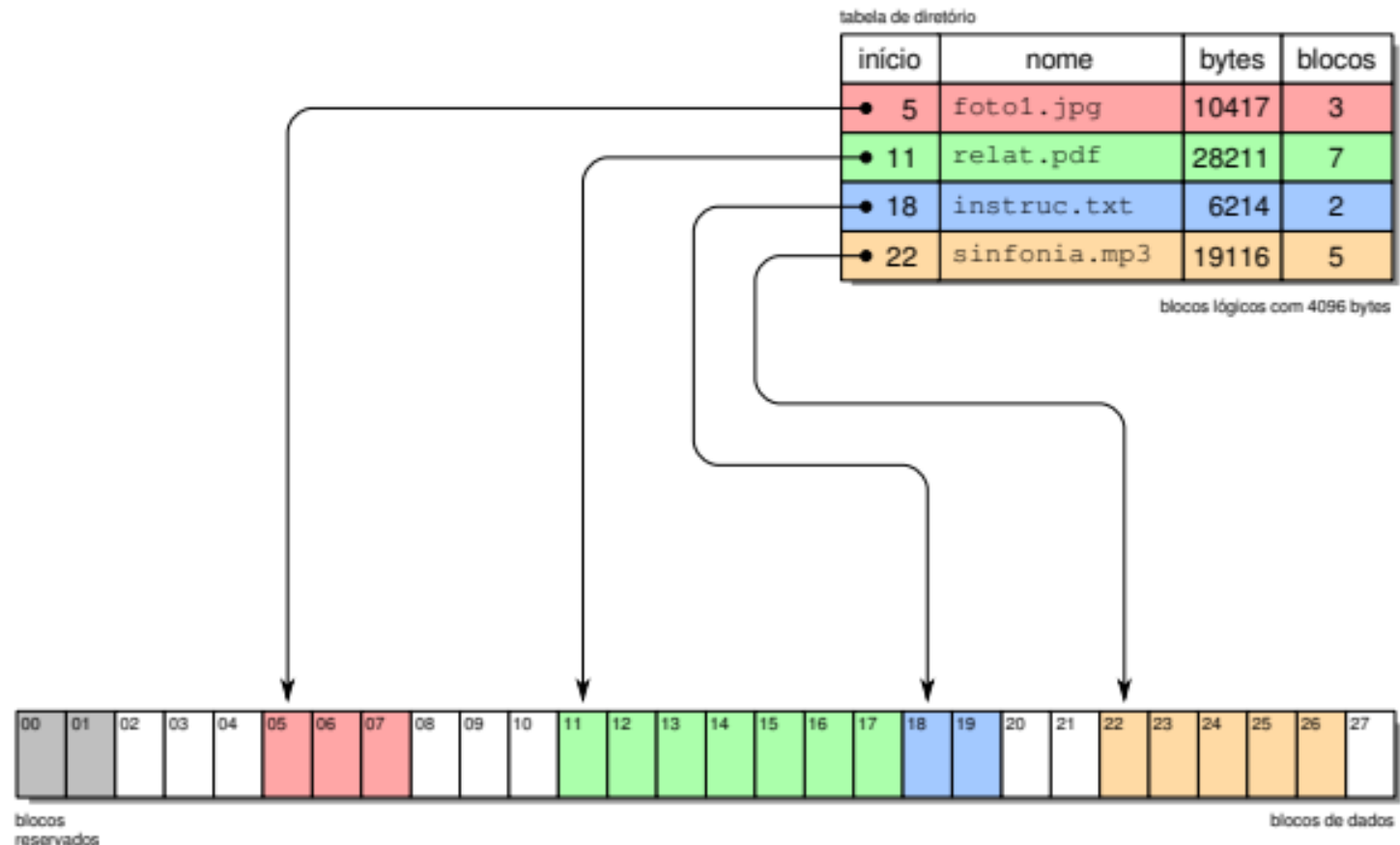
6. Gestão do espaço livre

Outra melhoria similar consiste em manter uma **tabela de grupos de blocos livres**, contendo a localização e o tamanho de um conjunto de blocos livres contíguos no disco;

Cada entrada dessa tabela contém o número do bloco inicial e o número de blocos no grupo;

6. Gestão do espaço livre

No exemplo, a tabela de grupos de blocos livres teria o seguinte conteúdo: {[2, 3], [8, 3], [20, 2], [27, 1]}, com entradas na forma [bloco, tamanho].



6. Gestão do espaço livre

Por outro lado, a abordagem de alocação FAT usa a própria tabela de alocação de arquivos para gerenciar os blocos livres, que são indicados por flags específicos;

No exemplo do slide seguinte, os blocos livres estão indicados com o flag “F” na tabela;

Para encontrar blocos livres ou liberar blocos usados, basta consultar ou modificar as entradas da tabela;

É importante lembrar que, além de manter o registro dos blocos livres, na alocação indexada também é necessário gerenciar o uso dos inodes, ou seja, manter o registros de quais inodes estão livres;

Isso geralmente também é feito através de um mapa de bits.

6. Gestão do espaço livre

