



SISTEMAS OPERACIONAIS

Módulo 9 – Sistema de Arquivos

Prof. Daniel Sundfeld
daniel.sundfeld@unb.br



INTRODUÇÃO

- As aplicações de um computador precisam armazenar as informações durante o processamento
- Algumas vezes a memória principal não é suficiente e é necessário utilizar a memória secundária
- Na aula passada, vimos uma estratégia que visa combinar a memória principal e secundária, criando uma memória maior: a memória virtual



INTRODUÇÃO

- No entanto, muitas aplicações não são capazes de funcionar mantendo todos os dados na memória principal
- Sistemas bancários, sistemas de reservas de hotel, possuem uma grande quantidade de dados a armazenar, mas poucos dados a processar
- Nesses casos, o essencial é manter os dados armazenados durante um período de tempo, que podem ser alguns anos



INTRODUÇÃO

- Essas informações não devem ser perdidas durante uma falha do computador
- Além disso, é frequentemente desejado que diversas aplicações compartilhem esses dados e possam acessar simultaneamente



INTRODUÇÃO

- Existem três requisitos essenciais para o armazenamento de informações a longo prazo:
 - Deve ser possível armazenar uma quantidade muito grande de informações;
 - As informações devem sobreviver ao término do processo que as está utilizando;
 - Múltiplos processos têm de ser capazes de acessá-las ao mesmo tempo



INTRODUÇÃO

- Discos magnéticos (HD) são utilizados a mais de 60 anos para o armazenamento a longo prazo
- Mais recentemente, as unidades de estado sólido (SSD) estão se popularizando devido à alta velocidade de acesso e independência de dispositivos mecânicos
- Fitas e discos óticos já foram utilizados mas estão em desuso devido à lentidão quando comparados aos HDs.



INTRODUÇÃO

- Os arquivos são gerenciados pelo sistema operacional de maneira a facilitar o acesso dos usuários ao seu conteúdo
- Essa gerência é realizada pelo **sistema de arquivos**, quando comparado a outras gerências, gerência de processos e memória, o sistema de arquivos é mais conhecido popularmente
- Essa gerência busca ser independente dos dispositivos envolvidos



INTRODUÇÃO

- Para a CPU, o SO abstrai o conceito do processador para criar a abstração de um processo
- Para a memória principal, ele também abstrai o conceito de memória física para criar o espaço de endereçamento virtual
- Para a memória secundária, ele também utiliza uma abstração: o arquivo



ROTEIRO

- Introdução
- Arquivos
- Diretórios
- Gerência
 - Gerência de Espaço Livre em Disco
 - Gerência de Alocação de Espaço
- Proteção de Acesso
- Implementação de Caches



ARQUIVOS

- Arquivos são unidades lógicas de informação logicamente relacionados que são criadas por processos
- Um disco contém milhares ou até milhões de diferentes arquivos
- Cada arquivo possui um nome que o identifica, composto por uma sequência de caracteres



ARQUIVOS

- Em alguns sistemas de arquivos o nome dos arquivos são case-sensitive, ou seja, há distinção entre caracteres maiúsculos ou minúsculos como em sistemas UNIX, Linux, Mac OS X
- O MS-DOS não faz tal distinção
- Regras sobre a extensão máxima do nome de arquivos e quais caracteres pode-se utilizar (ç, ã) também são específicos dos sistemas de arquivos



ARQUIVOS

- Muitos sistemas aceitam nomes de arquivos de duas partes separadas com um ponto: a primeira é o nome do arquivo, a segunda a extensão

.bak	Cópia de Segurança	.o	Arquivo objeto
.bas	Código-Fonte Basic	.obj	Arquivo objeto
.c	Código-Fonte C	.pdf	Portable Document File
.exe	Arquivo executável	.ps	Arquivo Postscript
.gif	Graphical Interchange Format	.tgz	Arquivo Compactado
.mp3	Música MPEG 3	.zip	Arquivo Compactado



ARQUIVOS

- O nome do arquivo serve para identificá-lo
- Já a extensão diz algo a respeito
- Em todas as variações do UNIX a extensão do arquivo não são impostas pelo sistema operacional
- No entanto, alguns programas como um compilador C, pode se recusar a compilar um arquivo cuja extensão não seja .c
- O SO, no entanto, não se importa



ORGANIZAÇÃO DE ARQUIVOS

- A organização do arquivo consiste em como os dados serão armazenados internamente
- Diferentes formatos de arquivo podem levar a diferentes estruturas de dado
- Por exemplo, arquivos executáveis são estruturados de forma bem diferente de arquivos texto
- A estrutura pode ser definida pelo SO ou pela própria aplicação



ORGANIZAÇÃO DE ARQUIVOS

- A organização de arquivos mais comuns são:
 - Sequência de bytes;
 - Sequência de registros;
 - Árvore de registros.



ORGANIZAÇÃO DE ARQUIVOS

- **Sequência de bytes:** estrutura simples, onde não existe estrutura e o arquivo é visto como sequência simples de bytes
- O SO não tem conhecimento do significado e estruturação dos campos que compõem o arquivo
- A aplicação é livre para definir todo o formato
- Muito flexível para criar diferentes estruturas de dados porém todo o controle de acesso ao arquivo é exclusivo da aplicação



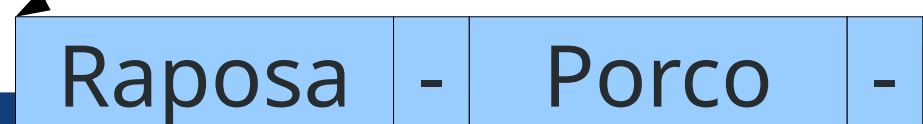
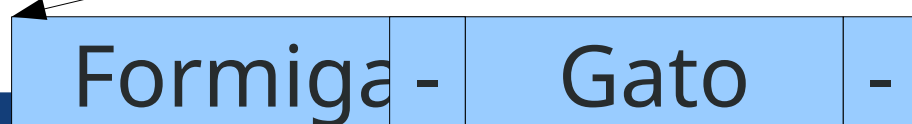
ORGANIZAÇÃO DE ARQUIVOS

- **Sequência de registros:** formato popular durante a época dos cartões perfurados
- Os arquivos são divididos em unidades de mesmo tamanho chamadas de registros
- Possui acesso otimizado por registro
- Dificuldade em utilizar os registros e há uma perda de espaço ao não se utilizar o tamanho inteiro do registro



ORGANIZAÇÃO DE ARQUIVOS

- **Árvore de registros:** o arquivo é uma árvore de registros que podem ter tamanho variável
- Cada registro é composto por um campo chave, contida em uma posição fixa. A árvore é ordenada pelo campo chave
- Busca otimizar acesso randômico a registros





TIPOS DE ARQUIVOS

- Principais tipos de arquivos:
 - Regulares;
 - Diretórios;
 - Caracteres;
 - Blocos.



TIPOS DE ARQUIVOS

- Muitos sistemas aceitam arquivos regulares, esses são arquivos que contêm informação do usuário
- Arquivos regulares são geralmente arquivos texto ASCII ou arquivos binários
- Diretórios são arquivos do sistema para manter a estrutura do sistema do arquivo



TIPOS DE ARQUIVOS

- Arquivos de caracteres são usados para modelar dispositivos de E/S seriais, como terminais, impressoras e redes
- Arquivos especiais de blocos são usados para modelar discos
- Exemplo de um arquivo de bloco (sda) e um arquivo de caractere (tty) em um sistema Linux
- Eles podem ser diferenciados pelo “c” e “b”

```
brw-rw---- 1 root disk 8, 0 May 11 13:32 /dev/sda
crw-rw-rw- 1 root tty  5, 0 Jun  5 09:23 /dev/tty
```



MÉTODOS DE ACESSO

- Nesta aula, nosso foco será arquivos regulares e de diretórios
- Dependendo da organização do arquivo, o sistema pode obter o arquivo de diferentes maneiras



MÉTODOS DE ACESSO

- Os computadores antigos utilizavam fitas magnéticas, e isso limitava o acesso que o hardware poderia realizar
- As operações eram sequenciais: o acesso à leitura dos registros era feito na ordem que eram gravados
- A gravação apenas ocorria no final do arquivo
- Porém os arquivos podem ser rebobinados e lidos inúmeras vezes



MÉTODOS DE ACESSO

- O acesso direto ou aleatório passou a existir com o advento dos discos magnéticos
- O primeiro a surgir foi o acesso direto: permite a leitura e gravação de um registro na sua posição
- Este método é realizado através do número do registro, que é sua posição relativa ao início do arquivo
- Não existe restrição sobre qual registro pode ser acessado, sendo necessário especificar o registro



MÉTODOS DE ACESSO

- É possível combinar o acesso sequencial e direto, de forma que um registro qualquer é acesso usando o modo direto e a partir dele é possível acessar sequencialmente os demais
- Em sistemas modernos existe a operação de **seek**, de busca, define qual registro deve ser lido em seguida e a operação **read**, de leitura, lê o próximo registro



ATRIBUTOS

- Cada arquivo possui informações de controle denominada atributos
- Muito dependente do sistema de arquivos
- Todos arquivos possuem um nome e uma data
- Alguns atributos são ligados à proteção, dizendo quem tem permissão de acessar o arquivo
- Outros dizem respeito a informações do sistema operacional: flag que indica se o arquivo é oculto, se um backup foi realizado recentemente, etc.

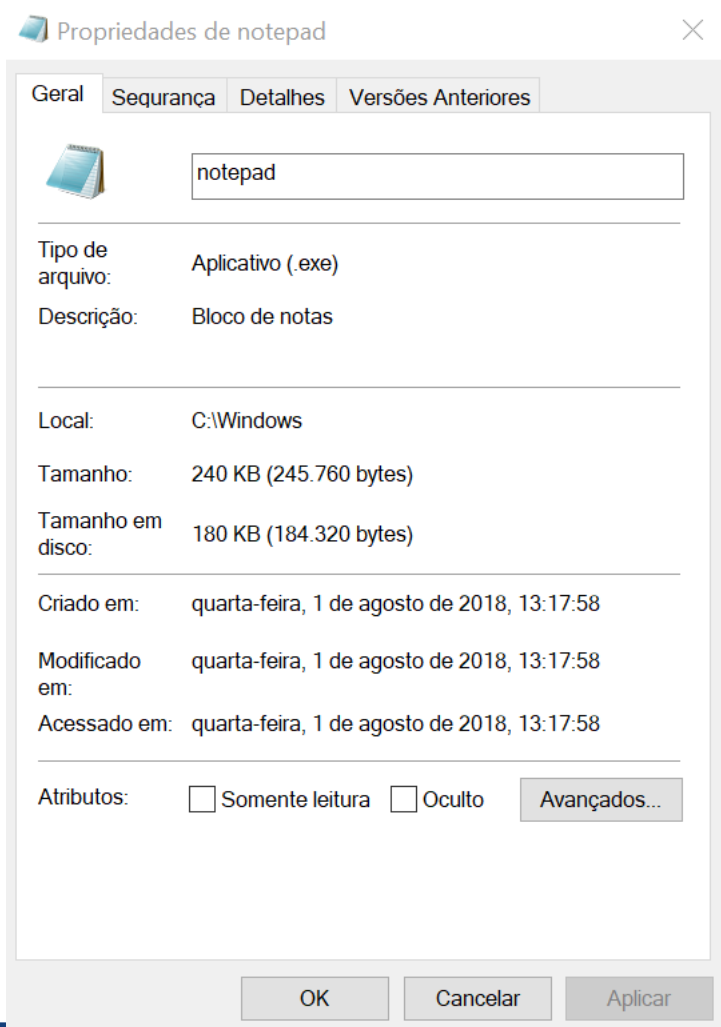


ATRIBUTOS

- Proteção: quem tem acesso ao arquivo e qual modo
- Criador: quem criou o arquivo
- Proprietário: o atual dono do arquivo
- Flag de acesso aleatório: indicando se pode ser feito acesso aleatório ou se deve ser sequencial
- Tamanho: o tamanho atual do arquivo

ATRIBUTOS

- Atributos em um arquivo Windows



Em sistemas Windows, o arquivo ser executável é uma convenção do nome (termina com .exe) e ser oculto é um atributo do arquivo.



ATRIBUTOS

- Atributos em um arquivo Linux

Em sistemas Linux, o arquivo executável é um atributo do arquivo e ser oculto é uma convenção do nome (começa com .)

```
sund@Nanifeld:~/my_dir$ ls
arquivo.txt
sund@Nanifeld:~/my_dir$ ls -la
total 0
drwxrwxrwx 1 sund sund 4096 Nov  5 15:56 .
drwxr-xr-x 1 sund sund 4096 Nov  5 15:55 ..
-rw-rw-rw- 1 sund sund   0 Nov  5 15:56 .arquivo_oculto.txt
-rw-rw-rw- 1 sund sund   0 Nov  5 15:56 arquivo.txt
sund@Nanifeld:~/my_dir$
```



ATRIBUTOS

- Em sistemas unix modernos, existem 3 permissões básicas:
 - De leitura, que permite o usuário ler dados do arquivo
 - De escrita, que permite o usuário alterar dados do arquivo
 - De execução, que permite o usuário criar um processo referenciando um arquivo como o conjunto de instruções
- O comando `chmod` altera as permissões:

```
-rw-r----- 1 sundfeld sundfeld 24 Jun  6 17:12 arq.txt  
-rwxr-x--- 1 sundfeld sundfeld 51 Jun  6 17:13 script.sh
```



ATRIBUTOS RWX

- Sistemas Unix possuem atributos de leitura/escrita/execução para usuário, grupo e outros
- Desta forma, é possível configurar permissões exclusivas para um grupo
- Colocando usuários em grupos comuns, permite que eles compartilhem arquivos entre si

```
-rw-rw-rw- 1 sund sund  0 Nov  5 15:56 arquivo.txt
```

→ Usuário (rw-)

→ Grupo (rw-)

→ Outros (rw-)



ROTEIRO

- Introdução
- Arquivos
- Diretórios
- Gerência
 - Gerência de Espaço Livre em Disco
 - Gerência de Alocação de Espaço
- Proteção de Acesso
- Implementação de Caches



DIRETÓRIOS

- Os diretórios são utilizados para organizar logicamente os arquivos contidos no disco
- O diretório armazena informações como localização física, nome, organização e demais atributos
- Ao abrir um arquivo, o SO verifica a sua entrada na estrutura de diretórios, armazenando as informações sobre atributos e localização do arquivo em uma tabela

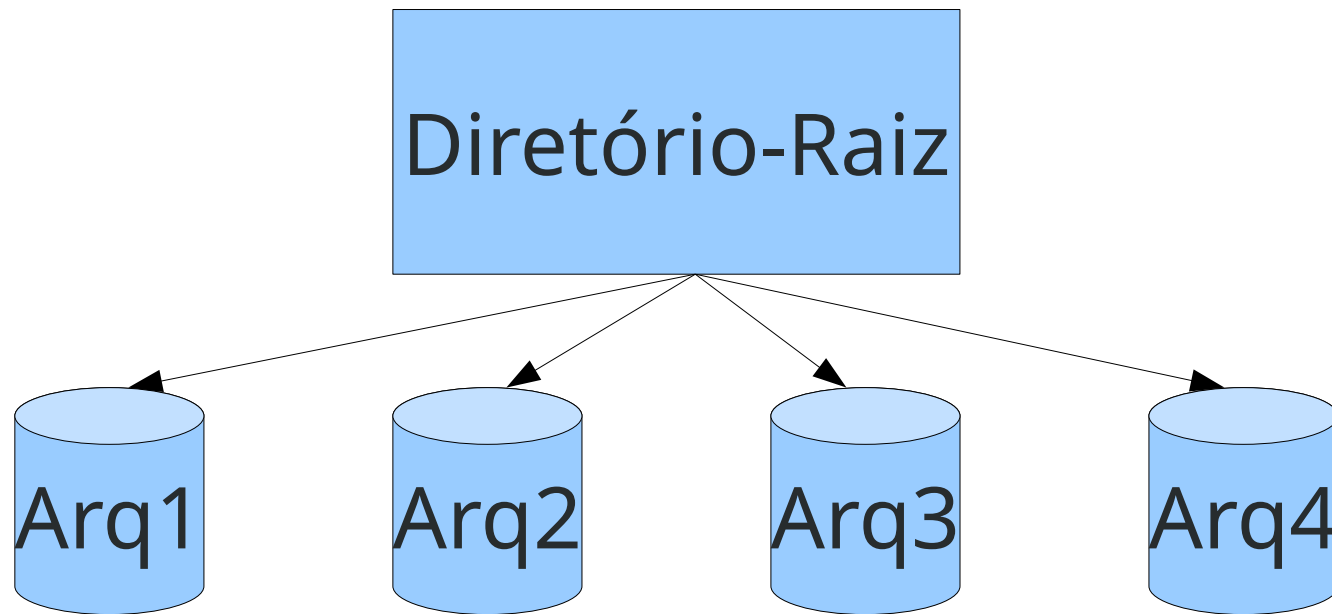


DIRETÓRIOS

- **Sistemas de diretório em nível único** são a forma mais simples de um sistema de diretórios
- Há um diretório contendo todos os arquivos, chamado de diretório raiz
- Existia nos primeiros computadores pessoais e o primeiro supercomputador do mundo
- Busca manter o design de software simples



DIRETÓRIOS



- Nos anos 2000, ainda era utilizado em dispositivos embarcados simples: câmera digital e players portáteis de música



DIRETÓRIOS

- **Sistemas de diretórios hierárquicos:** um nível é muito limitado, nos sistemas modernos esse projeto foi evoluído
- Uma hierarquia é necessária, gerando-se então uma árvore
- É possível criar um diretório dentro de outros diretórios, gerando então uma estrutura de árvore



DIRETÓRIOS

- Os sistemas de diretórios hierárquicos mais simples, organizam-se utilizando um diretório por usuário, e cada usuário tem a liberdade de criar novos diretórios



DIRETÓRIOS

- Ainda resta um problema: como organizar os arquivos do sistema?
- Bibliotecas compartilhadas, daemons de gerência de rede, arquivos temporários, entre outros.
- Os sistemas Linux modernos definem a Filesystem Hierarchy Standard (FHS)
- FHS é um padrão mantido pela Linux Foundation, inspirado pelo antigo Filesystem Standard dos sistemas Unix



FHS – ALGUNS EXEMPLOS

/	Diretório Raiz
/bin	Binários essenciais aos usuários (cat, ls)
/boot	Bootloaders e núcleos do SO
/dev	Arquivos de dispositivos
/home	Pasta para diretórios de usuários
/lib	Bibliotecas essenciais ao sistema
/mnt	Pontos de montagem temporários
/tmp	Arquivos temporários
/usr	Contém uma própria hierarquia para arquivos de somente leitura para usuários
/usr/bin	Binários não-essenciais aos usuários
/sbin	Binários essenciais ao administrador do sistema



NOMES DE CAMINHOS

- Quando o sistema é organizado em forma de árvore de diretórios, alguma maneira é necessária para especificar o nome dos arquivos
- A maneira mais simples é referência absoluta (full-path) onde o arquivo é referenciado por todo o seu nome:
 - /home/usuario/arq.txt



NOMES DE CAMINHOS

- Além disso, é necessário um separador para definir qual parte é nome de diretório ou arquivo
- Alguns separadores:
 - “/”: utilizado em Unix (/usr/share/vim/vim74)
 - “\”: utilizado em Windows (C:\Windows\System32\notepad.exe)
 - “>”: utilizado no antigo MULTICS:
“>usr>ast>caixapostal”



NOMES DE CAMINHOS

- No **nome de caminho absoluto**, o primeiro caractere sempre é o separador
- No entanto, é comum usar o **nome de caminho relativo**, onde é utilizado o conceito de **diretório de trabalho atual**
- Uma sessão shell do usuário “estagiario” inicia-se no diretório `/home/estagiario`



NOMES DE CAMINHOS

- Com isso, os nomes relativos (que não se iniciam com o caractere separador) são combinados com o diretório atual de trabalho
- Para nosso usuário, estagiario, após iniciar a sessão de shell, os comandos são equivalentes:
 - `cp 1.txt todo.txt`
 - `cp /home/estagiario/1.txt
/home/estagiario/todo.txt`



NOMES DE CAMINHOS

- O diretório atual de trabalho pode ser alterado através do comando “cd”:
 - `cd tarefas` → Diretório alterado para `/home/estagiario/tarefas`
- É possível referenciar diretório um nível abaixo usando “..” e o atual com “.”. Considerando diretório de trabalho sendo `/home/estagiario/tarefas`, são equivalentes:
 - `cp ../arq.txt .`
 - `cp /home/estagiario/arq.txt /home/estagiario/tarefas/`



ROTEIRO

- Introdução
- Arquivos
- Diretórios
- Gerência
 - Gerência de Espaço Livre em Disco
 - Gerência de Alocação de Espaço
- Proteção de Acesso
- Implementação de Caches



GERÊNCIA DE ARQUIVOS

- O SO precisa implementar duas soluções no sistema de arquivos:
- **Gerência de Espaço Livre em Disco:**
responsável por manter informações de quais blocos não estão sendo utilizados no disco
- **Gerência de Alocação de Espaço:**
responsável por dizer quais blocos pertencem a quais arquivos



GERÊNCIA DE ESPAÇO LIVRE EM DISCO

- Da mesma forma que a gerência de memória, o sistema de arquivos precisa implementar políticas para gerenciar o espaço livre em disco
- Neste aspecto, diversos dos conceitos que estudamos em gerência de memória se aplicam a sistema de arquivos



GERÊNCIA DE ESPAÇO LIVRE EM DISCO

- Para isso, existe alguma estrutura de dados que armazenam as informações dos blocos livres no sistema
- Ao deletar um arquivo, todos os blocos alocados são passados para a lista de blocos livres
- A forma mais simples, é implementar através da tabela de mapa de bits
- Cada entrada da tabela, possui um bit 0 ou 1, indicando se o bloco está alocado



GERÊNCIA DE ESPAÇO LIVRE EM DISCO

- Deve-se decidir apropriada o tamanho do “bloco de disco”, considerando inclusive as características dos hardwares mais comuns
- Um bit representa um bloco em disco
- Operações bit a bit são caras e costumam ser relativamente lentas



GERÊNCIA DE ESPAÇO LIVRE EM DISCO

- Uma outra maneira é realizar o controle com uma estrutura de lista encadeada de todos os blocos livres do disco
- Cada bloco possui uma área reservada para armazenamento do endereço do próximo bloco
- A partir do primeiro bloco livre, é possível o acesso sequencial aos demais de forma encadeada





GERÊNCIA DE ESPAÇO LIVRE EM DISCO

- Uma terceira solução é a tabela de blocos livres, implementada com alocação contígua, onde um espaço contínuo no disco é reservado
- Esses blocos são alocados e liberados simultaneamente
- O disco é então um conjunto de segmentos de blocos livres
- Assim, mantém-se uma tabela com o endereço do primeiro bloco de cada segmento e o número de bloco livres que o seguem



GERÊNCIA DE ALOCAÇÃO DE ESPAÇO

- Além de gerenciar os espaços livres, também é muito importante gerenciar o espaço ocupado pelos arquivos, para isso temos diversas estratégias:
 - Alocação Contígua;
 - Alocação Encadeada;
 - Alocação Indexada.



ALOCAÇÃO CONTÍGUA

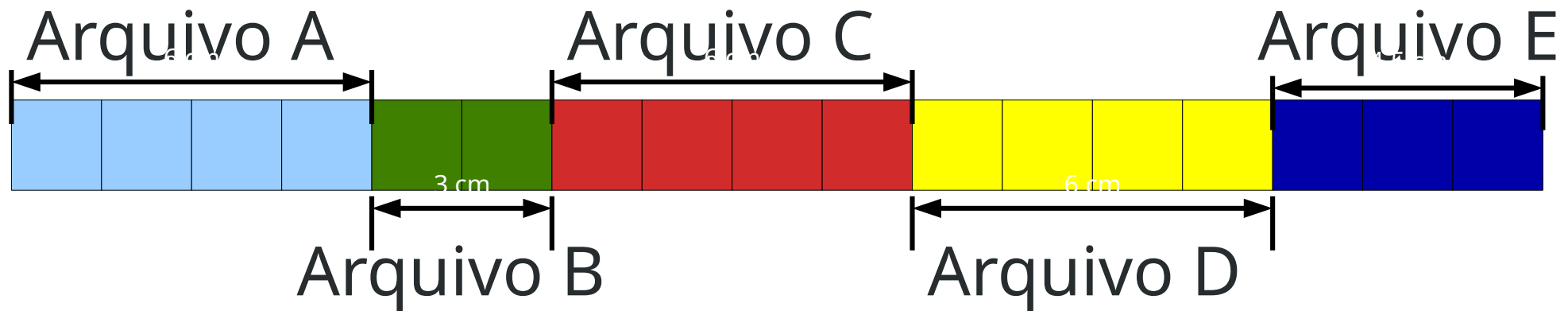
- Na alocação contígua, um arquivo é armazenado sequencialmente no dispositivo
- O sistema pode então localizar arquivos através do endereço do primeiro bloco e do seu tamanho

Arquivo	Bloco	Tamanho
Nota.txt	4	9
Todo.txt	15	1
a.out	20	2



ALOCACÃO CONTÍGUA

Alocação contígua com 5 arquivos



Após os arquivos B e D serem removidos



ALOCAÇÃO CONTÍGUA

- Na alocação contígua o acesso a arquivos é muito simples e direto de ser implementado
- Seu problema está na alocação de espaço livre para novos arquivos
- Para arquivos que estão tendo seu tamanho alterado, é necessário que exista uma quantidade de blocos livres, senão a operação falha



ALOCAÇÃO CONTÍGUA

- O que fazer quando é necessário alocar um arquivo?
- Diferentes estratégias podem ser usadas onde colocar o novo arquivo:
 - First-Fit: o arquivo será colocado no primeiro segmento livre. Busca sequencial e interrompida assim que se encontra um espaço livre
 - Best-Fit: o arquivo é colocado no menor segmento livre disponível. Toda a lista é percorrida.
 - Worst-Fit: arquivo colocado no maior segmento livre



ALOCAÇÃO CONTÍGUA

- Todas essas estratégias levam ao problema da fragmentação: área livre disponível entre os blocos de arquivos e que podem ser subaproveitadas
- No sistema de arquivos, esse processo é chamado de desfragmentação e possui aquelas desvantagens que o processo de compactação da memória
- É um processo demorado e precisa ser realizado com frequência pois não existe solução permanente



ALOCAÇÃO CONTÍGUA

- Um outro problema é o fato de o tamanho de um arquivo não ser normalmente conhecido no momento de criação, mas variar de acordo com o uso
- Por isso, os sistemas operacionais modernos utilizam estratégias mais dinâmicas para alocação de disco

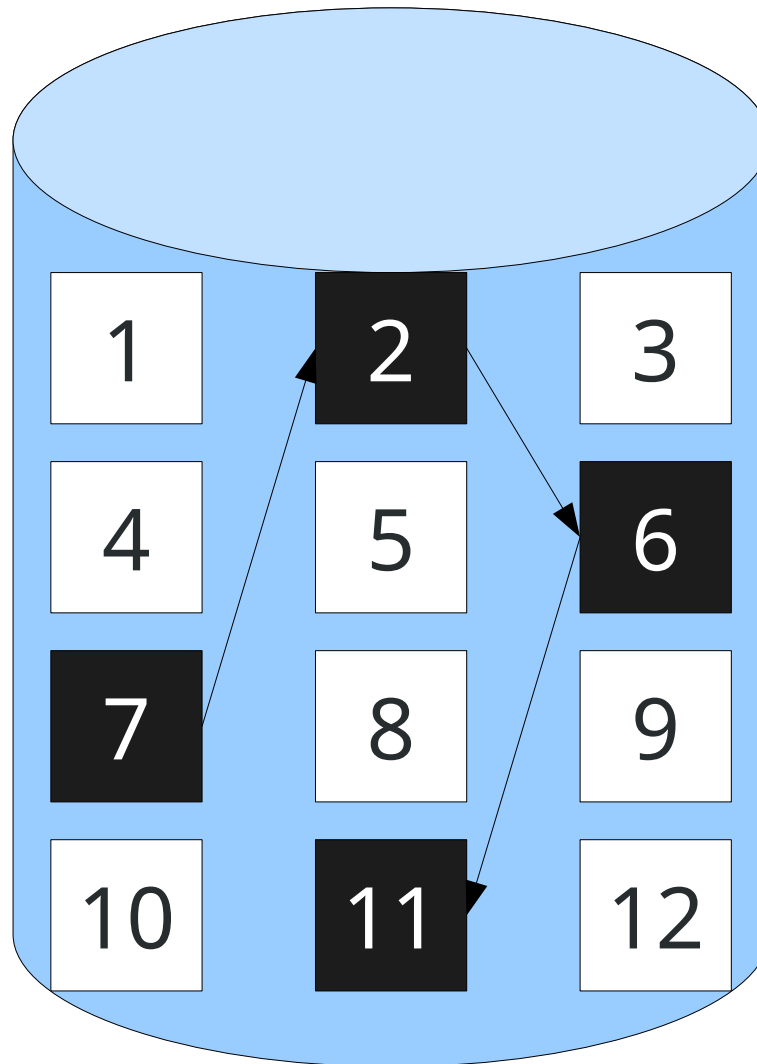


ALOCAÇÃO ENCADEADA

- Na alocação encadeada, um arquivo é organizado como um conjunto de blocos ligados logicamente no disco, independente de sua localização física
- Cada bloco deve possuir um ponteiro para o bloco seguinte do arquivo e assim sucessivamente
- Blocos livres e ocupados não precisam estar contínuos
- É mantida uma tabela de arquivos, contendo o primeiro bloco deles



ALOCAÇÃO ENCADEADA



Arquivo	Bloco
Nota.txt	7



ALOCAÇÃO ENCADEADA

- No entanto essa estratégia possui uma grande dificuldade para realizar operações de seek
- O acesso ao arquivo deve ser sequencial
- Características do hardware fazem que o acesso a blocos contínuos no disco seja melhor do que diversos blocos diferentes
- Por isso o procedimento de desfragmentação ainda pode ser utilizado para agrupar os blocos pertencentes ao mesmo arquivo

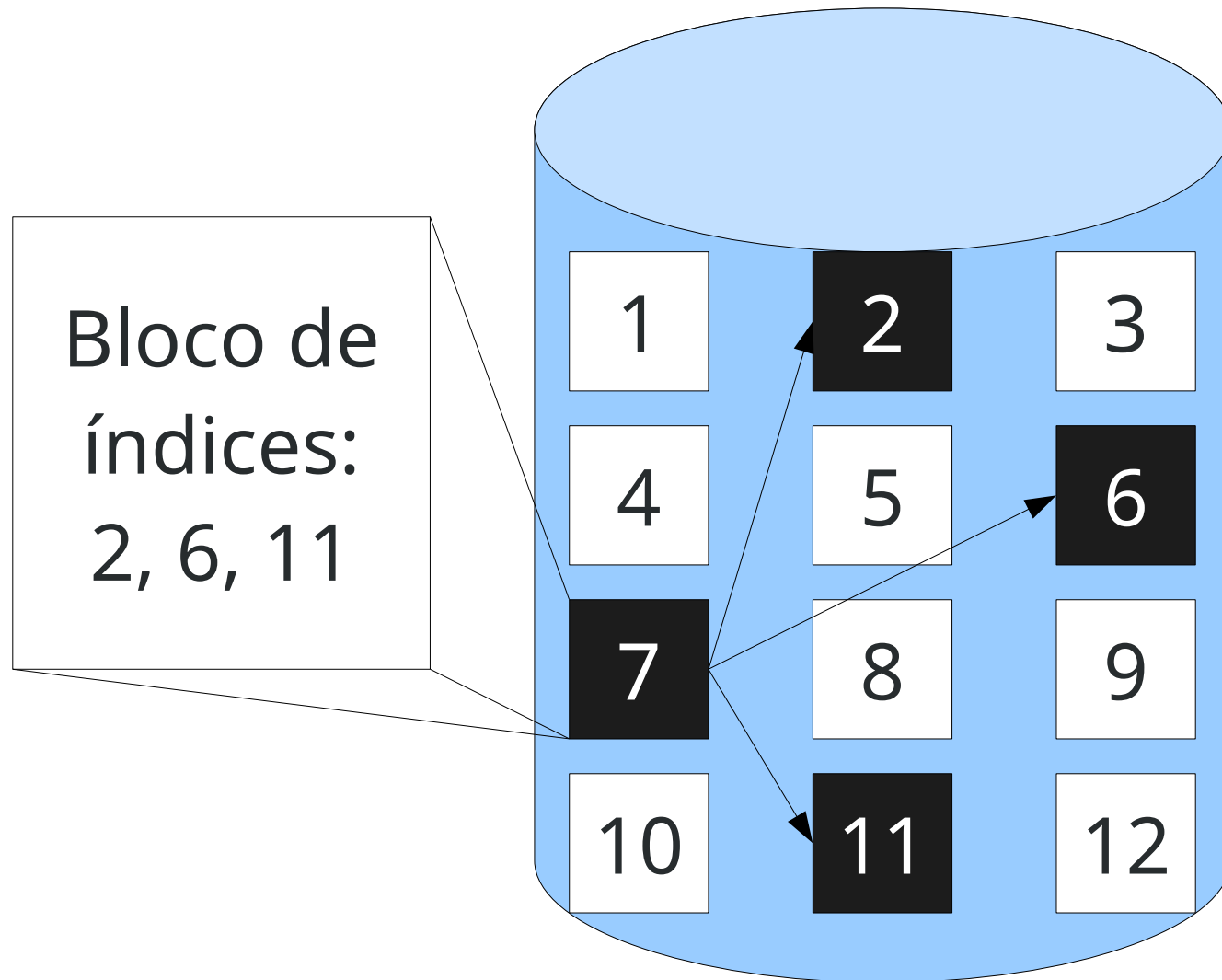


ALOCAÇÃO INDEXADA

- A alocação indexada resolve o problema da lista encadeada para se obter o acesso direto a apenas partes do arquivos
- O primeiro bloco é na verdade um bloco de índices, que contém a informação de todos os blocos usados pelo arquivos
- Também conhecida como I-Nodos



ALOCAÇÃO INDEXADA



Arquivo	Bloco
Nota.txt	7



ROTEIRO

- Introdução
- Arquivos
- Diretórios
- Gerência
 - Gerência de Espaço Livre em Disco
 - Gerência de Alocação de Espaço
- Proteção de Acesso
- Implementação de Caches



PROTEÇÃO DE ACESSO

- É responsabilidade do sistema operacional gerenciar e proteger os arquivos dos diversos usuários envolvidos
- Para evitar que os usuários escrevam diretamente no HD, existe um bit de proteção na CPU, dizendo se o processo que está executando é de usuário ou do sistema operacional



PROTEÇÃO DE ACESSO

- Proteção de acesso por atributo (rwx): o sistema operacional lê os atributos do arquivo para decidir se o usuário pode ler/escrever/executar
- Proteção de Acesso por senha de acesso: alguns sistemas de arquivos permitem criptografar todo o sistema



PROTEÇÃO DE ACESSO

- Porém, um usuário mal intencionado pode retirar o HD e instalar em outro computador, ou pode iniciar um SO que ele tem permissão de administrador através de um pendrive
- Se o sistema de arquivos é criptografado, apenas é possível acessar arquivos desse sistema mediante a apresentação de uma senha pré-cadastrada



PROTEÇÃO DE ACESSO

- Instruções que causam mudanças no disco rígido são instruções privilegiadas que só podem ser executadas pelo SO
- As manipulações de arquivos são feitas por chamadas de sistemas
- Ao se alterar um arquivo, o controle da CPU é dado ao SO
- Antes de efetuar a operação, o SO pode verificar todas as permissões antes das alterações



PROTEÇÃO DE ACESSO

- Para o usuário modificar um arquivo permanentemente ele deve realizar através de uma chamada de sistema
- Durante uma chamada de sistema, o controle da CPU é passado para o SO
- Assim, o SO pode verificar as permissões da operação requisitada, antes de executá-la



OPERAÇÕES COM ARQUIVOS

- São as principais chamadas de sistema de manipulação de arquivo:
- **Create:** um arquivo é criado sem dados.
Anunciar que o arquivo está vindo e deve-se estabelecer os atributos padrão
- **Delete:** o arquivo não é mais necessário e deve ser removido do sistema de arquivos



OPERAÇÕES COM ARQUIVOS

- **Open:** Antes de usar um arquivo, o usuário deve especificar qual é, de forma que o SO verifica as permissões e atributos do arquivo
- **Close:** Quando todos os acessos são concluídos o usuário informa ao SO que não deseja mais utilizar o arquivo. Muitos sistemas limitam o número máximo de arquivos (ou descritores) que um processo pode manter aberto simultaneamente



OPERAÇÕES COM ARQUIVOS

- **Read:** os dados são lidos do arquivo, normalmente especificando a quantidade de dados a serem lidos e são lidos da posição atual
- **Write:** os dados são escritos para o arquivo, normalmente na posição final e incrementando o tamanho do arquivo (Pela posição EOF). Porém os dados podem ser escritos no meio do arquivo e com isso eles sobrescrevem os dados antigos que são perdidos



OPERAÇÕES COM ARQUIVOS

- **Append:** chamada é uma forma restrita de write, onde os dados são exclusivamente adicionados ao fim do arquivo
- **Seek:** implementados em arquivos de acesso aleatório para alterar a posição atual do arquivo para uma determinada pelo usuário
- **Get attributes:** retorna os atributos conhecidos pelo sistema de arquivo, como permissões ou data de alteração/criação dos arquivos



OPERAÇÕES COM ARQUIVOS

- **Set attributes:** alguns dos atributos podem ser alterados pelo usuário. Por exemplo, o comando UNIX “touch” atualiza a data de modificação de um arquivo
- **Rename:** com frequência o nome de um arquivo deve ser alterado, para isso é utilizada a operação de renomear



SISTEMAS DE ARQUIVOS

- Os sistemas operacionais modernos possuem suporte a vários sistemas de arquivos
- O Windows 95 e 98 usavam o mesmo sistema de arquivo do MS-DOS chamado de FAT-16
- O Windows 98 introduziu mudanças no sistema, que passou a ser chamado FAT-32
- A partir do ano 2000, as versões de Windows (NT, Windows 10), usam um sistema de arquivos modernos chamado de NTFS (native file system)



SISTEMAS DE ARQUIVOS

- No Linux, a primeira versão utiliza o sistema de arquivos do MINIX 1. Depois foi substituída por um novo sistema de arquivos chamado de ext, que permitia tantos nomes de arquivos mais longos e arquivos maiores
- O ext foi substituído pelo ext2, versão mais rápida dele



SISTEMAS DE ARQUIVOS

- Quedas de energia podem causar a perda de arquivos permanentemente
- Para melhorar a robustez, o Linux conta com sistemas de arquivos com diários (journaling)
- A primeira versão foi o ext3, que rapidamente foi substituída pelo ext4, permite arquivos maiores e tamanho de sistemas de arquivos globais maiores

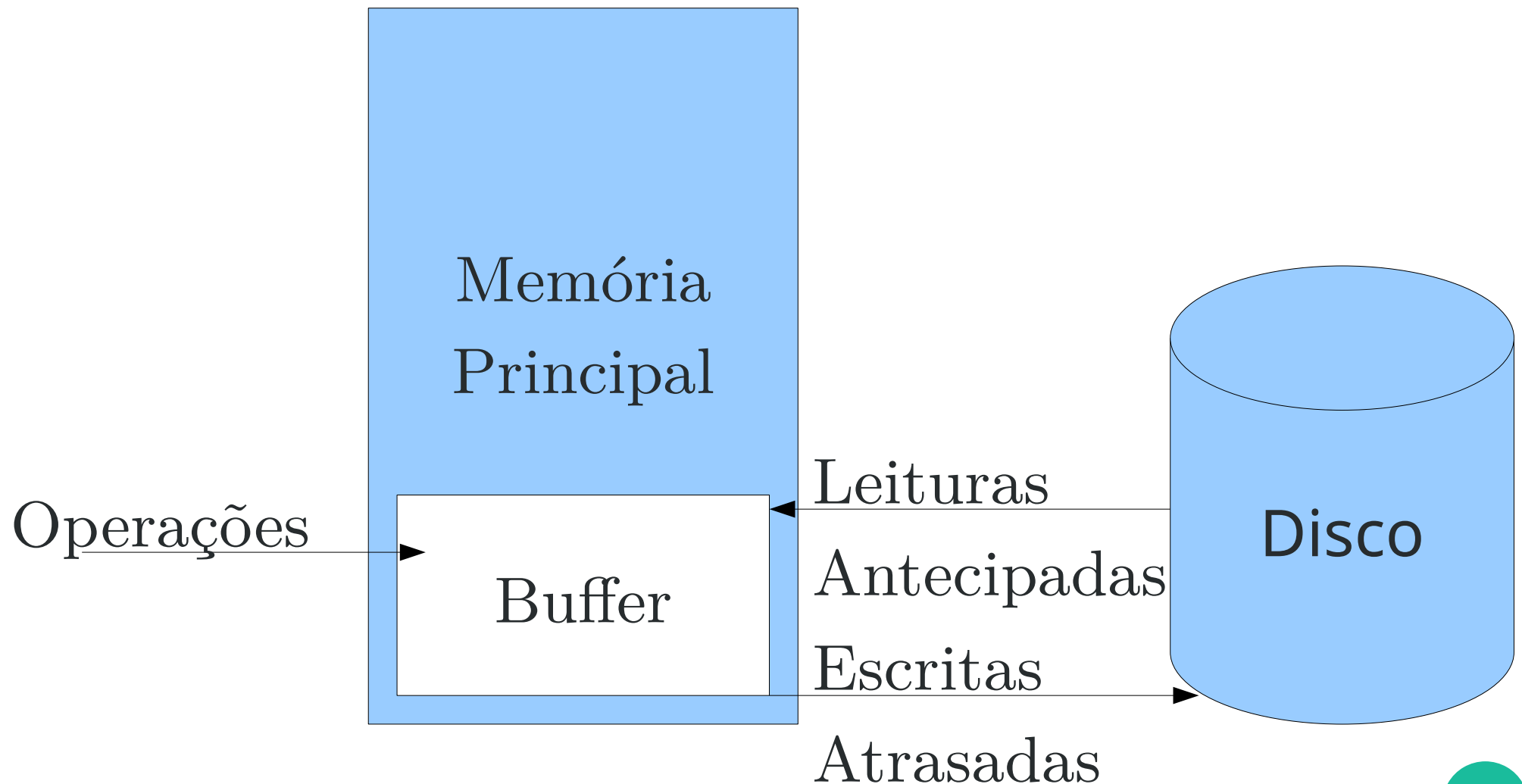


IMPLEMENTAÇÃO DE CACHES

- Como vimos nas aulas passadas, o acesso a disco é muito mais lento que a memória
- Com isso, a técnica de buffering (cache de blocos) pode ser utilizada para minimizar o tempo de acesso
- Nela, uma área da memória principal é reservada para operações pendentes de entrada e saída, mas o processo é desbloqueado e pode prosseguir sem se bloquear aguardando o término da operação de E/S



IMPLEMENTAÇÃO DE CACHES





IMPLEMENTAÇÃO DE CACHES

- Em um cache, as leituras não são realizadas diretamente no disco. São realizadas no buffer de cache. Se o bloco está presente, a leitura é rápida
- Caso contrário, o bloco é carregado do disco para o cache e as referências futuras são feitas mais rapidamente



IMPLEMENTAÇÃO DE CACHES

- As operações de escritas são realizadas com escritas atrasadas. Ao invés de esperar a operação do disco, a operação é realizada no buffer
- Então, os dados do buffer são copiados para o disco pelo SO, mas a operação de escrita não gera um bloqueio (escritas atrasadas)



IMPLEMENTAÇÃO DE CACHES

- No entanto o cache pode se tornar muito alto
- E o espaço é limitado: podemos aplicar os algoritmos de substituição de páginas para substituir blocos do cache
- No entanto, referência a blocos do cache são mais raras que referências a memória, os algoritmos mais lentos podem ser utilizados na prática



REFERÊNCIAS

- Capítulo 4 – TANENBAUM, A. S. *Sistemas Operacionais Modernos*. 4ª ed. Prentice Hall, 2016.
- Capítulo 11 – MACHADO, F. B.; MAIA, L. P. *Arquitetura de Sistemas Operacionais*. 5ª ed. Rio de Janeiro: LTC, 2013.