

## Sistemas Operacionais

Marcos Laureano

---

---

---

---

---

---

---

---

## Roteiro

- Arquivos
- Organização de arquivos
- Diretórios
- Métodos de acesso
- Métodos de alocação em disco
- Fragmentação
- Tamanho de bloco
- Proteção de acesso

---

---

---

---

---

---

---

---

## 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.

---

---

---

---

---

---

---

---

## Arquivos

- 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.

4/68

---

---

---

---

---

---

---

---

## Idéias Básicas

- A representação da informação dentro de um computador *digital* é feita na forma de *números*.
- No Linux ou no Windows, arquivos são concebidos e tratados como STREAMS (minhocas) de BYTES.
- O que é um BYTE? Um BYTE é um número entre 0 e 255.
  - Um BYTE é um conjunto de 8 bits (dígitos binários)

5/68

---

---

---

---

---

---

---

---

## 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

6/68

---

---

---

---

---

---

---

---

## 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 que eu *considero* como 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*

7/68

---

---

---

---

---

---

---

---

---

---

## Representação de um arquivo

Visualizando o arquivo no meu **Terminal de Vídeo**:

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

↑  
 ↑  
 ↑  
 ↑

```

$ cat arquivo
ABC
123
DE
ab
$
  
```

8/68

---

---

---

---

---

---

---

---

---

---

## Arquivos

- 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.

9/68

---

---

---

---

---

---

---

---

---

---

## Extensão de Arquivos

Extensão	Descrição
arquivo.c	Arquivo fonte em C
arquivo.cob	Arquivo fonte em Cobol
arquivo.exe	Arquivo executável
arquivo.obj	Arquivo objeto
arquivo.pas	Arquivo fonte em Pascal
arquivo.txt	Arquivo texto
arquivo.mp3	Arquivo de música
arquivo.dll	Arquivo de biblioteca dinâmica

10/68

---

---

---

---

---

---

---

---

## Organização de Arquivos

- Consiste no modo como seus dados estão internamente armazenados.
- Quando o arquivo é criado pode-se definir que organização será adotada que pode ser uma estrutura suportada pelo sistema operacional ou definida pela própria aplicação.
- A forma mais simples de organização de arquivos é através de uma seqüência não-estruturada de bytes, onde o sistema de arquivos não impõe nenhuma estrutura lógica para os dados, a aplicação deve definir toda a organização.

11/68

---

---

---

---

---

---

---

---

## 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 a seqüencial, relativa e indexada.

12/68

---

---

---

---

---

---

---

---

## Organização de Arquivos

- Nestes tipos de organização, podemos visualizar um arquivo como um conjunto de registros.
- Quando definidos sempre com o mesmo tamanho são chamados de registros de tamanho fixo e caso contrário são chamados de registros de tamanho variável.

13/68

---

---

---

---

---

---

---

---

## Organização de Arquivos

14/68

---

---

---

---

---

---

---

---

## Métodos de Acesso

- Em função de como o arquivo esta 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 do arquivo. Pode-se combinar o acesso seqüencial com o direto e com isso acessar diretamente um arquivo e os demais m forma seqüencial.

15/68

---

---

---

---

---

---

---

---

## 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 a ordem em que os registros são lidos ou gravados. Somente é possível quando é definido com registros de tamanho fixo.
- 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.

16/68

---

---

---

---

---

---

---

---

## Acesso Direto



17/68

---

---

---

---

---

---

---

---

## 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 tem como função oferecer uma interface simples e uniforme entre a aplicação e os diversos dispositivos.

18/68

---

---

---

---

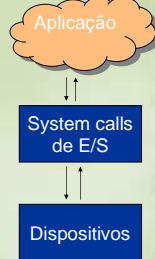
---

---

---

---

## Operações de Entrada / Saída



19/68

---

---

---

---

---

---

---

---

## 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

20/68

---

---

---

---

---

---

---

---

## Atributos

- São informações de controle que dependendo do sistema de arquivos variam, porém 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 existe alguns que podem ser alterados pelo usuário tais como proteção, tamanho e senha.

21/68

---

---

---

---

---

---

---

---

## 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

22/68

---

---

---

---

---

---

---

---

## Exemplo de Atributos

```
$ ls -l
drwxr-xr-x 5 laureano prof 4096 Abr 14 11:34 a
drwxr-xr-x 3 laureano prof 4096 Mar 31 12:25 abc
-rw-r--r-- 1 laureano prof 647 Abr 28 12:24 arqcut1
-rw-r--r-- 1 laureano prof 2335 Abr 28 12:24 arqgrep
drwxr-xr-x 11 laureano prof 4096 Mai 30 2005 arquivos
drwxr-xr-x 2 laureano prof 4096 Mai 18 2005 asu
drwx----- 11 laureano prof 4096 Jul 5 2004 bash-2.05b
-rw-r--r-- 1 laureano prof 8437760 Jul 5 2004 bash5.tar
drwxr-xr-x 2 laureano prof 4096 Mai 22 2005 c
-rw----- 1 laureano prof 1470 Set 28 2004 client.c
```

23/68

---

---

---

---

---

---

---

---

## 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.

24/68

---

---

---

---

---

---

---

---



## Diretórios

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

25/68

---

---

---

---

---

---

---

---

## Estrutura de diretórios de nível único

Identificação  
 Proteção  
 Organização  
 Localização física  
 Atributos

Diretório                  Arquivos

26/68

---

---

---

---

---

---

---

---

## 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 isto é chamado de *path* (caminho).

27/68

---

---

---

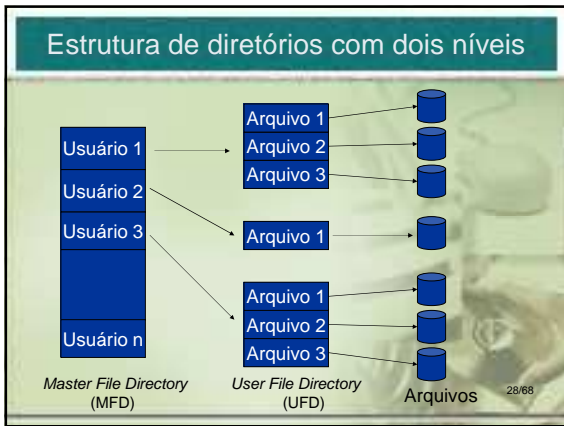
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

- ### 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 e na maioria dos sistemas os diretórios são tratados como arquivos tendo atributos e identificação.
- 29/68

---

---

---

---

---

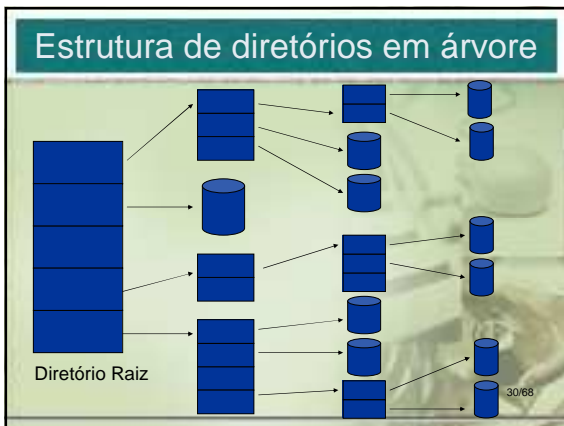
---

---

---

---

---




---

---

---

---

---

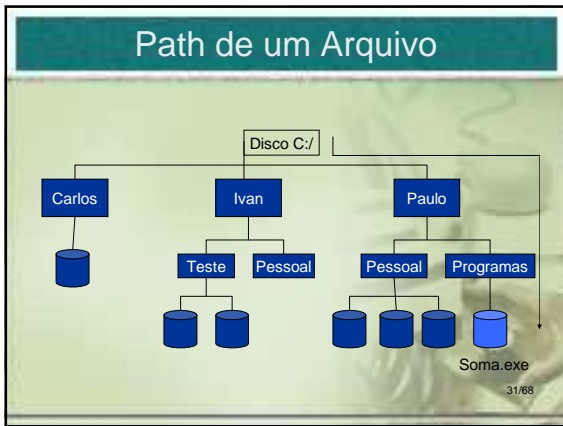
---

---

---

---

---




---

---

---

---

---

---

---

---

### Alocação de Espaço em Disco

- A criação de arquivos exige que o sistema operacional tenha controle de quais áreas ou blocos no disco estão livres e este controle é realizado através de uma estrutura (geralmente lista ou tabela) de dados que armazenam informações e possibilitam ao sistema de arquivos gerenciar o espaço livre.

32/68

---

---

---

---

---

---

---

---

### Alocação de Espaço em Disco

- A forma mais simples de implementar uma estrutura de espaços livres é através de uma tabela chamada mapa de bits (*bit map*) onde cada entrada da tabela é associada a um bloco e representado por um bit, que pode assumir valor igual a 0 (bloco livre) ou 1 (bloco alocado).
- Esta estrutura gera um gasto excessivo de memória já que para cada bloco deve existir uma entrada na tabela.

33/68

---

---

---

---

---

---

---

---

## Alocação de Espaço em Disco

- Outra forma é realizar o controle por meio da ligação encadeada de todos os blocos livres e cada bloco deve possuir uma área reservada para armazenamento do endereço do próximo.
- A partir do primeiro bloco pode-se ter acesso seqüencial aos demais de forma encadeada. Apresenta restrições se considerarmos que o algoritmo de busca de espaço livre sempre deve realizar uma pesquisa seqüencial na lista.

34/68

---

---

---

---

---

---

---

---

## Alocação de Espaço em Disco

- Outra solução leva em conta que blocos contíguos são geralmente alocados ou liberados simultaneamente, com base neste conceito é possível manter uma tabela com o endereço do primeiro bloco de cada segmento e o número de blocos livres contíguos que se seguem.

35/68

---

---

---

---

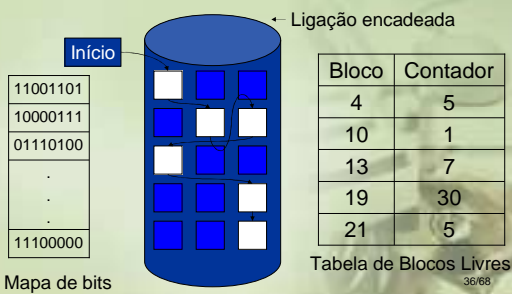
---

---

---

---

## Alocação de Espaço em Disco



---

---

---

---

---

---

---

---

## Alocação Contígua

- Consiste em armazenar um arquivo em blocos seqüencialmente dispostos. Neste tipo, o sistema localiza um arquivo através do endereço do primeiro bloco e da sua extensão em blocos.
- O acesso é bastante simples tanto para a forma seqüencial tanto para a direta, seu principal problema é a alocação de novos arquivos nos espaços livres, pois para colocar **n** blocos é necessário que se tenha uma cadeia com **n** blocos dispostos seqüencialmente no disco. 37/68

---

---

---

---

---

---

---

---

## Alocação Contígua

- Existe alguns problemas como determinar o espaço necessário a um arquivo quando é criado e depois pode existir a necessidade de extensão e esta é uma operação complexa, sendo assim a pré-alocação seria uma solução mas pode ocasionar que parte do espaço alocado permaneça ocioso por um logo período de tempo.
- Quando o sistema operacional deseja alocar espaço para um novo arquivo, pode existir mais de um segmento livre disponível com o tamanho exigido e é necessário alguma estratégia de alocação seja adotada para selecionar qual segmento deve ser escolhido. 38/68

---

---

---

---

---

---

---

---

## Alocação Contígua

arquivo	inicio	#blocos
readme.txt	010	003
prova.doc	002	008
Aula.pdf	017	005

0 1 2 3 4 5 6 7

39/68

---

---

---

---

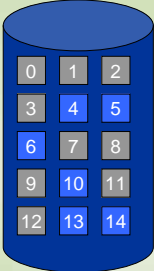
---

---

---

---

## Alocação Contígua



Arquivo	Bloco	Extensão
A. TXT	4	3
B. TXT	10	1
C. TXT	13	2

40/68

---

---

---

---

---

---

---

---

## Alocação Contígua

- Analisaremos as três principais estratégias:
  - First-fit: o primeiro segmento livre com tamanho suficiente para alocar o arquivo é selecionado. A busca na lista é seqüencial, sendo interrompida tão logo se encontre um segmento adequado.
  - Best-fit: seleciona o menor segmento livre disponível com tamanho suficiente para armazenar o arquivo. A busca em toda a lista se faz necessária para a seleção do segmento, a não ser que a lista esteja ordenada por tamanho.
  - Worst-fit: o maior segmento é alocado e a busca por toda a lista se faz necessária, a menos que exista uma ordenação por tamanho.

41/68

---

---

---

---

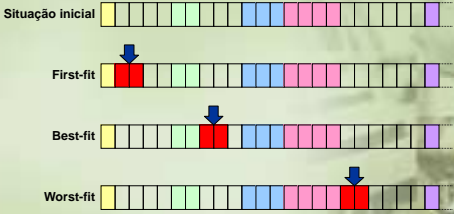
---

---

---

---

## Alocando um arquivo c/ 2 blocos



42/68

---

---

---

---

---

---

---

---

## Alocação Contígua

- Independente da estratégia utilizada, a alocação apresenta um problema chamado fragmentação de espaços livres. O problema pode se tornar crítico quando um disco possui blocos livres disponíveis, porém sem um segmento contíguo onde o arquivo possa ser alocado.
- Deve ser feita a desfragmentação periodicamente (visando que este problema seja resolvido) para reorganizar os arquivos no disco a fim de que exista um único segmento de blocos livres. Há um grande consumo de tempo neste processo e tem efeito temporário.

43/68

---

---

---

---

---

---

---

---

## Fragmentação externa

- Espaços vazios **entre** blocos de arquivos.
- À medida que o sistema evolui:
  - arquivos são criados e removidos
  - mais espaços vazios aparecem.
  - os espaços vazios ficam menores.

→ Alocar novos arquivos torna-se difícil !

44/68

---

---

---

---

---

---

---

---

## Evolução da fragmentação

Agora, como alocar um arquivo com 4 blocos ?

45/68

---

---

---

---

---

---

---

---

## Desfragmentação

- Mover arquivos para reagrupar os fragmentos em espaços maiores
- Visa permitir alocar arquivos maiores
- Deve ser feita periodicamente
- Uso de algoritmos para minimizar movimentação de arquivos (rapidez)

46/68

---

---

---

---


---


---


---


---

## Estratégias de desfragmentação

Situação inicial 

Moveu 6 blocos 

Moveu 4 blocos 

Moveu 2 blocos 

47/68

---

---

---

---

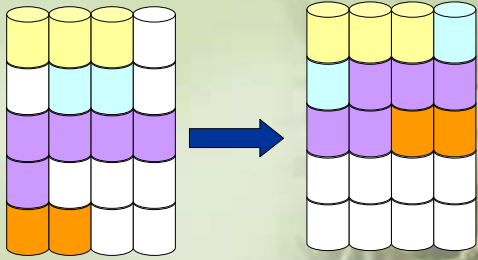
---

---

---

---

## Defragmentação



48/68

---

---

---

---

---

---

---

---



## Alocação Encadeada

- O arquivo é organizado como um conjunto de blocos ligados no disco, independente de sua localização física e cada um deve possuir um ponteiro para o bloco seguinte.
- O que ocorre neste método é a fragmentação de arquivos (quebra do arquivo em diversos pedaços denominados extents) o que aumenta o tempo de acesso ao arquivo, pois o disco deve deslocar-se diversas vezes para acessar todas as extents.
- É necessário que o disco seja desfragmentado periodicamente, esta alocação só permite acesso seqüencial e desperdiça espaço nos blocos com armazenamento de ponteiros.

49/68

---

---

---

---

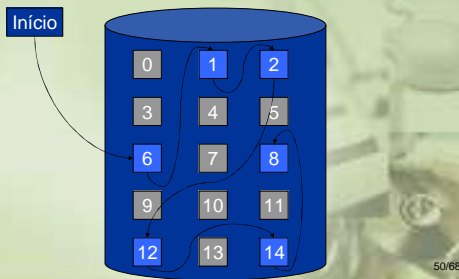
---

---

---

---

## Alocação Encadeada



50/68

---

---

---

---

---

---

---

---

## Alocação Encadeada

arquivo	inicio	#blocos
readme.txt	010	003
prova.doc	002	008
Aula.pdf	017	005



51/68

---

---

---

---

---

---

---

---

## Alocação Indexada

- O princípio desta técnica é manter os ponteiros de todos os blocos de arquivos em uma única estrutura denominada bloco de índice.
- Além de permitir o acesso direto aos blocos do arquivo, não utiliza informações de controle nos blocos de dados como existe na alocação encadeada.

52/68

---

---

---

---

---

---

---

---

## Alocação Indexada

53/68

---

---

---

---

---

---

---

---

## Alocação Indexada

arquivo	inicio	#blocos
readme.txt	010	003
prova.doc	002	008
Aula.pdf	017	005

54/68

---

---

---

---

---

---

---

---

## Fragmentação interna

- Arquivos são alocados em blocos:
  - Os blocos têm tamanho fixo.
  - Entre 512 bytes e 8 Kbytes.
  - Um bloco não pode ser alocado parcialmente.
- Se usarmos blocos de 4096 bytes:
  - um arquivo de 5700 bytes ocupará 2 blocos.
  - 2492 bytes serão perdidos no último bloco.
- Em média, perde-se 1/2 bloco por arquivo.

55/68

---

---

---

---

---

---

---

---

## Tamanho dos blocos

- A escolha do tamanho dos blocos é importante para a eficiência do sistema.
- Blocos pequenos:
  - menor perda por fragmentação interna
  - mais blocos por arquivo: maior custo de gerência
- Blocos grandes:
  - maior perda por fragmentação interna
  - menos blocos por arquivo: menor custo de gerência

56/68

---

---

---

---

---

---

---

---

## Proteção de Acesso

- Considerando que os meios de armazenamento são compartilhados é necessário ter mecanismos de proteção para garantir a proteção de arquivos e diretórios.
- Qualquer sistema de arquivos deve possuir mecanismos próprios para proteger o acesso as informações gravadas e o tipo de acesso é mediante concessão ou não de acessos que podem ser realizados como a leitura (read), gravação (write), execução (execute) e eliminação (delete).

57/68

---

---

---

---

---

---

---

---

## Proteção de Acesso

- Há diferenças entre o controle de acesso a diretórios e arquivos. O controle da criação/eliminação de arquivos nos diretórios, visualização do seu conteúdo e eliminação do próprio diretório são operações que também devem ser protegidas.
- Existem diferentes mecanismos e níveis de proteção e para cada tipo de sistema um modelo é mais adequado do que o outro.

58/68

---

---

---

---

---

---

---

---

## Tipos de Acesso

Acesso	Descrição
Leitura	Qualquer tipo de operação em que o arquivo possa ser visualizado, como a exibição de seu conteúdo, edição ou cópia de um novo arquivo
Gravação	Alteração no conteúdo do arquivo, como inclusão ou alteração de registros.
Execução	Associado a arquivos executáveis ou arquivos de comandos, indicando o direito de execução do arquivo.
Eliminação	Permissão para se eliminar um arquivo.

59/68

---

---

---

---

---

---

---

---

## Senha de Acesso

- É bastante simples e se resume ao usuário ter conhecimento da senha e a liberação do acesso ao arquivo concedida pelo sistema.
- Cada arquivo possui apenas uma senha, o acesso é liberado ou não na sua totalidade.
- Não é possível determinar quais tipos de operações podem ou não ser concedidas e outra desvantagem é a dificuldade de compartilhamento já que todos os demais usuários deveriam ter conhecimento da senha.

60/68

---

---

---

---

---

---

---

---

## Grupos de Usuários

- Tem como princípio a associação de cada usuário do sistema a um grupo. Os usuários são organizados com o objetivo de compartilhar arquivos entre si.
- Implementa três tipos de proteção: owner (dono), group (grupo) e all (todos) e na criação do arquivo é especificado quem e o tipo de acesso aos três níveis de proteção.
- Em geral, somente o dono ou usuários privilegiados é que podem modificar a proteção dos arquivos.

61/68

---

---

---

---

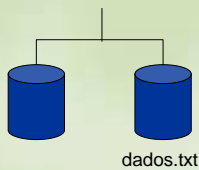
---

---

---

---

## Proteção por Grupo de Usuários



Nível de proteção	Tipo de Acesso
Owner	Leitura Escrita Execução Eliminação
Group	Leitura
All	--

62/68

---

---

---

---

---

---

---

---

## Lista de Controle de Acesso

- Access Control List – ACL consiste em uma lista associada a cada arquivo onde são especificados quais os usuários e os tipos de acesso permitidos.
- O tamanho desta estrutura pode ser bastante extenso se um arquivo tiver seu acesso compartilhado por diversos usuários.
- Existe um overhead adicional devido a pesquisa seqüencial que o sistema deverá realizar na lista sempre que solicitado.
- É possível encontrar tanto a proteção por grupos de usuários quanto pela lista de acesso oferecendo uma maior flexibilidade ao mecanismo de proteção.

63/68

---

---

---

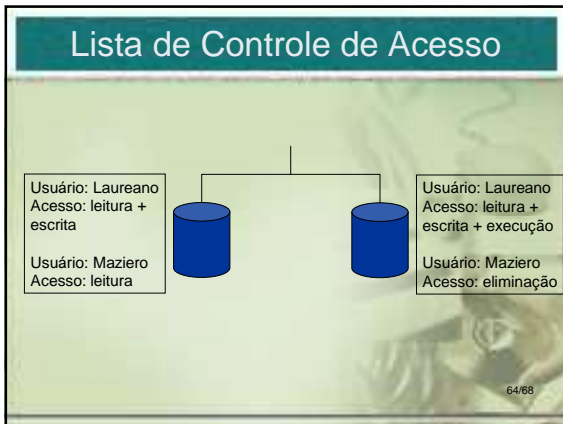
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

### Implementação de Caches

- O acesso a disco é bastante lento ao comparado a memória principal e este é o fator para que as operações de E/S serem um problema ao desempenho do sistema.
- Com o objetivo de minimizar este problema, a maioria dos sistemas operacionais implementa a técnica de buffer cache onde o sistema reserva uma área na memória para que se tornem disponíveis caches utilizados em operações de acesso a disco.
- Quando uma operação é realizada o sistema procura no cache a informação e caso não encontre, ele busca no disco e depois atualiza a buffer cache.

65/68

---

---

---

---

---

---

---

---

---

---

### Implementação de Caches

- Como existe limite para o tamanho do cache o sistema adota políticas de substituição como o FIFO (First in First out) ou a LRU (Least Recently Used).
- No caso de dados permanecerem por um longo tempo na memória a ocorrência com problemas de energia pode resultar na perda de tarefas já executadas e consideradas salvas em disco.
- Existem duas maneiras de tratar deste problema: o sistema pode possuir uma rotina que executa, em intervalos de tempo, atualizações em disco de todos os blocos modificados no cache.

66/68

---

---

---

---

---

---

---

---

---

---

## Implementação de Caches

- Uma segunda alternativa é que toda vez que um bloco do cache for modificado, realizar uma atualização no disco (write-through caches).
- Podemos concluir que a primeira técnica implica em menor quantidade de operações de E/S porém o risco de perda de dados é maior, pois pode ocorrer que dados atualizados de um arquivo ainda no cache sejam perdidos na falta de energia. Isso já não acontece nos caches tipo write-through em função de seu funcionamento porém existe um aumento considerável nas operações de E/S o que o torna menos eficiente.
- A maioria dos sistemas utiliza a primeira técnica.

67/68

---

---

---

---

---

---

---

---

## Então....

- Dúvidas ?
- Perguntas ?
- Sugestões ?

68/68

---

---

---

---

---

---

---

---