

# Sistemas Operacionais

Sistemas de Arquivos

# Introdução

- O armazenamento e recuperação de informações é uma atividade essencial para qualquer tipo de aplicação. Um processo deve ser capaz de ler e gravar de forma permanente grande volume de dados em dispositivos como fitas e discos, além de poder compartilhá-los com outros processos. A maneira pela qual o SO estrutura e organiza estas informações é através da implementação de arquivos.

# Introdução

- Os arquivos são gerenciados pelo SO de maneira a facilitar o acesso dos usuários ao seu conteúdo. A parte do sistema responsável por essa gerência é denominada “sistema de arquivos”. O sistema de arquivos é a parte mais visível de um SO, pois a manipulação de arquivos é uma atividade frequentemente realizada pelos usuários, devendo sempre ocorrer de maneira uniforme, independente dos diferentes dispositivos de armazenamento.

# Arquivos

- Um arquivos é constituído por informações logicamente relacionadas. Estas informações podem representar instruções ou dados. Um arquivo executável, por exemplo, contém instruções compreendidas pelo processador, enquanto um arquivo de dados pode ser estruturado livremente como um arquivo texto ou de forma mais rígida como em um banco de dados relacional.

# Arquivos

- Na realidade, um arquivo é um conjunto de registros definidos pelo sistema de arquivos, tornando seu conceito abstrato e generalista. A partir dessa definição, o conteúdo do arquivo pode ser manipulado segundo conceitos preestabelecidos.

# Arquivos

- Os arquivos são armazenados pelo SO em diferentes dispositivos físicos, como fitas magnéticas, discos magnéticos e discos ópticos. O tipo de dispositivo no qual o arquivo é armazenado deve ser isolado pelo SO, de forma que exista uma independência entre os arquivos a serem manipulados e o meio de armazenamento.

# Arquivos

- Um arquivo é identificado por um nome, composto por uma sequência de caracteres. Em alguns sistemas de arquivos é feita distinção entre caracteres alfabéticos maiúsculos e minúsculos. Regras como extensão máxima do nome e quais são os caracteres válidos também pode variar.

# Arquivos

- Em alguns SO, a identificação de um arquivo é composta por duas partes separadas com um ponto. A parte após o ponto é denominada extensão do arquivo e tem como finalidade identificar o conteúdo do arquivo.



# Arquivos: Organização de Arquivos

- A “organização de arquivos” consiste em como os seus dados estão internamente armazenados. A estrutura dos dados pode variar em função do tipo de informação contida no arquivo. Arquivos texto possuem propósitos completamente distintos de arquivos executáveis, conseqüentemente, estruturas diferentes podem adequar-se melhor a um tipo do que a outro.

# Arquivos: Organização de Arquivos

- No momento da criação de um arquivo, seu criador pode definir qual a organização adotada. Esta organização pode ser uma estrutura suportada pelo SO ou definida pela própria aplicação.

# Arquivos: Organização de Arquivos

- A forma mais simples de organização é através de uma sequência não-estruturada de bytes. Neste tipo de organização, o sistema de arquivos não impõe nenhuma estrutura lógica para os dados. A aplicação deve definir toda a organização, estando livre para estabelecer seus próprios critérios. A grande vantagem deste modelo é a flexibilidade para criar diferentes estruturas de dados, porém todo o controle de acesso ao arquivo é de inteira responsabilidade da aplicação.

# Arquivos: Organização de Arquivos

- Alguns SO possuem diferentes organizações de arquivos. Neste caso, cada arquivo criado deve seguir um modelo suportado pelo sistema de arquivos. As organizações mais conhecidas e implementadas são a sequencial, relativa e indexada. Nestes tipos de organização, podemos visualizar um arquivo como um conjunto de registros. Os registros podem ser classificados em registros de tamanho fixo, quando possuírem sempre o mesmo tamanho, ou registros de tamanho variável.

# Arquivos: Métodos de Acesso

- Em função de como o arquivo está organizado, o sistema de arquivos pode recuperar registros de diferentes maneiras. Inicialmente, os primeiros SO só armazenavam arquivos em fitas magnéticas. Com isso, o acesso era restrito a leitura dos registros na ordem em que eram gravados, e a gravação de novos registros só era possível no final do arquivo.

# Arquivos: Métodos de Acesso

- Este tipo de acesso, chamado de “acesso sequencial”, era próprio da fita magnética, que, como meio de armazenamento, possuía esta limitação.
- Com o advento dos discos magnéticos, foi possível a introdução de métodos de acesso mais eficientes.

# Arquivos: Métodos de Acesso

- O primeiro a surgir foi o “acesso direto”, que permite a leitura/gravação de um registro diretamente na sua posição. Este método é realizado através do número do registro, que é a sua posição relativa ao início do arquivo. No acesso direto não existe restrição à ordem em que os registros são lidos ou gravados, sendo sempre necessária a especificação o número do registro.

# Arquivos: Métodos de Acesso

- O acesso direto pode ser combinado com o acesso sequencial. Com isso é possível acessar diretamente um registro qualquer de um arquivo e, a partir deste, acessar sequencialmente os demais.



# Arquivos: Métodos de Acesso

- Um método de acesso mais sofisticado que tem como base o acesso direto, é o chamado “acesso indexado” ou “acesso por chave”.
- Para este acesso, o arquivo deve possuir uma área de índice onde existam ponteiros para os diversos registros.

# Arquivos: Métodos de Acesso

- Sempre que a aplicação desejar acessar um registro, deverá ser especificada uma chave através da qual o sistema pesquisará na área de índice o ponteiro correspondente. A partir desta informação é realizado um acesso direto ao registro desejado.

# Arquivos: Operações de Entrada/Saída

- O sistema de arquivos disponibiliza um conjunto de rotinas que permite às aplicações realizarem operações de E/S, como tradução de nomes em endereços, leitura e gravação de dados e criação/eliminação de arquivos. Na realidade, as rotinas de E/S têm como função disponibilizar uma interface simples e uniforme entre a aplicação e os diversos dispositivos.

# Arquivos: Operações de Entrada/Saída

- Algumas rotinas de E/S:
  - CREATE – Criação de arquivos
  - OPEN – Abertura de arquivos
  - READ – Leitura de um arquivos
  - WRITE – Gravação de um arquivo
  - CLOSE – Fechamento de um arquivo
  - DELETE – Eliminação de um arquivo

# Arquivos: Atributos

- Cada arquivo possui informações de controle denominadas “atributos”.
- Os atributos variam dependendo do sistema de arquivos, porém alguns, como tamanho de arquivo, proteção, identificação do criador e data de criação estão presentes em quase todos os sistemas.

# Diretórios

- A estrutura de “diretórios” é como o sistema organiza logicamente os diversos arquivos contidos em um disco. O diretório é uma estrutura de dados que contém entradas associadas aos arquivos onde cada entrada armazena informações como localização física, nome, organização e demais atributos.

# Diretórios

- Quando um arquivo é aberto, o SO procura a sua entrada na estrutura de diretórios, armazenando as informações sobre atributos e localização 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.

# Diretórios

- É importante que ao término do uso de arquivos, este seja fechado, ou seja, que se libere espaço na tabela de arquivos abertos.



# Diretórios

- A implementação mais simples de uma estrutura de diretórios é chamado de nível único (single-level directory). Neste caso, somente existe um único diretório contendo todos os arquivos do disco. Este modelo é bastante limitado, já que não permite que usuários criem arquivos com o mesmo nome, o que ocasionaria um conflito no acesso aos arquivos.

# Diretórios

- Como o sistema de nível único é bastante limitado, uma evolução do modelo foi a implementação de uma estrutura onde para cada usuário existiria um diretório particular denominado User File Directory (UFD). Com esta implementação, cada usuário passa a poder criar arquivos com qualquer nome, sem a preocupação de conhecer os demais arquivos do disco.

# Diretórios

- Para que o sistema possa localizar arquivos nessa estrutura, deve haver um nível de diretório adicional para controlar os diretórios individuais dos usuários. Este nível, denominado Master File Directory (MFD), é indexado pelo nome do usuário, onde cada entrada aponta para o diretório pessoal.

# Diretórios

- A estrutura de diretórios com dois níveis é análoga a uma estrutura de dados em árvore.
- Sob o ponto de vista do usuário, a organização dos seus arquivos em um único diretório não permite uma organização adequada.

# Diretórios

- A extensão do modelo de dois níveis para um de múltiplos níveis permitiu que os arquivos fossem logicamente melhor organizados. Este novo modelo, chamado estrutura de diretórios, em árvore (tree-structured directory), é adotado pela maioria dos sistemas.

# Diretórios

- Na estrutura em árvore, cada usuário pode criar diversos níveis de diretórios, também chamados subdiretórios. Cada diretório pode conter arquivos ou outros diretórios. O número de níveis de uma estrutura em árvore é dependente do sistema de arquivos de cada SO.

# Gerência de Espaço Livre em Disco

- A criação de arquivos em disco exige que o SO tenha o controle de quais áreas ou blocos no disco estão livres. Este controle é realizado utilizando-se alguma estrutura de dados que armazena informações que possibilitam ao sistema de arquivos gerenciar o espaço livre do disco.

# Gerência de Espaço Livre em Disco

- Nesta estrutura, geralmente uma lista ou tabela, é possível identificar blocos livres que poderão ser alocados a um novo arquivo. Neste caso, o espaço é removido da estrutura para que não seja reutilizado. No momento em que um arquivo é eliminado, todos os seus blocos são liberados para a lista de espaços livres.



# Gerência de Espaço Livre em Disco

- A forma mais simples de implementar uma estrutura de espaços livres é através de uma tabela denominada mapa de bits (bit map). Cada entrada na tabela é associada a um bloco do disco representado por um bit, podendo assumir valor igual a 0 (indicando bloco livre) ou 1 (indicando bloco alocado).

# Gerência de Espaço Livre em Disco

- Uma segunda maneira de realizar este controle é com uma estrutura de lista encadeada de todos os blocos livres do disco. Para que isso seja possível, cada bloco possui uma área reservada para armazenamento do endereço do próximo bloco. A partir do primeiro bloco livre e, então, possível o acesso sequencial aos demais de forma encadeada.

# Gerência de Espaço Livre em Disco

- Este esquema apresenta algumas restrições se considerarmos que, além do espaço utilizado no bloco com informação e controle, o algoritmo de busca de espaço livre sempre deve realizar uma pesquisa sequencial na lista.

# Gerência de Espaço Livre em Disco

- Uma outra solução leva em consideração que blocos contíguos são geralmente alocados ou liberados simultaneamente. Podemos, desta forma, enxergar o disco como um conjunto de segmentos de blocos livres. 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. Esta técnica de gerência de espaço livre é conhecida como tabela de blocos livres.

# Gerencia de Alocação de Espaço em Disco

- Alocação Contígua
- Alocação Encadeada
- Alocação Indexada

# Gerencia de Alocação de Espaço em Disco: Alocação Contígua

- Consiste em armazenar um arquivo em blocos sequencialmente dispostos no disco. Neste tipo de alocação, o sistema localiza um arquivo através do endereço do primeiro bloco e da sua extensão em blocos.

# Gerencia de Alocação de Espaço em Disco: Alocação Contígua

- O acesso a arquivos dispostos contiguamente no disco é bastante simples tanto para a forma sequencial quanto para a direta. Seu principal problema é a alocação de espaço livre para novos arquivos. Caso um arquivo deve ser criado com determinado tamanho, é necessário existir uma quantidade suficiente de blocos contíguos no disco para realizar a alocação.

# Gerencia de Alocação de Espaço em Disco: Alocação Contígua

- Para este tipo de alocação, podemos enxergar o disco como um grande vetor, onde os elementos podem ser considerados segmentos com tamanhos diferentes de blocos contíguos. Estes segmentos estão dispostos alternadamente entre segmentos ocupados e segmentos livres.



# Gerencia de Alocação de Espaço em Disco: Alocação Contígua

- No momento em que o SO deseja alocar espaço para armazenar um novo arquivo, pode existir mais de um segmento livre disponível com o tamanho exigido. Neste caso, é necessário que alguma estratégia de alocação seja adotada para selecionar qual o segmento na lista de blocos livres deva ser escolhido.

# Gerencia de Alocação de Espaço em Disco: Alocação Contígua

- Principais estratégias:
  - First-Fit: o primeiro segmento livre com tamanho suficiente para alocar o arquivo é selecionado. A busca na lista é sequencial, sendo interrompida tão logo se localiza e um segmento com tamanho adequado.

# Gerencia de Alocação de Espaço em Disco: Alocação Contígua

- Best-Fit: seleciona o menor segmento livre disponível com tamanho suficiente para armazenar o arquivo. A busca em toda lista se faz necessária para a seleção do segmento, a não ser que a lista esteja ordenada por tamanho.

# Gerencia de Alocação de Espaço em Disco: Alocação Contígua

- Worst-Fit: o maior segmento é alocado. Mais uma vez a busca em toda lista se faz necessária, a menos que exista uma ordenação por tamanho.
- Independente da estratégia utilizada, a alocação contígua apresenta um problema chamado fragmentação dos espaços livres.

# Gerencia de Alocação de Espaço em Disco: Alocação Contígua

- Como os arquivos são criados e eliminados frequentemente, os segmentos livres vão se fragmentando em pequenos pedaços por todo o disco.
- O problema pode tornar-se crítico quando um disco possui blocos livres disponíveis, porém não existe um segmento contíguo em que o arquivo possa ser alocado.

# Gerencia de Alocação de Espaço em Disco: Alocação Contígua

- O problema da fragmentação pode ser contornado através de rotinas que reorganizem todos os arquivos no disco de maneira que só exista um único segmento de blocos livres. Este procedimento, denominado desfragmentação, geralmente utiliza uma área de trabalho no próprio disco. Existe um grande consumo de tempo neste tipo de operação. É importante também ressaltar que a desfragmentação é um procedimento com efeito temporário e deve, portanto, ser realizada periodicamente.

# Gerencia de Alocação de Espaço em Disco: Alocação Encadeada

- Um arquivo pode ser organizado como um conjunto de blocos ligados logicamente no disco, independentemente da sua localização física. Cada bloco deve possuir um ponteiro para o bloco seguinte do arquivo e assim sucessivamente.

# Gerencia de Alocação de Espaço em Disco: Alocação Encadeada

- A fragmentação dos espaços livres, apresentado no método anterior, não ocasiona nenhum problema na alocação encadeada, pois os blocos livres alocados para um arquivo não precisam necessariamente estar contíguos.
- O que ocorre neste método é a fragmentação de arquivos, que é a quebra do arquivo em diversos pedaços denominados extents.



# Gerencia de Alocação de Espaço em Disco: Alocação Encadeada

- A fragmentação resulta no aumento do tempo de acesso aos arquivos, pois o mecanismo de leitura/gravação do disco deve se deslocar diversas vezes sobre sua superfície para acessar cada extent (excessivo tempo de seek). Para otimizar o tempo das operações de E/S neste tipo de sistema, é importante que o disco seja periodicamente desfragmentado. Apesar de ter propósitos diferentes, o procedimento de desfragmentação é idêntico ao da alocação contígua.

# Gerencia de Alocação de Espaço em Disco: Alocação Encadeada

- A alocação encadeada só permite que se realize acesso sequencial aos blocos dos arquivos. Isto constitui uma das principais desvantagens desta técnica, já que não é possível o acesso direto aos blocos. Além disso, essa técnica desperdiça espaço nos blocos como armazenamento de ponteiros.

# Gerencia de Alocação de Espaço em Disco: Alocação Indexada

- Soluciona uma das principais limitações da alocação encadeada, que é a impossibilidade do acesso direto aos blocos dos arquivos. O princípio desta técnica é manter os ponteiros de todos os blocos do arquivo em uma única estrutura denominada bloco de índice.
- A alocação indexada, além de permitir acesso direto aos blocos do arquivo, não utiliza informações de controle nos blocos de dados, como existentes na alocação encadeada.

# Proteção de Acesso

- Considerando que os meios de armazenamento são compartilhados entre diversos usuários, é de fundamental importância que mecanismos de proteção sejam implementados para garantir a proteção individual de arquivos e diretórios. Qualquer sistema de arquivos deve possuir mecanismos próprios para proteger o acesso às informações gravadas em discos e fitas, além de possibilitar o compartilhamento de arquivos entre usuários, quando desejado.

# Proteção de Acesso

- O tipo de acesso a arquivos é implementado mediante a concessão ou não dos diferentes acessos que podem ser realizados, como leitura (read), gravação (write), execução (execute) e eliminação (delete).

# Proteção de Acesso: Senha de Acesso

- A associação de uma senha de acesso a um arquivo é um princípio bastante simples. O controle de acesso se resume ao usuário ter o conhecimento da senha e, conseqüentemente, ter a liberação do acesso ao arquivo concedida pelo sistema.
- Como cada arquivo possui apenas uma senha, o acesso é liberado ou não na sua totalidade. Isto significa que não é possível determinar quais tipos de operação podem ou não ser concedidas.

# Proteção de Acesso: Senha de Acesso

- Outra desvantagem deste método é a dificuldade de compartilhamento de arquivos, pois, além do dono do arquivo, todos os demais usuários teriam que conhecer a senha de acesso.

# Proteção de Acesso: Grupos de Usuário

- A proteção baseada em grupos de usuários é implementada por diversos SO. Este tipo de proteção tem como princípio a associação de cada usuário do sistema a um grupo. Os grupos de usuários são organizados logicamente com o objetivo de compartilhar arquivos e diretórios.



# Proteção de Acesso: Grupos de Usuário

- Esse mecanismo implementa 3 níveis de proteção ao arquivo: owner (dono), group (grupo) e all (todos).
- Na criação do arquivo, o usuário especifica se o arquivo deve ser acessado somente pelo seu criador, pelos usuários do grupo ao qual ele pertence ou por todos os usuários do sistema. Nessa especificação é necessário associar o tipo de acesso aos três níveis de proteção.

# Proteção de Acesso: Lista de Controle de Acesso

- Consiste em uma lista associada a cada arquivo, onde são especificados quais os usuários e os tipos de acesso permitidos. Nesse caso, quando um usuário tenta acessar um arquivo, o SO verifica se a lista de controle autoriza a operação desejada.

# Proteção de Acesso: Lista de Controle de Acesso

- O tamanho dessa estrutura de dados pode ser bastante extenso se considerarmos que um arquivo pode ter seu acesso compartilhado por diversos usuários. Além deste fato, existe um overhead adicional, se comparado como mecanismo de proteção por grupo de usuários, devido à pesquisa sequencial que o sistema realizar na lista sempre que um acesso for solicitado.

# Proteção de Acesso: Lista de Controle de Acesso

- Em determinados sistemas de arquivos é possível encontrar tanto o mecanismo de proteção por grupos de usuários quanto o de lista de controle de acesso, oferecendo, desta forma, uma maior flexibilidade ao mecanismo de proteção de arquivos e diretórios.

# Implementação de Caches

- O acesso a disco é bastante lento se comparado o acesso à memória principal, devido à arquitetura dos discos magnéticos. Este é o principal motivo das operações de E/S com discos serem um problema para o desempenho do sistema.
- Como objetivo de minimizar este problema, a maioria dos sistemas de arquivos implementa uma técnica denominada “buffer cache”.

# Implementação de Caches

- Neste esquema, o SO reserva uma área da memória para que se tornem disponíveis caches utilizados em operações de acesso ao disco. Quando uma operação é realizada, seja leitura ou gravação, o sistema verifica se a informação desejada se encontra no buffer cache. Em caso positivo, não é necessário o acesso ao disco. Caso o bloco requisitado não se encontre no cache, a operação de E/S é realizada e o cache é atualizado.

# Implementação de Caches

- Como existe uma limitação no tamanho do cache, cada sistema adota políticas para substituição de blocos.
- Apesar de esta implementação melhorar o desempenho do sistema, aspectos de segurança devem ser levados em consideração.

# Implementação de Caches

- No caso de blocos de dados permanecerem por um longo período de tempo na memória principal, a ocorrência de problemas de energia pode ocasionar a perda de tarefas já realizadas e consideradas já salvas em disco.



# Implementação de Caches

- Existem duas maneiras distintas de tratar este problema. No primeiro caso, o SO possui uma rotina que executa periodicamente em um intervalo de tempo, atualizando em disco todos os blocos modificados no cache. Uma segunda alternativa é, toda vez que um bloco do cache for modificado, que seja realizada imediatamente uma atualização no disco (write-through caches).

# Implementação de Caches

- Analisando comparativamente as duas técnicas, podemos concluir que a primeira implica menor quantidade de operações de E/S, porém o risco de perda de dados é maior. Apesar de tal probabilidade ser pequena, pode ocorrer que dados atualizados de um arquivo e ainda no cache sejam perdidos, no caso de faltar energia. Isto já não aconteceria nos caches do tipo write-through, em função do seu próprio funcionamento, porém o aumento considerável de E/S tornam este método menos eficiente. Atualmente, a maioria dos sistemas utiliza a primeira técnica de otimização.