EAD

EDUCAÇÃO A DISTÂNCIA

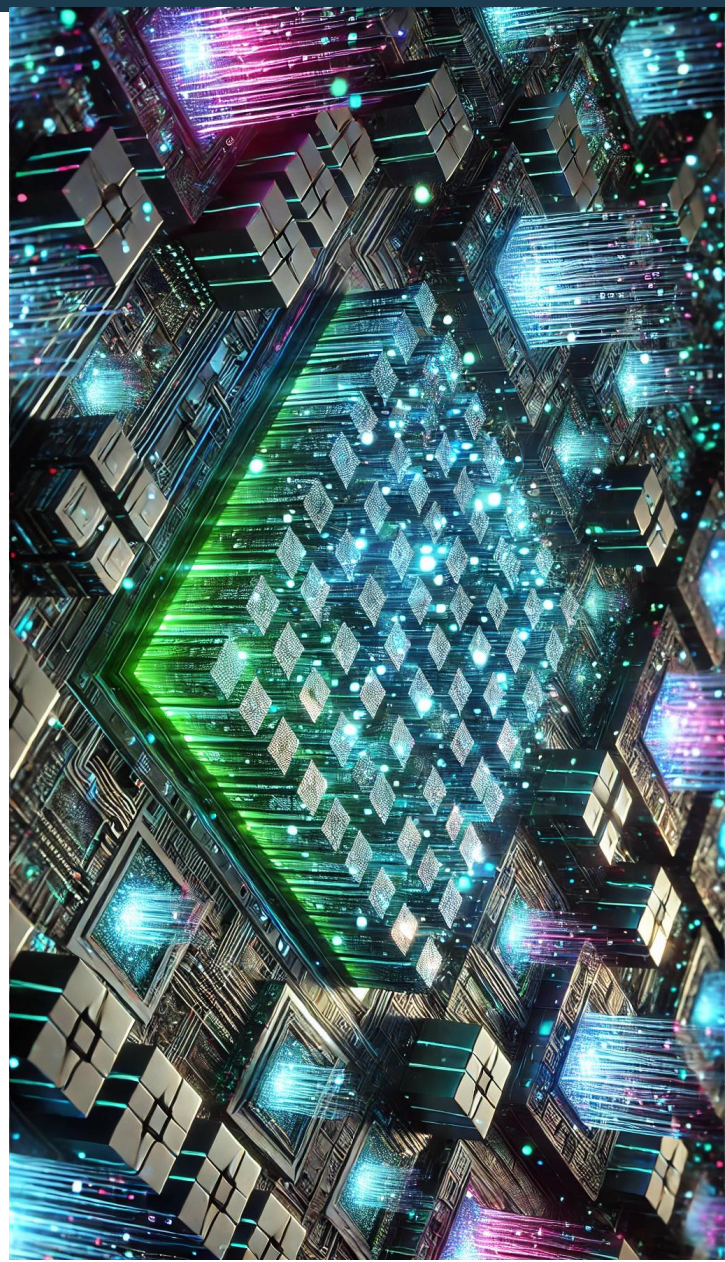
UNIDADE IV

Infraestrutura Computacional

Profa. Sandra Bozolan

Gerenciamento de memória

- A memória é um dispositivo essencial para a operação de um sistema de computação moderno. A memória consiste em um grande array de bytes, cada um com seu próprio endereço.
  A CPU extrai instruções da memória, de acordo com o valor do contador do programa. Essas instruções podem causar carga adicional a partir de endereços específicos da memória e armazenamento em endereços específicos da memória.



Fonte: Flux Fast 1.1, 2025 – A imagem representa, de forma abstrata, o visual da memória como dispositivo essencial na computação moderna.

Hardware básico

- A **memória principal** e os **registradores** embutidos dentro do próprio processador são o único espaço de armazenamento de uso geral que a CPU pode acessar diretamente.

- **Há instruções de máquina** que usam endereços da memória como argumentos, mas nenhuma que use endereços de disco.

- Os **registradores** que estão embutidos na **CPU** geralmente podem ser acessados dentro de um ciclo do relógio da CPU.

Memória virtual

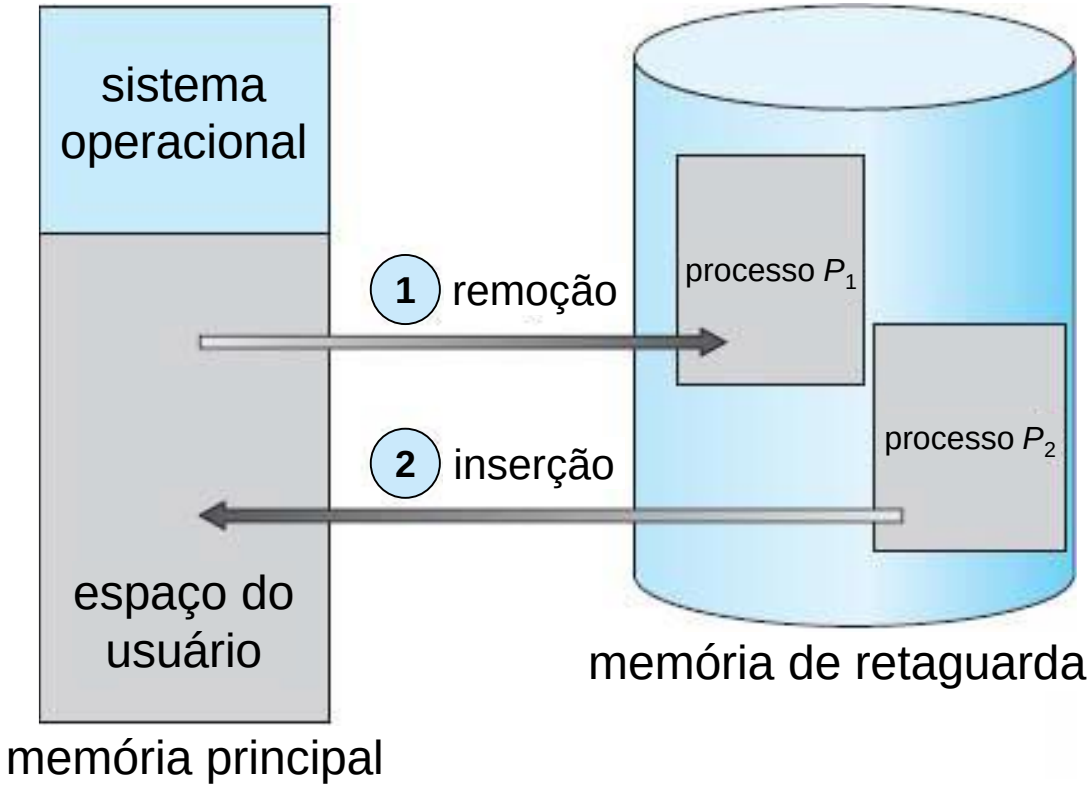
- Memória virtual é uma técnica sofisticada e poderosa de gerência de memória, em que as memórias principal e secundária são combinadas dando ao usuário a ilusão de existir uma memória muito maior que a capacidade real da memória principal. O conceito de memória virtual fundamenta-se em não vincular o endereçamento feito pelo programa dos endereços físicos da memória principal. Desta forma, programas e suas estruturas de dados deixam de estar limitados ao tamanho da memória física disponível, pois podem possuir endereços associados à memória secundária.

Memória virtual

- A primeira implementação de memória virtual foi realizada no início da década de 1960, no sistema Atlas, desenvolvido na Universidade de Manchester (Kilburn, 1962). Posteriormente, a IBM introduziria este conceito comercialmente na família System/370, em 1972. Atualmente, a maioria dos sistemas implementa memória virtual, com exceção de alguns sistemas operacionais de supercomputadores.
- O conceito de memória virtual se aproxima muito da ideia de um vetor, existente nas linguagens de alto nível. Quando um programa faz referência a um elemento do vetor, não há preocupação em saber a posição de memória daquele dado. O compilador se encarrega de gerar instruções que implementam esse mecanismo, tornando-o totalmente transparente ao programador.

Swapping

- Um processo deve estar na memória para ser executado. No entanto, ele pode ser transferido temporariamente da memória principal para uma memória de retaguarda e, então, trazido de volta à memória principal para continuar a execução.
- A permuta torna possível que o espaço de endereçamento físico de todos os processos exceda a memória física real do sistema, aumentando, assim, o grau de multiprogramação no sistema.



Fonte: Silberschartz (2015, p. 294).

Paginação por demanda

A paginação por demanda é uma técnica de gerenciamento de memória que permite que um processo carregue apenas as páginas necessárias em memória física, enquanto outras páginas permanecem armazenadas em disco. Isso reduz a quantidade de memória física necessária para executar um processo e aumenta a capacidade multitarefa do sistema.
Quando um processo tenta acessar uma página que não está na memória física, ocorre uma falta de página (page fault).
O sistema operacional então carrega a página necessária do disco para a memória física, o que pode resultar em uma latência significativa.

Organização da memória

A memória de um computador é organizada em uma hierarquia, com diferentes níveis de velocidade e custo. O nível mais rápido e caro é o cache, que é usado para armazenar dados e instruções frequentemente acessados. O próximo nível é a memória principal (RAM), que é mais lenta e menos cara do que o cache. O nível mais lento e menos caro é o armazenamento secundário (disco rígido), que é usado para armazenar dados que não estão sendo ativamente usados. A paginação por demanda usa a memória principal e o armazenamento secundário para gerenciar os dados de um processo.

Padrões de acesso à memória

Os padrões de acesso à memória descrevem como um processo acessa os dados em memória. Esses padrões podem variar dependendo do tipo de aplicação, do algoritmo usado e de outros fatores. Existem dois padrões principais de acesso à memória:

- localidade espacial;
- localidade temporal.

Padrões de acesso à memória

- A localidade espacial ocorre quando um processo acessa dados em locais adjacentes na memória.
- Já a localidade temporal ocorre quando um processo acessa os mesmos dados repetidamente em um curto período de tempo. Esses padrões são importantes para o desempenho da paginação por demanda, pois podem ajudar a minimizar o número de faltas de página.

Interatividade

Quais são os dois tipos de localidade padrão de acesso à memória?

a) Localidade global e localidade temporal.
b) Localidade local e localidade geoespacial.
c) Localidade em cache e localidade de RAM.
d) Localidade de ROM e localidade de RAM.
e) Localidade espacial e localidade temporal.

Resposta

Quais são os dois tipos de localidade padrão de acesso à memória?

a) Localidade global e localidade temporal.
b) Localidade local e localidade geoespacial.
c) Localidade em cache e localidade de RAM.
d) Localidade de ROM e localidade de RAM.
e) Localidade espacial e localidade temporal.

Impacto dos padrões de acesso na paginação por demanda

- Se um processo exibe alta localidade espacial ou temporal, a paginação por demanda pode funcionar muito bem, pois o sistema operacional pode manter as páginas necessárias na memória física e evitar a necessidade de carregar páginas do disco. No entanto, se um processo exibe baixa localidade, a paginação por demanda pode resultar em um número significativo de faltas de página, levando a uma latência significativa e um desempenho lento. Isso ocorre porque, com baixa localidade, é mais provável que um processo acesse dados em diferentes partes da memória, exigindo que o sistema operacional carregue muitas páginas diferentes do disco.

Impacto dos padrões de acesso na paginação por demanda

Dessa forma, existem várias estratégias que podem ser usadas para otimizar os padrões de acesso à memória, incluindo:

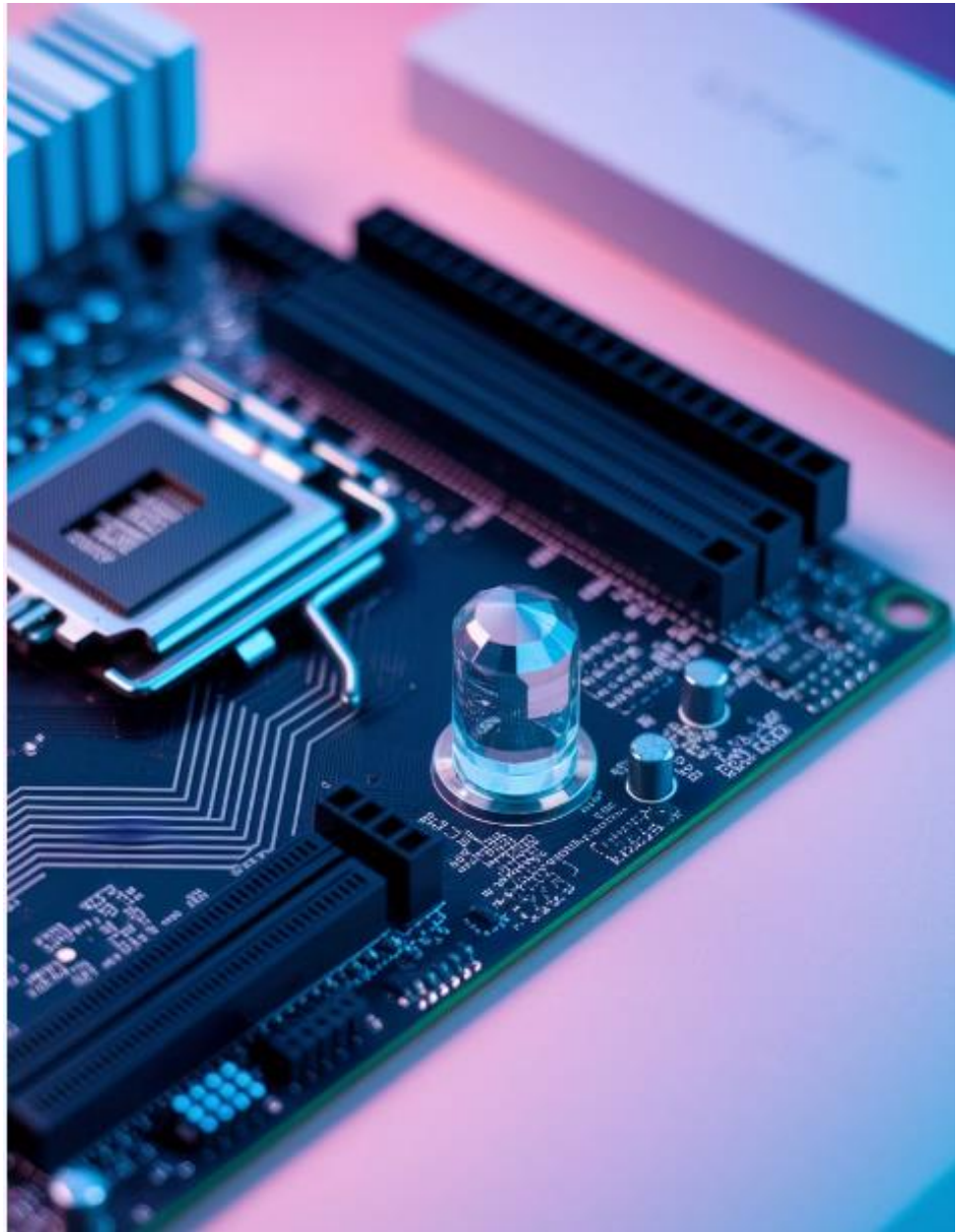
- **Otimização do código:** os programadores podem otimizar seus programas para melhorar os padrões de acesso à memória. Por exemplo, podem usar técnicas como a alocação de memória contínua para reduzir a fragmentação e melhorar a localidade espacial.
- **Algoritmos de alocação de memória:** o sistema operacional pode usar diferentes algoritmos de alocação de memória para otimizar o uso da memória e reduzir o número de faltas de página. Algoritmos como FIFO e LRU (Least Recently Used) podem ajudar a gerenciar a memória de forma mais eficiente, evitando que as páginas mais usadas sejam substituídas prematuramente.

Impacto dos padrões de acesso na paginação por demanda

- **Gerenciamento de cache:** o sistema operacional pode usar caches para armazenar as páginas mais usadas na memória, evitando que sejam carregadas do disco a cada acesso. Isso pode reduzir significativamente o número de faltas de página.
- Imagine um sistema de gerenciamento de banco de dados que processa consultas complexas que requerem acesso a grandes quantidades de dados. Se o sistema for projetado de forma que as consultas acessem os dados de forma aleatória, sem exibir localidade espacial ou temporal, a paginação por demanda pode resultar em um grande número de faltas de página. Isso pode levar a um desempenho lento, impactando a resposta às consultas e a capacidade do sistema de lidar com o volume de transações.

Permuta-padrão

- Envolve a transferência de processos entre a memória principal e uma memória de retaguarda. A memória de retaguarda é comumente um disco veloz. Ela deve ser suficientemente grande para acomodar cópias de todas as imagens da memória para todos os usuários e deve fornecer acesso direto a essas imagens da memória. O sistema mantém uma fila de prontos composta por todos os processos cujas imagens da memória estão na memória de retaguarda ou na memória principal e que estão prontos para serem executados.



Fonte: Flux Fast 1.1, 2025 – A imagem representa, de forma abstrata, o clock do computador.

Permuta-padrão

- Caso não esteja, e se não houver uma região de memória livre, o despachante remove um processo correntemente em memória e o permuta com o processo desejado. Em seguida, ele recarrega os registradores e transfere o controle ao processo selecionado.

O tempo de mudança de contexto nesse sistema de permuta é bem alto. Para termos uma ideia desse tempo, suponha que o processo do usuário tenha um tamanho de 100 MB e a memória de retaguarda seja um disco rígido padrão com taxa de transferência de 50 MB por segundo. A transferência real do processo de 100 MB em uma das transferências (para dentro ou para fora) da memória principal leva:

100 MB / 50 MB por segundo = 2 segundos

Permuta-padrão

- Suponha que a operação de I/O esteja enfileirada porque o dispositivo está ocupado. Se removermos o processo P1 da memória e inserirmos o processo P2, a operação de I/O poderia tentar usar a memória que agora pertence ao processo P2. As duas principais soluções para esse problema são: nunca permutar um processo com I/O pendente, ou executar operações de I/O somente em buffers do sistema operacional. Assim, as transferências entre buffers do sistema operacional e a memória do processo ocorrerão apenas quando o processo for inserido na memória. Esse armazenamento duplo em buffer adiciona overhead por si só.

Permuta-padrão

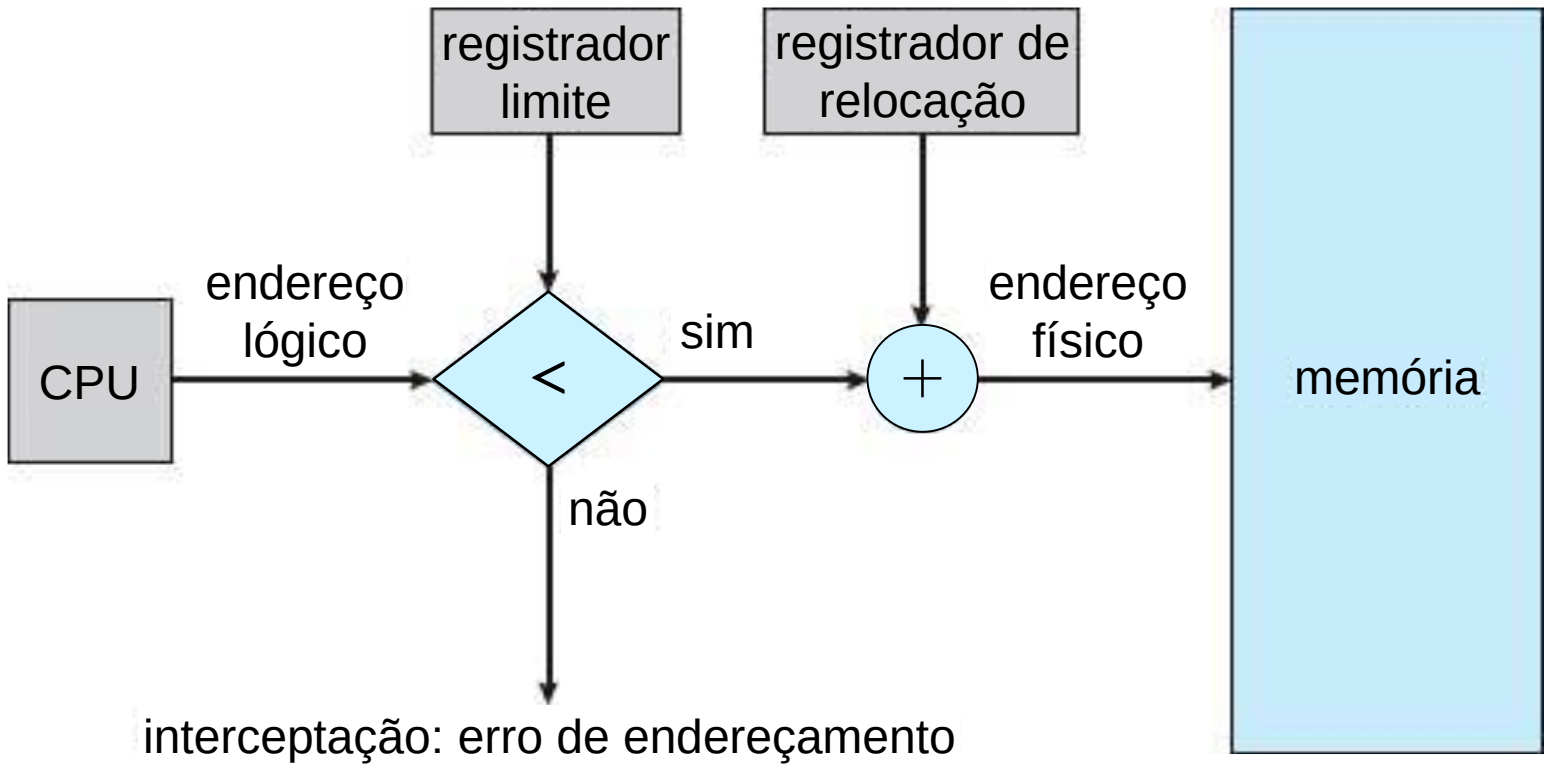
- A permuta-padrão não é usada nos sistemas operacionais modernos. Ela requer muito tempo de permuta e fornece muito pouco tempo de execução para ser uma solução razoável para o gerenciamento da memória. Versões modificadas de permuta, no entanto, são encontradas em muitos sistemas, inclusive no UNIX, no Linux e no Windows. Em uma variação comum, a permuta normalmente é desabilitada, sendo iniciada se o montante de memória livre (memória não utilizada disponível para o sistema operacional ou os processos usarem) cai abaixo de um valor limite.

Permuta em sistemas móveis

- Embora a maioria dos sistemas operacionais para PCs e servidores dê suporte a alguma versão modificada de permuta, os sistemas móveis normalmente não suportam de forma alguma a permuta. Os dispositivos móveis costumam usar memória flash, em vez dos discos rígidos mais volumosos, como seu espaço de armazenamento persistente. A restrição de espaço resultante é uma razão para os projetistas de sistemas operacionais móveis evitarem a permuta.
- O Android não dá suporte à permuta e adota uma estratégia semelhante à usada pelo iOS. Ele pode encerrar um processo se não houver memória livre suficiente disponível. No entanto, antes de encerrar um processo, o Android grava seu estado da aplicação na memória flash para que ela possa ser rapidamente reiniciada.

Proteção da memória

- O registrador de relocação contém o valor do menor endereço físico; o registrador limite contém o intervalo de endereços lógicos (por exemplo, relocação = 100040 e limite = 74600). Cada endereço lógico deve pertencer ao intervalo especificado pelo registrador limite. A MMU (memory management unit) mapeia o endereço lógico dinamicamente adicionando o valor ao registrador de relocação; assim, esse endereço mapeado é enviado à memória.



Fonte: Silberschartz (2015, p. 296).

Alocação de memória

- Um dos métodos mais simples para alocação da memória é dividir a memória em várias partições de tamanho fixo. Cada partição pode conter exatamente um processo, devido ao fato do grau de multiprogramação ser limitado pelo número de partições. Nesse método de partições múltiplas, quando uma partição está livre, um processo é selecionado da fila de entrada e carregado na partição disponível. Quando o processo termina, a partição torna-se disponível para outro processo. Esse método, denominado de MFT, foi originalmente usado pelo sistema operacional IBM OS/360, mas não está mais em uso.

- O método chamado MVT é uma generalização do esquema de partições fixas, sendo usado principalmente em ambientes batch.

Alocação de memória

- Em geral, os blocos de memória disponíveis compõem um conjunto de brechas de vários tamanhos espalhadas pela memória. Quando um processo chega e precisa de memória, o sistema procura no conjunto por uma brecha que seja suficientemente grande para esse processo. Se a brecha for grande demais, ela será dividida em duas partes. Uma parte é alocada ao processo que chegou; a outra é devolvida ao conjunto de brechas. Quando um processo é encerrado, ele libera seu bloco de memória que é, então, colocado novamente no conjunto de brechas.

    - Se a nova brecha for adjacente a outras brechas, essas brechas adjacentes serão mescladas para formar uma brecha maior. Nesse momento, o sistema pode precisar verificar se existem processos esperando por memória e se essa memória recém-liberada e recombinada poderia atender às demandas de algum desses processos em espera.

Alocação de memória

- Há muitas soluções para esse problema. As estratégias do primeiro-apto (first-fit), do mais-apto (best-fit) e do menos-apto (worst-fit) são as mais usadas para selecionar uma brecha livre no conjunto de brechas disponíveis.

- Primeiro-apto: aloca a primeira brecha que seja suficientemente grande. A busca pode começar tanto no início do conjunto de brechas quanto na locação na qual a busca anterior pelo primeiro-apto terminou. Podemos encerrar a busca assim que encontrarmos uma brecha livre suficientemente grande.

Alocação de memória

- Mais-apto: aloca a menor brecha que seja suficientemente grande. Devemos pesquisar a lista inteira, a menos que ela seja ordenada por tamanho. Essa estratégia produz a brecha com menos espaço sobrando.

- Menos-apto: aloca a maior brecha. Novamente, devemos pesquisar a lista inteira, a menos que ela seja classificada por tamanho. Essa estratégia produz a brecha com mais espaço sobrando, que pode ser mais útil do que a brecha com menos espaço sobrando da abordagem do mais-apto.

  - Simulações têm mostrado que tanto o primeiro-apto quanto o mais-apto são melhores do que o menos-apto em termos de redução de tempo e utilização de memória. Nem o primeiro-apto, nem o mais-apto é claramente melhor do que o outro em termos de utilização de memória, mas o primeiro-apto geralmente é mais rápido.

Interatividade

Suponha que o processo do usuário tenha um tamanho de 100 MB e a memória de retaguarda seja um disco rígido padrão com taxa de transferência de 50 MB por segundo. A transferência real do processo de 100 MB em uma das transferências (para dentro ou para fora) da memória principal leva 100 MB / 50 MB por segundo = 2 segundos. Assinale qual é a alternativa correta de milissegundos.

a) 200 milissegundos.
b) 2.000 milissegundos.
c) 20.000 milissegundos.
d) 22.000 milissegundos.
e) 200.000 milissegundos.

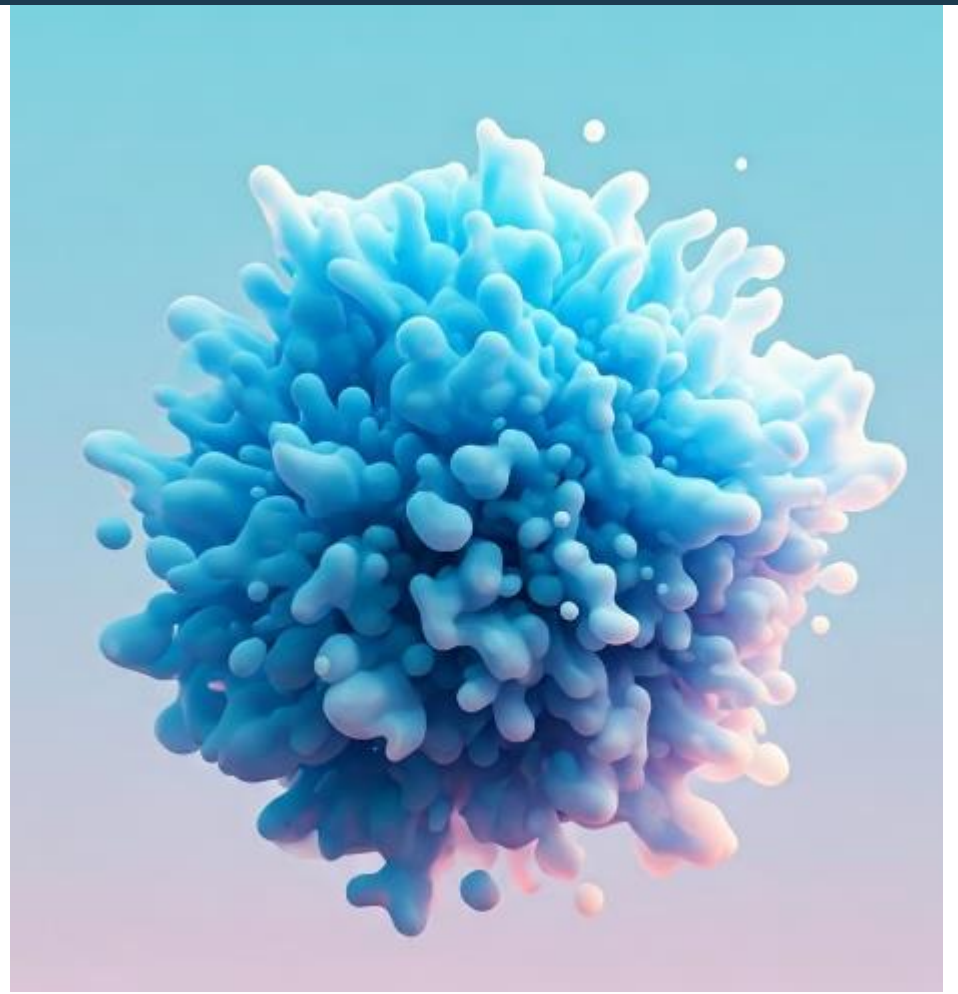
Resposta

Suponha que o processo do usuário tenha um tamanho de 100 MB e a memória de retaguarda seja um disco rígido padrão com taxa de transferência de 50 MB por segundo. A transferência real do processo de 100 MB em uma das transferências (para dentro ou para fora) da memória principal leva 100 MB / 50 MB por segundo = 2 segundos. Assinale qual é a alternativa correta de milissegundos.

a) 200 milissegundos.
b) 2.000 milissegundos.
c) 20.000 milissegundos.
d) 22.000 milissegundos.
e) 200.000 milissegundos.

Sistema de arquivos

- O armazenamento e a recuperação de informações são atividades essenciais para qualquer tipo de aplicação. Um processo deve ser capaz de ler e gravar de forma permanente um grande volume de dados em dispositivos como fitas e discos, além de poder compartilhá-los com outros processos. A maneira pela qual o sistema operacional estrutura e organiza estas informações é por intermédio da implementação de arquivos.



Fonte: Flux Fast 1.1, 2025 – A imagem representa, de forma abstrata, um arquivo.

Arquivo

- Um arquivo é constituído por informações logicamente relacionadas, onde tais 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. Um arquivo pode ser representado como 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 seguindo conceitos preestabelecidos.

Tipos de arquivos

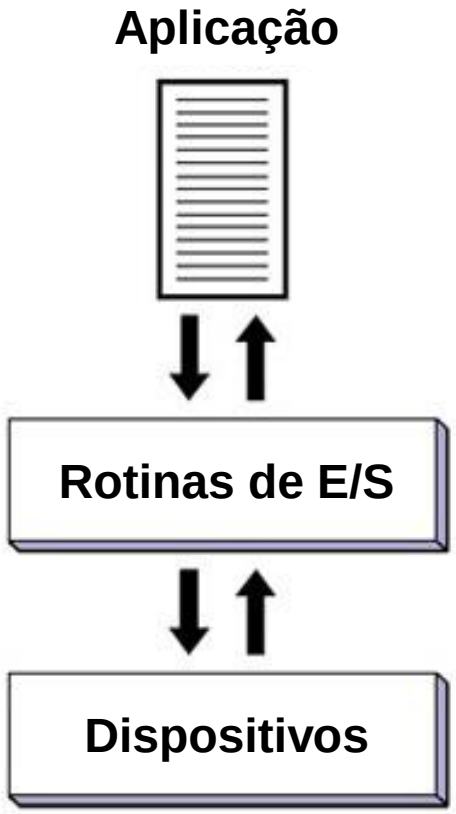
- **Arquivos Executáveis:** Contêm instruções diretamente interpretáveis pelo processador. São fundamentais para execução de programas. Exemplos: .exe, .bin, .app.

- **Arquivos de Dados:** Armazenam informações para uso por aplicações. Podem seguir formatação livre ou estruturada. Exemplos: .txt, .csv, .xml, .json.

- **Arquivos de Sistema:** Fundamentais para funcionamento do SO. Incluem configurações e recursos essenciais. Exemplos: .sys, .dll, .so.

Operações de entrada/saída

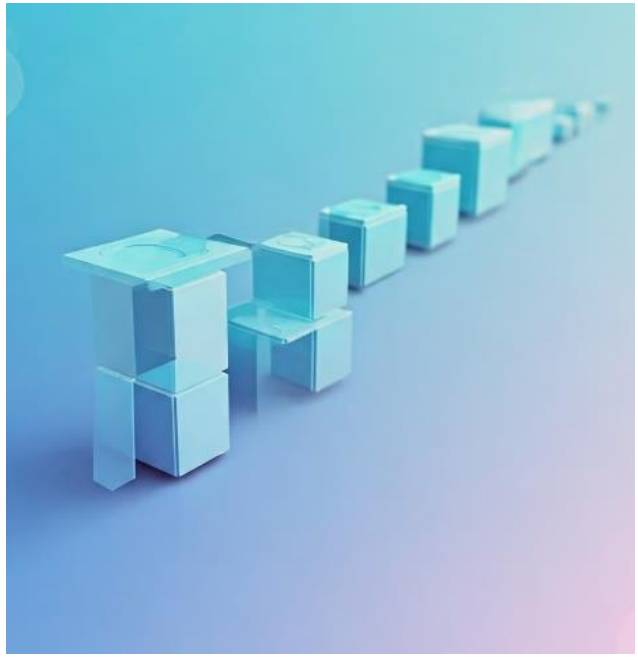
- O sistema de arquivos disponibiliza um conjunto de rotinas que permite às aplicações realizarem operações de E/S, como tradução de nomes em endereços, leitura e gravação de dados e criação/eliminação de arquivos. Na realidade, as rotinas de E/S têm como função disponibilizar uma interface simples e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



Fonte: Machado (2013, p. 264) – A imagem representa, de forma figurativa, as operações de entrada e saída.

Atributos

- Cada arquivo possui informações de controle denominadas atributos. Os atributos variam dependendo do sistema de arquivos, porém alguns, como tamanho do arquivo, proteção, identificação do criador e data de criação, estão presentes em quase todos os sistemas.
- Alguns atributos especificados na criação do arquivo não podem ser modificados em função de sua própria natureza, como organização e data/hora de criação. Outros são alterados pelo próprio sistema operacional, como tamanho e data/hora do último backup realizado. Existem, ainda, atributos que podem ser modificados pelo próprio usuário, como proteção do arquivo, tamanho máximo e senha de acesso.



Fonte: Flux Fast 1.1, 2025 – A imagem representa, de forma figurativa, um atributo de arquivo.

Diretórios

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

Metadados de Arquivos

Nome	Identificador utilizado pelo usuário para referenciar o arquivo
Extensão	Indica o tipo e formato do conteúdo
Tamanho	Espaço ocupado em bytes ou blocos
Proprietário	Usuário criador ou responsável pelo arquivo
Timestamps	Datas de criação, modificação e acesso
Permissões	Direitos de leitura, escrita e execução

Fonte: Machado (2013, p. 265) – Extensão de arquivos.

Sistemas de arquivos remotos

- Com o advento das redes, a comunicação entre computadores remotos tornou-se possível. A comunicação em rede permite o compartilhamento de recursos espalhados por um campus ou até mesmo ao redor do globo.

- O primeiro método implementado envolve a transferência manual de arquivos entre máquinas por meio de programas como o FTP (File Transfer Protocol). O segundo grande método usa um sistema de arquivos distribuído (DFS – distributed file system, em inglês) em que diretórios remotos são visíveis a partir de um computador local.

    - Em alguns aspectos, o terceiro método, a World Wide Web, é uma volta ao primeiro. É necessário um navegador para a obtenção de acesso aos arquivos remotos e são usadas operações separadas (essencialmente um encapsulador do FTP) para transferir arquivos. Cada vez mais, a computação em nuvem vem sendo usada para o compartilhamento de arquivos.

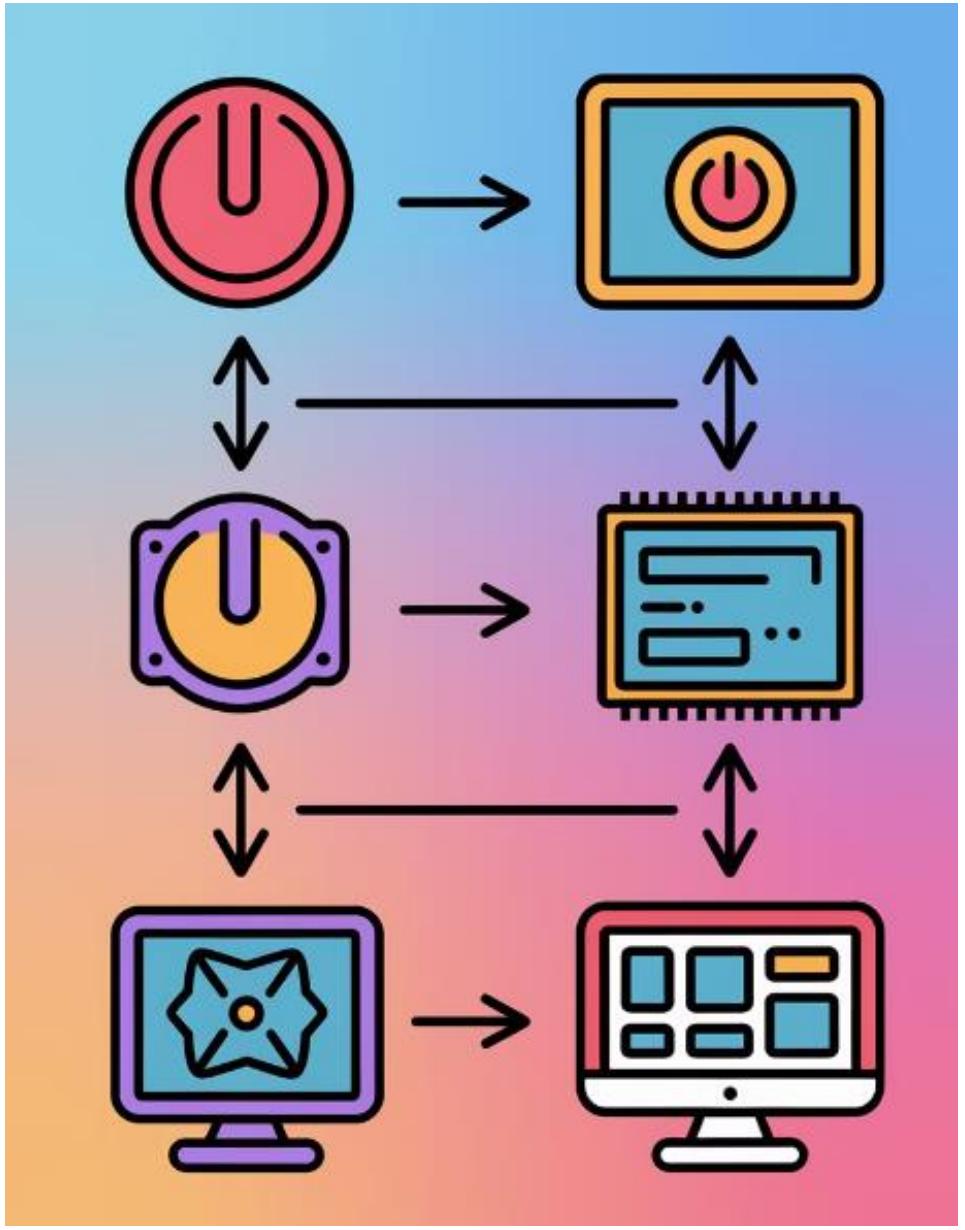
Implementação

Fonte: Flux Fast 1.1, 2025 – A imagem representa, de forma figurativa, a utilização de arquivos.

Várias estruturas em disco e em memória são usadas para implementar um sistema de arquivos. Essas estruturas variam, dependendo do sistema operacional e do sistema de arquivos, mas alguns princípios gerais são aplicáveis. Em disco, o sistema de arquivos pode conter informações sobre como inicializar um sistema operacional, já que nele há estruturas como:

Sistema de arquivos – Implementação

As informações em memória são usadas tanto no gerenciamento do sistema de arquivos quanto na melhoria do desempenho por meio do armazenamento em cache. Os dados são carregados em tempo de montagem, atualizados durante operações sobre o sistema de arquivos e descartados na desmontagem. Vários tipos de estruturas podem ser incluídos, como:



Fonte: Flux Fast 1.1, 2025 – A imagem representa, de forma figurativa, os sistemas de arquivos.

Implementação

1. **Bloco de controle de inicialização (por volume):** pode conter as informações requeridas pelo sistema para inicializar um sistema operacional a partir desse volume.

2. **Bloco de controle de volume (por volume):** contém detalhes do volume (ou partição), tais como o número de blocos na partição, o tamanho dos blocos, uma contagem de blocos livres e ponteiros para blocos livres e uma contagem de FCBs livres e ponteiros para FCBs.

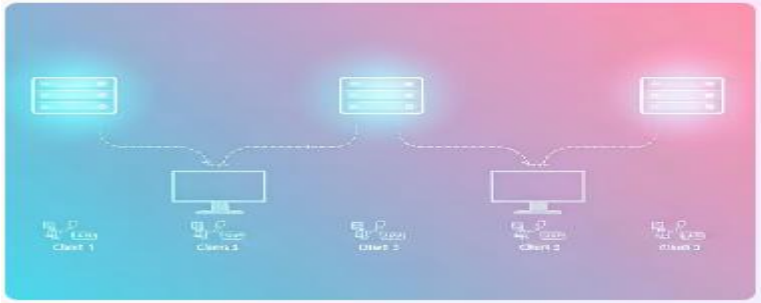
3. **Estrutura de diretório (por sistema de arquivos):** é usada para organizar arquivos. No UFS, ela inclui os nomes de arquivo e os números de inode associados.

4. **Arquivos individuais:** um FCB por arquivo contém muitos detalhes sobre o arquivo. O FCB possui um número identificador exclusivo para permitir a associação a uma entrada do diretório.

Implementação

1. Uma **tabela de montagens** em memória contém informações sobre cada volume montado.
2. Um **cache em memória** da estrutura de diretórios mantém as informações referentes aos diretórios acessados recentemente.
3. A **tabela de arquivos abertos em todo o sistema** contém uma cópia do FCB de cada arquivo aberto, assim como outras informações.
4. A **tabela de arquivos abertos por processo** contém um ponteiro para a entrada apropriada na tabela de arquivos abertos em todo o sistema, assim como outras informações.
5. **Buffers** mantêm blocos do sistema de arquivos quando eles estão sendo lidos do disco ou gravados em disco.

Sistemas de arquivos distribuídos



- **Arquitetura Distribuída:** Dados armazenados em múltiplos servidores conectados em rede. Alta disponibilidade e tolerância à falha.
- **Replicação e Consistência:** Cópias dos dados mantidas em diversos locais. Exige protocolos para garantir consistência entre réplicas.
- **Desafios de Segurança:** Proteção de dados: transmitidos pela rede. Requer autenticação robusta e criptografia ponta a ponta.

Fonte: Flux Fast 1.1, 2025 – As imagens representam, de forma figurativa, os sistemas de arquivos.

Interatividade

As informações em memória são usadas tanto no gerenciamento do sistema de arquivos quanto na melhoria do desempenho por meio do armazenamento em cache. Os dados são carregados em tempo de montagem, atualizados durante operações sobre o sistema de arquivos e descartados na desmontagem. Quais das alternativas abaixo **não** é um tipo de estrutura para gerenciamento de arquivos?

a) Tabela de montagens em memória.
b) Cache em memória da estrutura de diretórios.
c) Tabela de arquivos abertos em todo o sistema que contém uma cópia do FCB.
d) Buffers que mantêm blocos do sistema de arquivos.
e) Tabela de arquivos abertos por processo.

Resposta

As informações em memória são usadas tanto no gerenciamento do sistema de arquivos quanto na melhoria do desempenho por meio do armazenamento em cache. Os dados são carregados em tempo de montagem, atualizados durante operações sobre o sistema de arquivos e descartados na desmontagem. Quais das alternativas abaixo **não** é um tipo de estrutura para gerenciamento de arquivos?

a) Tabela de montagens em memória.
b) Cache em memória da estrutura de diretórios.
c) Tabela de arquivos abertos em todo o sistema que contém uma cópia do FCB.
d) Buffers que mantêm blocos do sistema de arquivos.
e) Tabela de arquivos abertos por processo.

Sistema de arquivos – Segurança

- Dizemos que um sistema é seguro quando seus recursos são usados e acessados como esperado sob todas as circunstâncias. Infelizmente, a segurança total não pode ser atingida, mas, mesmo assim, devemos possuir mecanismos que tornem as brechas de segurança uma ocorrência rara, e não a norma.

- Violações (ou a má utilização) da segurança do sistema podem ser categorizadas como intencionais (maliciosas) ou acidentais. É mais fácil se proteger contra a má utilização acidental do que contra a maliciosa.

Sistema de arquivos – Segurança

A lista a seguir inclui vários tipos de violações acidentais e maliciosas de segurança:

- **Brecha de sigilo:** esse tipo de violação envolve a leitura não autorizada de dados (ou roubo de informações). Normalmente, uma brecha de sigilo é o objetivo de um invasor que captura dados secretos de um sistema ou fluxo de dados, tais como informações de cartões de crédito ou informações de credenciais para roubo de identidades, o que pode resultar diretamente em dinheiro para o intruso.

- **Brecha de integridade:** essa violação envolve a modificação não autorizada de dados. Esses ataques podem, por exemplo, resultar na transferência de responsabilidade para terceiros inocentes ou na modificação do código-fonte de uma aplicação comercial importante.

Sistema de arquivos – Segurança

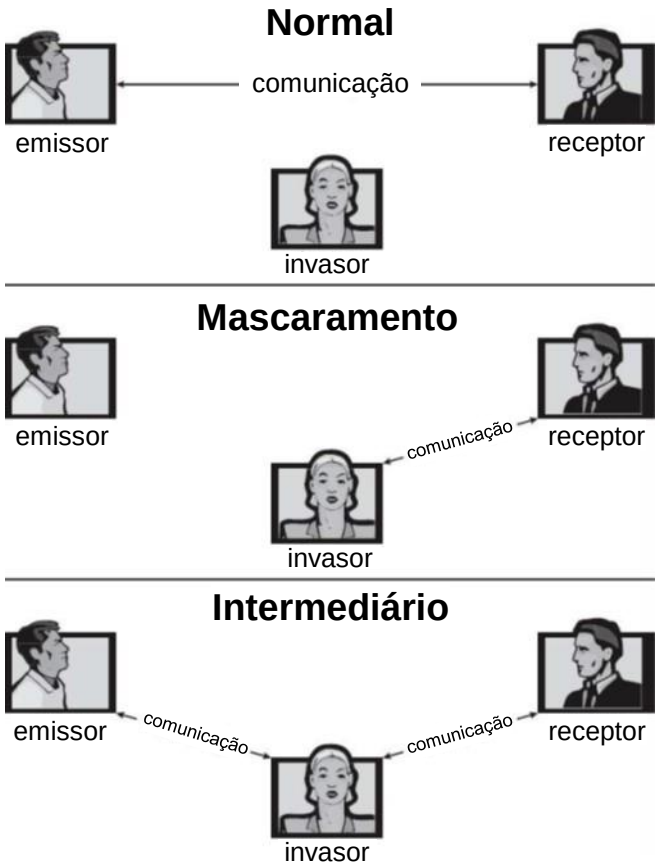
- **Brecha de disponibilidade:** essa violação envolve a destruição não autorizada de dados. Alguns crackers preferem provocar destruição e ganhar status ou se vangloriar de direitos, em vez de obter ganhos financeiros. A desfiguração de websites é um exemplo comum desse tipo de brecha de segurança.

- **Roubo de serviço:** essa violação envolve o uso não autorizado de recursos. Por exemplo, um invasor (ou programa invasor) pode instalar um daemon em um sistema que atue como servidor de arquivos.

- **Recusa de serviço:** essa violação envolve o impedimento do uso legítimo do sistema. Ataques de recusa de serviço (DOS – denial-of-service, em inglês) são, algumas vezes, acidentais. O verme original da Internet transformou-se em um ataque DOS quando um bug não conseguiu retardar sua rápida disseminação.

Sistema de arquivos – Segurança

- Os agressores usam vários métodos-padrão em suas tentativas de violar a segurança. O mais comum é o mascaramento, em que um participante de uma comunicação finge ser alguém que não é (outro hospedeiro ou outra pessoa). Por meio do mascaramento, os agressores violam a autenticação ou a precisão da identidade; eles podem obter acesso que, normalmente, não receberiam ou aumentar seus privilégios que, normalmente, não lhes seriam atribuídos.



Fonte: Silberschartz (2015, p. 530).
Ataques-padrão à segurança.

Sistema de arquivos – Segurança

Para proteger um sistema, devemos tomar medidas de segurança em quatro níveis:

- **Físico:** o sítio ou os sítios que contêm os sistemas de computação devem ser fisicamente protegidos contra a entrada forçada ou furtiva de intrusos, assim como, tanto as salas das máquinas quanto os terminais ou estações de trabalho que têm acesso às máquinas devem ser protegidos.
- **Humano:** a autorização deve ser feita cuidadosamente para assegurar que apenas usuários apropriados tenham acesso ao sistema. Até mesmo usuários autorizados, no entanto, podem ser “encorajados” a deixar outras pessoas utilizarem seu acesso (mediante suborno, por exemplo), podendo, também, ser levados a permitir o acesso pela engenharia social.

Sistema de arquivos – Segurança

- **Sistema operacional:** o sistema deve proteger a si próprio contra brechas de segurança acidentais ou propositais. Um processo fora de controle poderia constituir um ataque acidental de recusa de serviço; uma consulta a um serviço poderia revelar senhas; um estouro de pilha poderia permitir o acionamento de um processo não autorizado etc. A lista de brechas possíveis é quase infinita.

- **Rede:** muitos dados nos sistemas modernos viajam por linhas privadas dedicadas, linhas compartilhadas como a Internet, conexões sem fio ou linhas dial-up.

    - A interceptação desses dados poderia ser tão danosa quanto uma invasão em um computador, e a interrupção de comunicações poderia constituir um ataque remoto de recusa de serviço, diminuindo o uso do sistema e a confiança dos usuários.

Sistema de arquivos – Segurança

- A segurança nos dois primeiros níveis deve ser mantida para que a segurança do sistema operacional seja assegurada. Uma vulnerabilidade em um nível alto de segurança (físico ou humano) permite que medidas de segurança estritamente de baixo nível (sistema operacional) sejam burladas. Portanto, o antigo provérbio de que uma corrente é tão forte quanto seu elo mais fraco é particularmente verdadeiro quando se trata da segurança de sistemas.

Ameaça de programas

- Os processos, junto com o kernel, são o único meio de execução de tarefas em um computador. Logo, escrever um programa que crie uma brecha de segurança, ou faça um processo normal mudar seu comportamento e criar uma brecha, é um objetivo comum dos crackers. Na verdade, até mesmo a maioria dos eventos de segurança não relacionados com programas têm como objetivo causar uma ameaça de programa. Por exemplo, embora seja útil fazer login em um sistema sem autorização, é muito mais útil deixar para trás um daemon de porta dos fundos que forneça informações ou permita o acesso fácil, mesmo se o ataque original for bloqueado.

Compreendendo processos

Execução

Realizam tarefas do sistema operacional e aplicativos.

Programação

Seguem instruções definidas por desenvolvedores.

Permissões

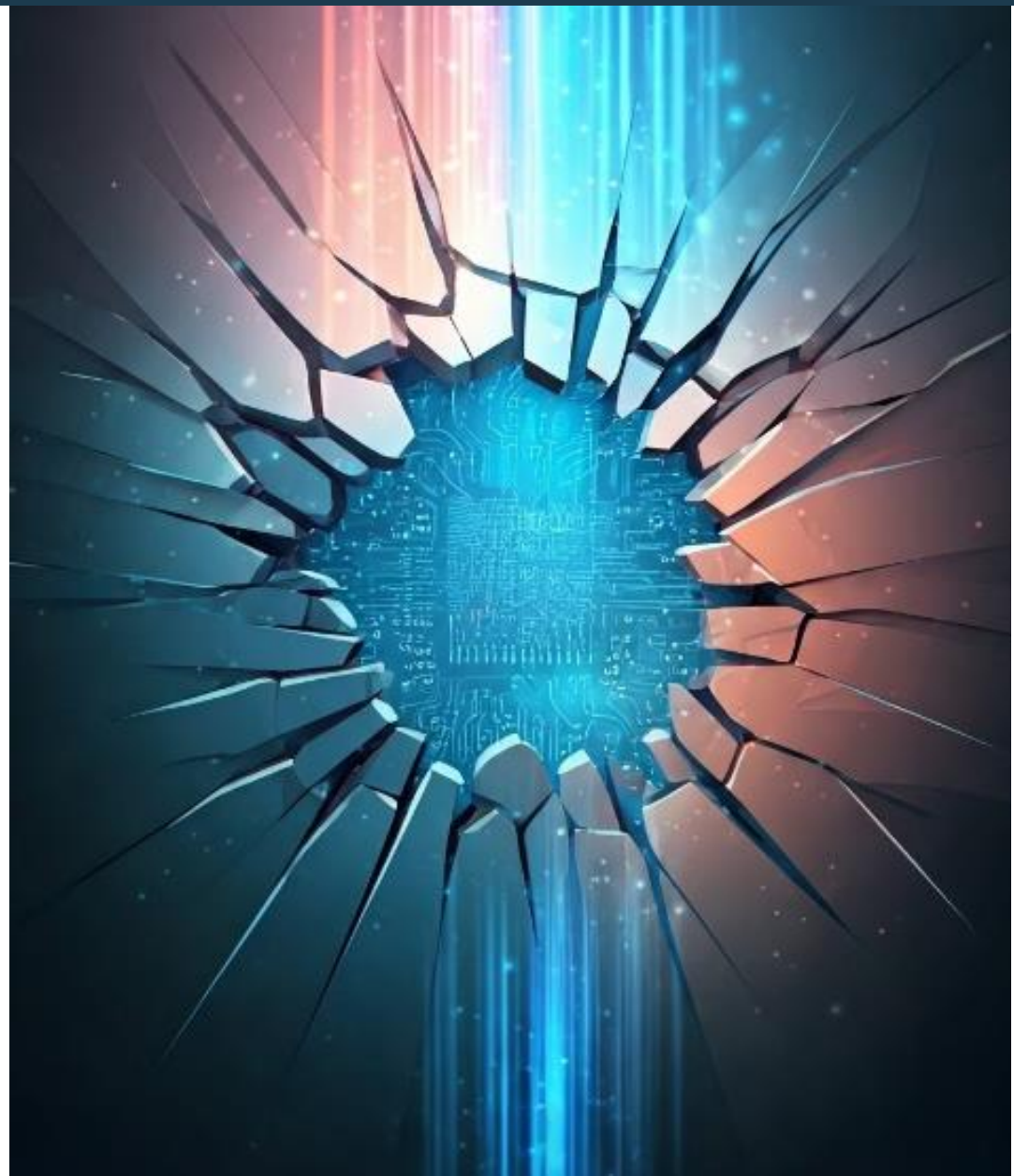
Operam com privilégios específicos de sistema

Isolamento

Executam em espaços de memória protegidos.

Estratégias de ataque

- **Acesso Inicial:** Exploração de vulnerabilidades para primeira entrada no sistema.
- **Estabelecimento de Processo Malicioso:** Criação de daemon ou alteração de processo existente.
- **Ocultação de Rastros:** Remoção de evidências do acesso não autorizado.
- **Manutenção de Acesso:** Uso da porta dos fundos para futuras invasões.



Fonte: Flux Fast 1.1, 2025 – A imagem representa, de forma figurativa, estratégias de ataque.

Medidas de proteção

- A proteção efetiva exige abordagem em múltiplas camadas. Nenhuma medida isolada garante segurança total contra ameaças avançadas.

Interatividade

Para proteger um sistema, devemos tomar medidas de segurança em quatro níveis. Qual das alternativas abaixo **não** é considerada uma medida de segurança?

a) Sistema operacional.
b) Humano.
c) Físico.
d) Buffer.
e) Rede.

Resposta

Para proteger um sistema, devemos tomar medidas de segurança em quatro níveis. Qual das alternativas abaixo **não** é considerada uma medida de segurança?

a) Sistema operacional.
b) Humano.
c) Físico.
d) Buffer.
e) Rede.

ATÉ A PRÓXIMA!