**Sistemas Computacionais e Segurança  
  
  
  
Nome:** Kauan Camargo Miranda **RA:** 825141414 **Nome:** Marcello Andrade **RA:** 824216008 **Nome:** Ryan Rocha **RA:** 825155072 **1. Redes de Computadores**

Redes de computadores são fundamentais para a comunicação e o compartilhamento de recursos em ambientes corporativos, governamentais e pessoais. Elas permitem que dispositivos se conectem entre si, transmitam dados e compartilhem serviços, como impressoras e arquivos. Um exemplo simples de rede de computadores é a rede doméstica, onde dispositivos como computadores, smartphones, smart TVs e outros equipamentos estão conectados a um roteador, o qual por sua vez está ligado à internet.

Imagine que você está em casa e decide imprimir um documento. Em vez de conectar diretamente o seu computador à impressora, você pode enviar o comando de impressão pela rede sem fio para a impressora, que está conectada ao mesmo roteador. Este é um exemplo prático de como as redes de computadores facilitam a comunicação entre dispositivos e tornam mais eficiente o uso de recursos compartilhados.

**2. Verificação de Autenticação RBAC e MFA**

A autenticação é um componente crítico para garantir que apenas usuários autorizados tenham acesso a sistemas e dados sensíveis. Existem diversas abordagens para garantir a segurança de sistemas, como o **RBAC** (Controle de Acesso Baseado em Função) e o **MFA** (Autenticação Multifatorial).

* **RBAC** é um modelo de controle de acesso onde os usuários são atribuídos a funções específicas (por exemplo, administrador, gerente, usuário comum) e cada função tem permissões específicas. Em um ambiente corporativo, por exemplo, um gerente pode ter acesso a dados financeiros, enquanto um funcionário de nível inferior não tem essas permissões. Isso ajuda a limitar o acesso baseado na função de cada indivíduo, promovendo a segurança no gerenciamento de permissões.
* **MFA** é uma técnica que exige mais de um fator de autenticação para garantir que a identidade de um usuário seja verificada. Um exemplo de MFA seria quando um usuário tenta acessar um sistema e, além de inserir sua senha (fator algo que sabe), ele também precisa fornecer um código temporário enviado para seu celular (fator algo que possui). Isso aumenta a segurança, pois mesmo que alguém consiga descobrir a senha, ainda precisaria do segundo fator para acessar o sistema.

**3. Segurança em Sistemas Distribuídos**

Sistemas distribuídos envolvem a utilização de múltiplos computadores interconectados que trabalham juntos para alcançar um objetivo comum. Eles são utilizados em diversas áreas, como em serviços de nuvem (Amazon Web Services, Google Cloud), onde os dados são armazenados e processados por diversos servidores localizados em diferentes regiões do mundo.

A segurança em sistemas distribuídos envolve garantir que os dados e recursos distribuídos estejam protegidos contra acessos não autorizados, falhas e ataques. Um exemplo prático seria um serviço de armazenamento em nuvem, como o Dropbox. Imagine que você tem arquivos importantes armazenados nele. Para garantir a segurança, o serviço de nuvem implementa criptografia de ponta a ponta, ou seja, enquanto os dados são transferidos entre os servidores distribuídos, eles são criptografados, garantindo que, mesmo se um servidor for comprometido, os dados não serão acessíveis sem a chave de descriptografia.

Além disso, sistemas distribuídos devem ser resilientes, ou seja, devem continuar funcionando mesmo quando uma parte do sistema falha. Isso é alcançado através de técnicas como replicação de dados, onde uma cópia de dados críticos é mantida em múltiplos servidores, para que, se um servidor falhar, os dados ainda estejam disponíveis.

**4. Segurança em IoT (Internet das Coisas)**

A **Internet das Coisas (IoT)** se refere à rede de dispositivos físicos conectados à internet, como termostatos inteligentes, câmeras de segurança, lâmpadas, eletrodomésticos e até mesmo dispositivos médicos. Esses dispositivos coletam e trocam dados em tempo real para oferecer maior conveniência e eficiência. Porém, devido à sua natureza conectada, a segurança em IoT é uma preocupação crescente.

Um exemplo claro de risco de segurança em IoT seria um **dispositivo de câmera de segurança** conectado à internet. Se esse dispositivo não tiver medidas adequadas de segurança, como senhas fortes e criptografia de dados, um invasor poderia acessá-lo remotamente e visualizar as imagens capturadas pela câmera. Em ambientes industriais, a IoT pode ser usada para monitorar máquinas e processos, mas sem a devida segurança, um atacante poderia causar falhas no sistema e comprometer a operação. Para garantir a segurança, os dispositivos IoT devem ser projetados com criptografia de dados, atualizações de segurança frequentes e autenticação robusta para evitar o comprometimento.

**5. Arquitetura de Computadores**

A arquitetura de computadores refere-se à estrutura interna dos sistemas de computação, ou seja, a forma como os componentes de hardware e software interagem para executar tarefas. A arquitetura de computadores envolve várias camadas, como a unidade central de processamento (CPU), a memória, a unidade de entrada/saída (E/S) e os dispositivos periféricos.

Um exemplo clássico da arquitetura de computadores pode ser observado em um **processador moderno**, como o Intel i7. Esse processador possui múltiplos núcleos (cores) e threads, permitindo realizar várias operações simultaneamente, o que é fundamental para a execução de tarefas complexas. A **memória cache** é uma parte da arquitetura que armazena dados temporários para acesso rápido, acelerando o desempenho do sistema. Já a **unidade de controle** é responsável por coordenar as operações entre as várias partes do computador.

Em um ambiente mais avançado, como em **arquiteturas paralelas** ou **distribuídas**, múltiplos processadores podem ser usados para dividir o processamento de tarefas e acelerar os resultados. Esse tipo de arquitetura é comum em servidores de alto desempenho e supercomputadores.

Esses tópicos ilustram a complexidade e a interconexão dos componentes envolvidos em sistemas computacionais modernos e destacam a importância de abordagens robustas para garantir a segurança e o bom funcionamento desses sistemas. Cada uma dessas áreas exige atenção cuidadosa, desde o design de redes até a implementação de segurança em dispositivos e arquiteturas.

### Extra ****Sistemas Operacionais: Conceito e Funcionalidades****

Um **sistema operacional (SO)** é um conjunto de software responsável por gerenciar o hardware do computador e fornecer serviços para os programas de aplicação. Ele atua como uma camada intermediária entre o hardware do computador e o software, facilitando a interação entre ambos e oferecendo um ambiente controlado para a execução de tarefas. Os sistemas operacionais são fundamentais para o funcionamento de qualquer dispositivo computacional, desde desktops e servidores até dispositivos móveis e sistemas embarcados.

1. **Gerenciamento de Processos:** O sistema operacional gerencia todos os processos em execução no computador, garantindo que eles recebam os recursos necessários para a execução, como tempo de CPU, memória e dispositivos de entrada/saída. O **agendador de processos** é responsável por determinar a ordem de execução dos processos, garantindo que nenhum processo consuma todos os recursos e que o sistema seja responsivo. Além disso, o sistema operacional lida com a **criação**, **execução** e **finalização** de processos.

Exemplo: Em um sistema multitarefa, o SO alterna rapidamente entre vários processos (como navegar na internet, editar um documento e ouvir música), dando a impressão de que todos estão sendo executados simultaneamente.

1. **Gerenciamento de Memória:** A memória principal (RAM) precisa ser gerenciada de forma eficiente para que o sistema execute várias tarefas sem travamentos. O sistema operacional aloca e desaloja a memória para os processos, garantindo que cada um tenha a quantidade necessária de espaço sem interferir nos outros. O conceito de **memória virtual** permite que o sistema operacional simule mais memória do que fisicamente disponível, usando o disco rígido como "memória extra".

Exemplo: Quando você abre um programa como um editor de texto, o sistema operacional aloca uma parte da memória RAM para esse processo. Se você abrir mais programas, o SO vai gerenciando a memória disponível, movendo dados entre a RAM e o disco rígido, se necessário.

1. **Gerenciamento de Armazenamento:** O SO organiza e gerencia o acesso a dispositivos de armazenamento, como discos rígidos (HDDs), unidades de estado sólido (SSDs) e outros dispositivos de armazenamento. Ele implementa **sistemas de arquivos** para armazenar, organizar e recuperar dados. Sistemas como NTFS (Windows), ext4 (Linux) e APFS (macOS) são exemplos de sistemas de arquivos usados para gerenciar os dados em discos.

Exemplo: Quando você salva um arquivo em seu computador, o sistema operacional organiza esse arquivo no disco rígido, mantendo um índice de onde ele está localizado para acessos rápidos no futuro.

1. **Gerenciamento de Entrada/Saída (E/S):** O sistema operacional gerencia a comunicação entre o computador e os dispositivos de entrada/saída, como teclados, mouses, impressoras e telas. Ele lida com as operações de leitura e gravação de dados nesses dispositivos, garantindo que as interações com o usuário sejam rápidas e eficientes.

Exemplo: Ao clicar em um botão na interface de um aplicativo, o sistema operacional é responsável por interpretar esse comando e, em seguida, interagir com o hardware para realizar a ação, como abrir uma nova janela ou imprimir um documento.

1. **Controle de Acesso e Segurança:** O sistema operacional implementa mecanismos de segurança para garantir que apenas usuários autorizados acessem recursos do sistema. Ele controla **autenticação de usuários**, **permissões de arquivos** e **políticas de segurança** para proteger dados e impedir acesso não autorizado.

Exemplo: Ao tentar acessar um arquivo protegido, o sistema operacional verifica as permissões de acesso do usuário para determinar se ele tem ou não permissão para abrir o arquivo.

1. **Interface de Usuário:** A interface de usuário (UI) é uma das partes mais visíveis de um sistema operacional. Pode ser uma **interface gráfica de usuário (GUI)**, como o Windows ou macOS, ou uma **interface de linha de comando (CLI)**, como o terminal do Linux ou o prompt de comando do Windows. A interface permite que os usuários interajam com o sistema operacional e executem tarefas de forma intuitiva.

Exemplo: No Windows, a interface gráfica oferece uma área de trabalho com ícones de programas, onde o usuário pode clicar para abrir aplicativos. No Linux, usuários avançados podem interagir com o sistema operacional por meio do terminal, digitando comandos para realizar diversas tarefas.

1. **Sistemas Operacionais Monolíticos:** Em um SO monolítico, todo o gerenciamento de recursos (processos, memória, E/S, etc.) é feito por um único código central. O Linux é um exemplo de sistema operacional monolítico, onde o núcleo (kernel) gerencia todas as funções do sistema.
2. **Sistemas Operacionais Microkernel:** Em um SO microkernel, as funções do sistema operacional são divididas em módulos menores e mais independentes. O kernel central mantém apenas funções básicas, como comunicação entre processos, enquanto outros serviços, como controle de dispositivos e gerenciamento de memória, são executados em módulos separados. O **minix** é um exemplo de sistema com microkernel.
3. **Sistemas Operacionais de Tempo Real:** Esses sistemas operacionais são projetados para processar dados em tempo real e fornecer respostas dentro de um tempo determinado. São usados em sistemas embarcados, como aviões, carros e dispositivos médicos, onde um atraso pode ser catastrófico. Um exemplo é o **RTOS** (Real-Time Operating System).
4. **Sistemas Operacionais Distribuídos:** São sistemas em que múltiplos computadores atuam como um único sistema, compartilhando recursos e realizando tarefas em conjunto. O **Google Chrome OS** e o **Android** podem ser considerados exemplos de sistemas operacionais distribuídos, pois se baseiam em arquiteturas em nuvem para proporcionar serviços de computação distribuída.

* **Windows:** Um dos sistemas operacionais mais populares, amplamente utilizado em desktops e laptops. Suporta uma ampla gama de softwares e dispositivos.
* **Linux:** Um sistema operacional de código aberto, muito utilizado em servidores, computadores pessoais e sistemas embarcados. A flexibilidade e o controle sobre o sistema tornam o Linux ideal para desenvolvedores e administradores de sistemas.
* **macOS:** O sistema operacional da Apple, utilizado em seus computadores. Conhecido pela sua interface intuitiva e integração com o ecossistema da Apple.
* **Android e iOS:** São sistemas operacionais móveis usados em smartphones e tablets, com ênfase na interface de usuário e na compatibilidade com uma variedade de aplicativos móveis.

Em resumo, o sistema operacional é o "cérebro" que coordena e gerencia os recursos de um computador ou dispositivo, proporcionando um ambiente estável e eficiente para a execução de programas e a interação com os usuários. Ele é essencial para garantir que o hardware e o software trabalhem juntos de maneira fluida e segura, otimizando o desempenho e protegendo os dados dos usuários.

**Resumo**O campo de **sistemas computacionais e segurança** envolve diversas áreas interconectadas que garantem a funcionalidade e proteção dos sistemas de tecnologia. As **redes de computadores** são fundamentais para a comunicação e o compartilhamento de recursos, permitindo que dispositivos se conectem e transmitam dados, como exemplificado pela rede doméstica que conecta dispositivos a um roteador para acessar a internet e compartilhar recursos, como impressoras.

A **verificação de autenticação** é crucial para proteger sistemas. O **RBAC (Controle de Acesso Baseado em Função)** limita o acesso a dados e recursos conforme as funções atribuídas aos usuários, enquanto o **MFA (Autenticação Multifatorial)** fortalece a segurança, exigindo múltiplos fatores de verificação para acessar sistemas sensíveis, como senhas e códigos temporários enviados ao celular.

Nos **sistemas distribuídos**, a segurança é um fator crítico, já que múltiplos servidores podem estar interconectados e armazenar dados sensíveis. Exemplos práticos incluem serviços de nuvem, onde a criptografia de ponta a ponta e a replicação de dados garantem a proteção e a continuidade do serviço, mesmo diante de falhas.

A **segurança em IoT (Internet das Coisas)** se tornou um tema central devido ao aumento de dispositivos conectados à internet, como câmeras de segurança e eletrodomésticos. A falta de medidas de segurança adequadas nesses dispositivos pode expô-los a ataques, como no caso de uma câmera vulnerável que poderia ser acessada remotamente por um invasor.

Por fim, a **arquitetura de computadores** trata da estrutura interna dos sistemas, como a interação entre a CPU, memória e dispositivos de entrada/saída. Arquiteturas modernas, como processadores com múltiplos núcleos e cache, melhoram a eficiência e o desempenho dos sistemas, enquanto as arquiteturas distribuídas e paralelas são usadas em ambientes de alto desempenho.

Em resumo, garantir a segurança e a eficiência de sistemas computacionais envolve o design de redes robustas, controles de acesso eficazes, proteção de dados em sistemas distribuídos e IoT, além de uma arquitetura de hardware otimizada para suportar a crescente demanda por desempenho e proteção contra ameaças.