

# Sistemas Operacionais



# Introdução

- Um sistema operacional (SO) é um software que atua como intermediário entre o hardware de um computador e os programas de software que são executados nesse hardware.

# Introdução

- Ele fornece uma interface para que os usuários interajam com o computador e gerenciem recursos como processadores, memória, dispositivos de armazenamento e periféricos.

# Introdução

- Os sistemas operacionais desempenham várias funções essenciais, incluindo:
- Gerenciamento de recursos: Alocação e controle de recursos de hardware, como CPU, memória, dispositivos de armazenamento e periféricos.

# Introdução

- Facilitação da comunicação: Oferece uma interface para que os usuários interajam com o computador e para que os aplicativos se comuniquem com o hardware.

# Introdução

- Execução de programas: Gerencia a execução de programas e aplicativos, permitindo que múltiplos processos sejam executados simultaneamente.

# Introdução

- Gerenciamento de arquivos: Organiza e manipula arquivos de dados em dispositivos de armazenamento.

# Introdução

- Segurança: Protege o sistema contra ameaças, controlando o acesso a recursos e implementando medidas de segurança, como autenticação de usuários e controle de acesso.



# Introdução

- Gerenciamento de rede: Facilita a comunicação entre dispositivos em uma rede, fornecendo suporte para protocolos de rede e configurando conexões.

# Introdução

- Exemplos de sistemas operacionais incluem Windows, macOS, Linux, Android e iOS, cada um com suas próprias características e funcionalidades específicas.

# História

## Pioneiros e Sistemas Primitivos (Décadas de 1940 e 1950):

Nos primórdios da computação, os primeiros sistemas operacionais eram simples e rudimentares. Os computadores eram enormes e caros, e os programas eram escritos diretamente em linguagem de máquina.

# História

Pioneiros e Sistemas Primitivos (Décadas de 1940 e 1950):

Exemplos incluem o GM-NAA I/O System, usado no IBM 704, e o IBSYS (IBM Symbolic Operating System), utilizado nos mainframes IBM 709 e 7090.

# História

## Sistemas Batch (Década de 1960):

Com a introdução de mainframes mais poderosos, surgiram os sistemas batch, nos quais os trabalhos eram agrupados em lotes e processados em sequência.

Um marco importante foi o OS/360 da IBM, lançado em 1966, que era um sistema operacional abrangente projetado para funcionar em toda a linha de mainframes da IBM.

# História

## Sistemas de Tempo Compartilhado e Unix (Década de 1970):

O conceito de tempo compartilhado emergiu, permitindo que vários usuários interagissem com o computador simultaneamente. O Unix, desenvolvido na Bell Labs por Ken Thompson e Dennis Ritchie, foi um dos primeiros sistemas a explorar esse conceito.

Outros sistemas notáveis incluem o CP/M, desenvolvido por Gary Kildall, que se tornou amplamente utilizado em microcomputadores na década de 1970 e início da década de 1980.

# História

Era dos Microcomputadores e Sistemas Operacionais de Desktop (Décadas de 1980 e 1990):

Com o advento dos microcomputadores pessoais, surgiram sistemas operacionais populares como o MS-DOS, desenvolvido pela Microsoft, e o sistema operacional Apple DOS (posteriormente substituído pelo macOS).

O Windows, lançado pela Microsoft em 1985, trouxe uma interface gráfica de usuário (GUI) para os PCs e se tornou o sistema operacional dominante no mercado de desktops.

# História

Evolução Contínua e Diversificação (Década de 2000 até hoje):

O início do século XXI testemunhou a proliferação de sistemas operacionais móveis, como o Android, desenvolvido pelo Google, e o iOS, desenvolvido pela Apple, para smartphones e tablets.

O Linux, um sistema operacional de código aberto, ganhou popularidade e foi amplamente adotado em servidores e dispositivos embarcados.

Além disso, houve avanços significativos em áreas como virtualização, computação em nuvem, segurança e integração de dispositivos inteligentes (IoT).



# Estruturas

## Monolíticos:

Os sistemas operacionais monolíticos foram os primeiros a serem desenvolvidos e são caracterizados por terem todo o sistema operacional executado como um único programa no espaço do núcleo do sistema (kernel).

O kernel do sistema operacional monolítico é responsável por fornecer serviços essenciais, como gerenciamento de memória, gerenciamento de processos, gerenciamento de dispositivos e sistema de arquivos.

Exemplos de sistemas operacionais monolíticos incluem o MS-DOS e versões mais antigas do Linux.

# Estruturas

## Camadas:

Os sistemas operacionais em camadas dividem o sistema operacional em camadas hierárquicas, onde cada camada fornece um conjunto específico de funcionalidades e serviços.

As camadas são organizadas de forma que uma camada inferior forneça serviços para as camadas superiores, enquanto as camadas superiores possam chamar serviços de camadas inferiores.

# Estruturas

Camadas:

Esse modelo facilita a modularidade e a manutenção do sistema operacional, pois cada camada pode ser desenvolvida e modificada independentemente.

Exemplos de sistemas operacionais em camadas incluem o THEOS e o OpenVMS.

# Estruturas

## Máquinas Virtuais:

Os sistemas operacionais baseados em máquinas virtuais usam uma camada de software intermediária, conhecida como hipervisor ou monitor de máquina virtual, para criar e gerenciar várias máquinas virtuais em um único hardware físico.

Cada máquina virtual é uma instância isolada de um sistema operacional completo, permitindo a execução de múltiplos sistemas operacionais diferentes no mesmo hardware.

# Estruturas

## Máquinas Virtuais:

Esse modelo é comumente usado em ambientes de virtualização de servidores, onde várias máquinas virtuais podem ser provisionadas e gerenciadas de forma eficiente em um único servidor físico.

Exemplos de sistemas operacionais baseados em máquinas virtuais incluem VMware ESXi, Microsoft Hyper-V e Xen.

# Estruturas

## Modelo Cliente-Servidor:

No modelo cliente-servidor, o sistema operacional é dividido em duas partes distintas: o servidor que fornece serviços e o cliente que solicita e utiliza esses serviços.

# Estruturas

## Modelo Cliente-Servidor:

O servidor é responsável por executar tarefas essenciais, como gerenciamento de recursos de hardware, segurança e comunicação de rede.

Os clientes são os usuários ou aplicativos que interagem com o servidor para acessar recursos e serviços oferecidos pelo sistema operacional.

# Estruturas

## Modelo Cliente-Servidor:

Esse modelo é comumente usado em sistemas operacionais de rede, onde múltiplos clientes podem se conectar a um servidor central para compartilhar recursos e dados.

Exemplos de sistemas operacionais que seguem o modelo cliente-servidor incluem várias distribuições de sistemas operacionais de rede, como Windows Server e Linux Server.



# Estruturas

Essas diferentes estruturas de sistema operacional refletem abordagens distintas para organizar e gerenciar os recursos de hardware e software de um computador, cada uma com suas próprias vantagens e desvantagens em termos de desempenho, segurança, escalabilidade e flexibilidade.

A escolha da estrutura adequada depende dos requisitos específicos do sistema e das preferências do desenvolvedor.