****

Estruturas de Sistemas Operacionais

*Disciplina, Termo e Período*

Guilherme C. M. d Santos – guilherme.santos353@fatec.sp.gov.br – RA: 1571432312009

Athos Aurélio A. F. Barbosa – athos.barbosa@fatec.sp.gov.br – RA: 1571432312005

Presidente Prudente-SP

03/2023

**1. Arquitetura monolitica**

O sistema operacional para arquitetura monolítica é caracterizado por ter todos os seus serviços e funções no mesmo espaço de endereçamento. Isso significa que todas as funcionalidades do sistema operacional, como gerenciamento de memória, gerenciamento de processos e drivers de dispositivos, estão integrados em um único executável, o kernel.

Os sistemas operacionais para arquitetura monolítica são usados em uma ampla variedade de dispositivos, desde servidores até desktops e laptops. Alguns exemplos de sistemas operacionais para arquitetura monolítica incluem:

Windows - O Windows é um dos sistemas operacionais mais populares do mundo. É um sistema operacional para arquitetura monolítica que inclui um kernel, um conjunto de drivers de dispositivos e um conjunto de serviços para gerenciamento de memória, processos e arquivos.

Linux - O Linux é um sistema operacional de código aberto que é executado em uma ampla variedade de dispositivos, desde servidores até dispositivos móveis. O kernel do Linux é um kernel monolítico que inclui serviços para gerenciamento de processos, memória e sistemas de arquivos.

MacOS - O MacOS é o sistema operacional padrão para computadores Apple. Ele é baseado no kernel XNU, que é um kernel híbrido monolítico/microkernel. O XNU inclui serviços para gerenciamento de memória, processos e arquivos, bem como drivers de dispositivos.

**1.1 Vantagens e Desvantagens**

Os sistemas operacionais para arquitetura monolítica têm algumas vantagens e desvantagens. Uma vantagem é que eles geralmente têm melhor desempenho do que sistemas operacionais baseados em microkernel, pois o acesso aos serviços do sistema operacional é mais rápido. No entanto, eles podem ser menos estáveis do que sistemas operacionais baseados em microkernel, pois um erro em um serviço do sistema pode afetar todo o sistema operacional.

**1.1.1 Conclusão**

Em resumo, os sistemas operacionais para arquitetura monolítica são amplamente utilizados e oferecem serviços integrados em um único executável. Alguns exemplos de sistemas operacionais para arquitetura monolítica incluem o Windows, o Linux e o MacOS. Eles têm vantagens e desvantagens e são escolhidos de acordo com as necessidades do usuário e do dispositivo.

**2. Micro Kernel**

O sistema operacional para arquitetura micro-kernel é caracterizado por ter um kernel mínimo, com apenas as funções essenciais, e a maioria das funcionalidades do sistema operacional são executadas como processos do usuário. Isso significa que serviços como gerenciamento de memória, gerenciamento de processos e drivers de dispositivos são executados fora do kernel e se comunicam através de mensagens.

Os sistemas operacionais para arquitetura micro-kernel são usados em sistemas críticos, onde a segurança e a confiabilidade são essenciais. Alguns exemplos de sistemas operacionais para arquitetura micro-kernel incluem:

QNX - O QNX é um sistema operacional para arquitetura micro-kernel usado em sistemas embarcados e em tempo real. Ele é projetado para ser seguro e confiável, com uma arquitetura modular que permite que os serviços do sistema operacional sejam facilmente adicionados ou removidos.

L4 - O L4 é um kernel para arquitetura micro-kernel usado em sistemas embarcados e sistemas em tempo real. Ele é conhecido por sua segurança e confiabilidade, e é usado em sistemas militares e aeroespaciais.

MINIX - O MINIX é um sistema operacional de código aberto para arquitetura micro-kernel usado em dispositivos embarcados e em sistemas educacionais. Ele é projetado para ser simples e seguro, com um kernel mínimo e serviços do sistema operacional executados como processos do usuário.

**2.1 Vantagens e Desvantagens**

Os sistemas operacionais para arquitetura micro-kernel têm algumas vantagens e desvantagens. Uma vantagem é que eles são mais seguros e confiáveis do que sistemas operacionais para arquitetura monolítica, pois os serviços do sistema operacional são executados como processos do usuário e são mais isolados do kernel. No entanto, eles podem ter um desempenho um pouco pior do que sistemas operacionais para arquitetura monolítica, pois as comunicações entre processos podem ser mais lentas.

**2.1.1 Conclusão**

Em resumo, os sistemas operacionais para arquitetura micro-kernel são usados em sistemas críticos, onde a segurança e a confiabilidade são essenciais. Alguns exemplos de sistemas operacionais para arquitetura micro-kernel incluem o QNX, o L4 e o MINIX. Eles têm vantagens e desvantagens e são escolhidos de acordo com as necessidades do usuário e do dispositivo.

**3. Arquitetura Cliente-Servidor**

A arquitetura cliente-servidor é um modelo de comunicação entre computadores, no qual um ou mais computadores (clientes) se conectam a um servidor central para acessar recursos ou serviços compartilhados. Este modelo é amplamente utilizado em sistemas distribuídos, nos quais vários dispositivos precisam se comunicar entre si para realizar uma tarefa. Para entender melhor como funciona essa arquitetura, imagine uma rede de computadores em que um servidor central é responsável por armazenar e gerenciar dados e recursos compartilhados, como um banco de dados ou um sistema de arquivos.

Os clientes se conectam a esse servidor para acessar esses recursos e executar tarefas específicas. Existem várias implementações diferentes de arquitetura cliente-servidor, que podem variar em termos de como os clientes e servidores se comunicam entre si. Uma das abordagens mais comuns é a utilização de protocolos de rede, como HTTP ou FTP, que permitem que os clientes se conectem ao servidor e solicitem informações ou arquivos específicos.

Exemplos de sistemas operacionais que utilizam a arquitetura cliente-servidor incluem:

Windows - o sistema operacional Windows da Microsoft usa a arquitetura cliente-servidor em vários componentes, como o Active Directory e o sistema de gerenciamento de impressão.

Linux - o Linux é um sistema operacional de código aberto que também faz uso extensivo da arquitetura cliente-servidor em vários componentes, como o sistema de gerenciamento de arquivos NFS e o servidor web Apache.

MacOS - o sistema operacional da Apple também usa a arquitetura cliente-servidor em vários componentes, como o sistema de gerenciamento de impressão CUPS e o servidor de compartilhamento de arquivos AFP.

**3.1 Vantagens e Desvantagens**

Algumas vantagens da arquitetura cliente-servidor incluem:

Escalabilidade: a arquitetura cliente-servidor é altamente escalável, permitindo que novos clientes sejam adicionados facilmente ao sistema conforme a demanda aumenta.

Segurança: os servidores centralizados podem ser mais seguros, uma vez que os dados e recursos são mantidos em um único local e podem ser protegidos com medidas de segurança robustas.

Centralização de recursos: a arquitetura cliente-servidor permite que os recursos sejam compartilhados de forma centralizada, o que pode ser mais eficiente do que ter várias cópias do mesmo recurso distribuídas por toda a rede.

Algumas desvantagens da arquitetura cliente-servidor incluem:

Depende de um servidor central: a arquitetura cliente-servidor depende de um servidor central, o que pode tornar o sistema mais vulnerável a falhas e interrupções.

Complexidade: a implementação de uma arquitetura cliente-servidor pode ser complexa, especialmente em sistemas distribuídos com muitos clientes e servidores.

Desempenho: em alguns casos, a arquitetura cliente-servidor pode ser menos eficiente do que outras arquiteturas de rede, especialmente em redes com alta latência ou com muitos clientes.

**3.1.1 Conclusão**

Em resumo, a arquitetura cliente-servidor é um modelo de comunicação entre computadores no qual os clientes se conectam a um servidor central para acessar recursos ou serviços compartilhados. Exemplos de sistemas operacionais que usam essa arquitetura incluem o Windows, o Linux e o MacOS. As vantagens da arquitetura cliente-servidor incluem escalabilidade, segurança e centralização de recursos, enquanto as desvantagens incluem dependência de um servidor central, complexidade e possível impacto no desempenho.

**4. Arquitetura de máquina virtual**

Sistemas operacionais para arquitetura de máquina virtual são sistemas operacionais que são executados em uma camada de software que simula uma plataforma de hardware completa. Essa camada de software é chamada de máquina virtual e permite que o sistema operacional execute em diferentes plataformas de hardware sem a necessidade de modificar o código do sistema operacional. Essa abstração do hardware subjacente também permite que várias máquinas virtuais sejam executadas em um único servidor físico, tornando essa arquitetura popular em ambientes de data center.

Alguns exemplos de sistemas operacionais para arquitetura de máquina virtual:

VMware ESX - O VMware ESXi é um sistema operacional para arquitetura de máquina virtual usado em data centers e em ambientes empresariais. Ele permite que vários sistemas operacionais sejam executados em uma única máquina física e fornece recursos de gerenciamento centralizado para facilitar a administração do ambiente de TI.

Microsoft Hyper-V - O Microsoft Hyper-V é um sistema operacional para arquitetura de máquina virtual usado em ambientes Windows. Ele fornece recursos de virtualização de hardware para permitir que várias máquinas virtuais sejam executadas em um único servidor físico. O Hyper-V também fornece recursos de gerenciamento centralizado e integração com outros produtos da Microsoft, como o System Center.

Oracle VirtualBox - O Oracle VirtualBox é um sistema operacional para arquitetura de máquina virtual que é executado em uma variedade de plataformas, incluindo Windows, macOS e Linux. Ele é usado principalmente em ambientes de desenvolvimento e teste, permitindo que os desenvolvedores criem e testem aplicativos em várias plataformas de hardware sem precisar de hardware físico adicional.

Docker - O Docker é uma plataforma para desenvolvimento, empacotamento e execução de aplicativos em contêineres. Ele usa uma arquitetura de máquina virtual leve, chamada de contêiner, para isolar aplicativos de outros processos no sistema operacional. O Docker é usado em ambientes de desenvolvimento e produção para simplificar o processo de implantação e gerenciamento de aplicativos.

**4.1 Vantagens e Desvantagens**

Vantagens:

Isolamento: Os sistemas operacionais de VMs são executados em um ambiente virtual isolado, separado do hardware físico e dos outros sistemas operacionais de VMs em execução. Isso oferece uma camada adicional de segurança e proteção contra ataques maliciosos.

Flexibilidade: As VMs podem ser criadas e implantadas rapidamente, permitindo que você execute vários sistemas operacionais em uma única máquina física. Isso pode ser útil para testar diferentes configurações de sistema ou para executar aplicativos em plataformas específicas.

Portabilidade: As VMs são independentes do hardware físico subjacente, o que significa que podem ser movidas facilmente entre servidores físicos ou até mesmo para a nuvem.

Desvantagens:

Overhead: As VMs adicionam camadas adicionais de software e hardware virtualizado, o que pode resultar em algum overhead de desempenho em comparação com sistemas operacionais instalados diretamente em hardware físico.

Complexidade: Gerenciar múltiplas VMs pode ser complexo, especialmente se elas estiverem sendo executadas em vários hosts físicos.

Custos: As VMs requerem recursos de hardware adicionais, como memória e CPU, o que pode aumentar os custos de hardware e energia. Além disso, as licenças de software para sistemas operacionais de VMs podem ser mais caras do que para sistemas operacionais instalados diretamente em hardware físico.

**4.1.1 Conclusão**

Em resumo, sistemas operacionais para arquitetura de máquina virtual são sistemas operacionais que são executados em uma camada de software que simula uma plataforma de hardware completa. Essa arquitetura permite que os sistemas operacionais sejam executados em diferentes plataformas de hardware sem a necessidade de modificar o código do sistema operacional. Existem várias opções de sistemas operacionais para arquitetura de máquina virtual disponíveis, como o VMware ESXi, o Microsoft Hyper-V, o Oracle VirtualBox e o Docker. Cada sistema operacional tem suas próprias vantagens e desvantagens, e a escolha de um sistema operacional dependerá das necessidades do usuário e do ambiente de TI.

**5. Arquitetura em camadas**

Sistemas operacionais para arquitetura em camadas são sistemas operacionais que são projetados em camadas que se comunicam umas com as outras para fornecer funcionalidades do sistema operacional. Cada camada é responsável por fornecer um conjunto específico de funcionalidades e se comunica com outras camadas por meio de interfaces bem definidas. Essa abordagem torna o sistema operacional mais modular e facilita a manutenção e o desenvolvimento de novos recursos.

Abaixo, apresentamos

**5.1 Exemplos**

Alguns exemplos de sistemas operacionais para arquitetura em camadas:

UNIX - O UNIX é um sistema operacional de código aberto e baseado em texto que foi desenvolvido em 1969. Ele usa uma arquitetura em camadas, com cada camada fornecendo funcionalidades específicas. A camada mais baixa fornece acesso ao hardware, enquanto a camada mais alta fornece uma interface de usuário. O UNIX é usado principalmente em ambientes de servidores e é conhecido por sua estabilidade e confiabilidade.

Windows NT - O Windows NT é um sistema operacional da Microsoft que usa uma arquitetura em camadas. Ele é composto por três camadas principais: a camada de usuário, a camada de kernel e a camada de hardware. A camada de usuário fornece a interface de usuário do sistema operacional, enquanto a camada de kernel é responsável por gerenciar recursos do sistema, como memória e processos. A camada de hardware fornece acesso ao hardware subjacente do sistema.

Linux - O Linux é um sistema operacional de código aberto baseado no UNIX. Ele usa uma arquitetura em camadas semelhante ao UNIX, com cada camada fornecendo funcionalidades específicas. A camada mais baixa fornece acesso ao hardware, enquanto a camada mais alta fornece uma interface de usuário. O Linux é usado em uma variedade de dispositivos, incluindo servidores, dispositivos móveis e dispositivos de IoT.

MacOS - O MacOS é um sistema operacional desenvolvido pela Apple para seus computadores Mac. Ele usa uma arquitetura em camadas semelhante ao Windows e ao UNIX, com cada camada fornecendo funcionalidades específicas. A camada mais baixa fornece acesso ao hardware, enquanto a camada mais alta fornece uma interface de usuário. O MacOS é conhecido por sua facilidade de uso e integração com outros produtos da Apple, como o iPhone e o iPad.

**5.1.1 Conclusão**

Em resumo, sistemas operacionais para arquitetura em camadas são sistemas operacionais que são projetados em camadas que se comunicam umas com as outras para fornecer funcionalidades do sistema operacional. Existem várias opções de sistemas operacionais para arquitetura em camadas disponíveis, como o UNIX, o Windows NT, o Linux e o MacOS. Cada sistema operacional tem suas próprias vantagens e desvantagens, e a escolha de um sistema operacional dependerá das necessidades do usuário e do ambiente de TI.

**Referências:**

https://www.alura.com.br

https://dev.to

https://pt.wikibooks.org

https://materialpublic.imd.ufrn.br

univasf.edu.br