SO
Sistemas Operacionais

Prof. Rodrigo Martins
rodrigo.martins@francomontoro.com.br

Cronograma da Aula

- ARQUITETURA DE SISTEMAS OPERACIONAIS
  - Sistemas monolíticos,
  - Sistemas em camadas,
  - As máquinas virtuais,
  - Os exonúcleos e
  - Os sistemas cliente-servidor.
- Exercícios

Introdução

- O sistema operacional é formado por um conjunto de rotinas que oferece serviços aos usuários e às suas aplicações.

- Esse conjunto de rotinas é denominado **núcleo do sistema, ou kernel.**

Introdução

- Há três maneiras distintas de os usuários se comunicarem com o kernel do sistema operacional.
- Uma delas é por intermédio das chamadas rotinas do sistema realizadas por aplicações.
- Os usuários podem interagir com o núcleo mais amigavelmente por meio de utilitários ou linguagem de comandos.

Introdução

- Modelo de camadas detalhando a estrutura do sistema operacional e suas interfaces.

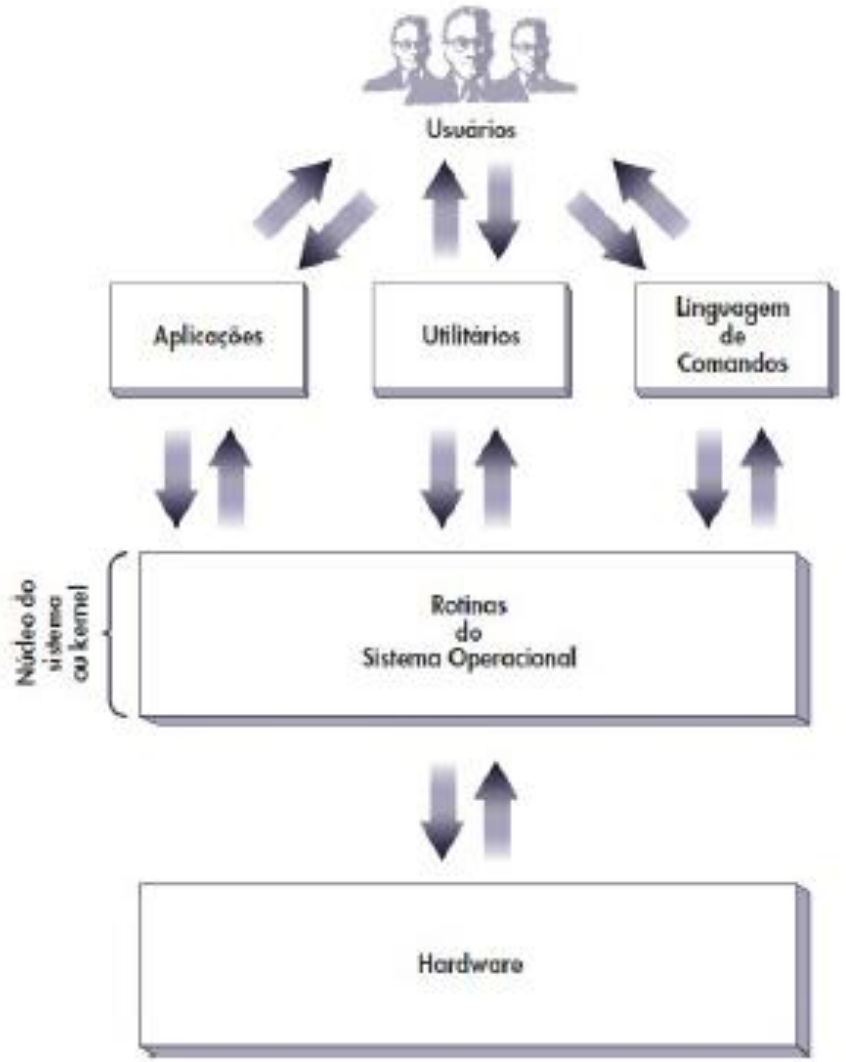

Fig. 4.1 Estrutura do sistema operacional.

Funções do Núcleo

- Diferentemente de uma aplicação convencional, com sequenciamento de início, meio e fim, as rotinas do sistema são executadas concorrentemente sem uma ordem predefinida, com base em eventos dissociados do tempo (eventos assíncronos).

- Muitos desses eventos estão relacionados ao hardware e a tarefas internas do próprio sistema operacional.

Funções do Núcleo

- As principais funções do núcleo encontradas nos sistemas operacionais são:
  - tratamento de interrupções e exceções;
  - criação e eliminação de processos e threads;
  - sincronização e comunicação entre processos e threads;
  - escalonamento e controle dos processos e threads;
  - gerência de memória;
  - gerência do sistema de arquivos;
  - gerência de dispositivos de E/S;
  - suporte a redes locais e distribuídas;
  - contabilização do uso do sistema;
  - auditoria e segurança do sistema.

Modo de Acesso

- Uma preocupação que surge nos projetos de sistemas operacionais é a implementação de mecanismos de proteção ao núcleo do sistema e de acesso aos seus serviços. Caso uma aplicação, que tenha acesso ao núcleo, realize uma operação que altere sua integridade, todo o sistema poderá ficar comprometido e inoperante.

- Muitas das principais implementações de segurança de um sistema operacional utilizam um mecanismo presente no hardware dos processadores, conhecido como **modo de acesso**.

Modo de Acesso

- Em geral, os processadores possuem dois modos de acesso: **modo usuário** e **modo kernel**.

- Quando o **processador** trabalha no **modo usuário**, uma aplicação só pode executar instruções conhecidas como não privilegiadas, tendo acesso a um **número reduzido de instruções**.

- Enquanto no **modo kernel** a aplicação pode ter acesso ao **conjunto total de instruções do processador**.

Rotinas do Sistema Operacional e System Calls

- As **rotinas do sistema operacional** compõem o núcleo do sistema, oferecendo serviços aos usuários e suas aplicações.

- Todas as funções do núcleo são implementadas por rotinas do sistema que necessariamente possuem em seu código instruções privilegiadas.

- A partir desta condição, para que estas rotinas possam ser executadas o processador deve estar obrigatoriamente em modo kernel, o que exige a implementação de mecanismos de proteção para garantir a confiabilidade do sistema.

Rotinas do Sistema Operacional e System Calls

- Todo o controle de execução de rotinas do sistema operacional é realizado pelo mecanismo conhecido como **system call**.

- Toda vez que uma aplicação desejar **chamar uma rotina do sistema operacional**, o mecanismo de **system call é ativado**.

Rotinas do Sistema Operacional e System Calls

- Considerando que a aplicação possua o devido privilégio para chamar a rotina do sistema desejada, o sistema operacional primeiramente salva o conteúdo corrente dos registradores, troca o modo de acesso do processador de usuário para kernel e realiza o desvio para a rotina alterando o registrador PC com o endereço da rotina chamada.

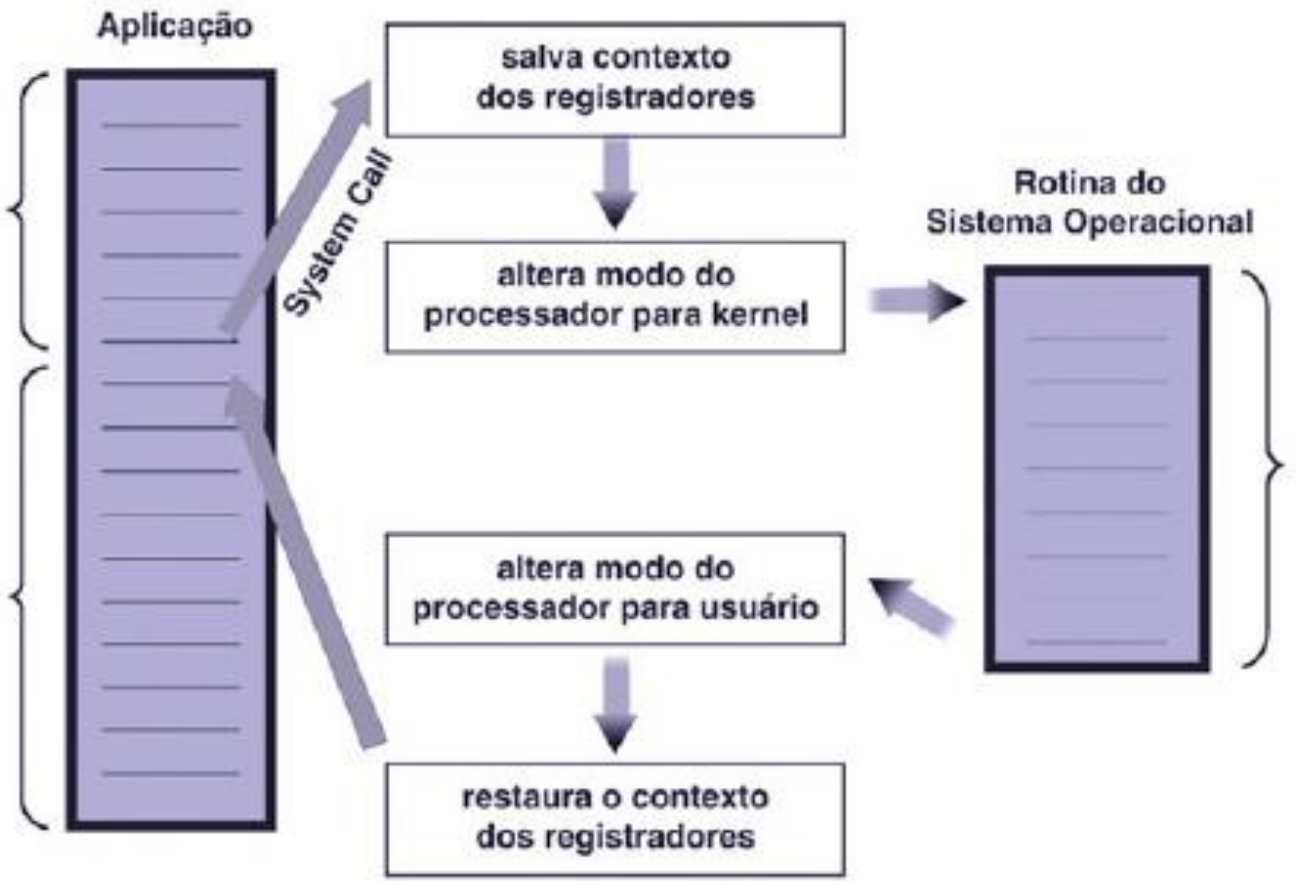

Fig. 4.2 Chamada a uma rotina do sistema (a).

Chamada a Rotinas do Sistema Operacional

- As rotinas do sistema e o mecanismo de system call podem ser entendidos como uma porta de entrada para o núcleo do sistema operacional e a seus serviços. Sempre que uma aplicação desejar algum serviço do sistema, deve ser realizada uma chamada a uma de suas rotinas através de uma system call.

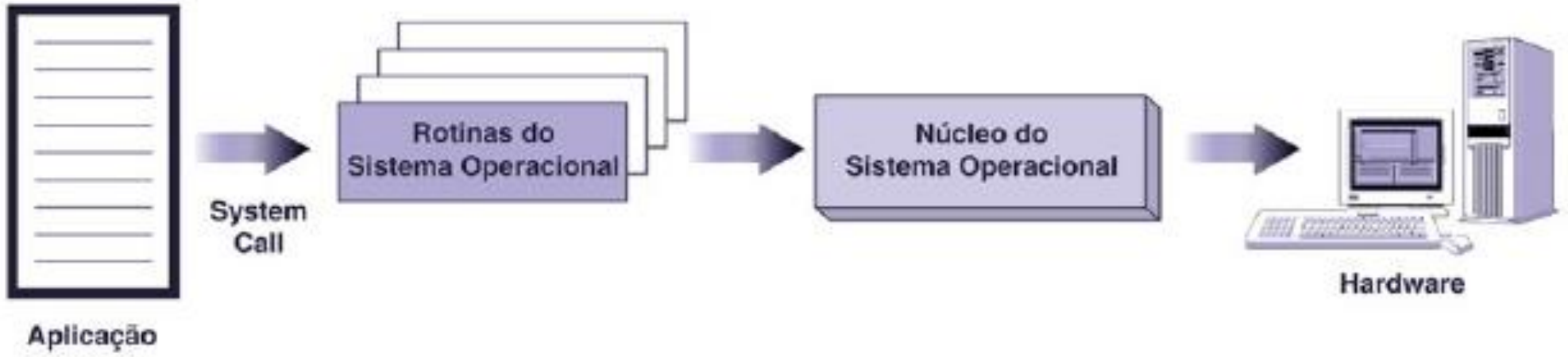

Fig. 4.3 Chamada a uma rotina do sistema (b).

Chamada a Rotinas do Sistema Operacional

- Existem duas maneiras distintas de chamada a uma rotina do sistema operacional: **explícita** e **implícita**.

- A **chamada explícita** é a descrita anteriormente, em que no código do programa há uma função explicitando a chamada a rotina do sistema com passagem de parâmetro.

- Já a **chamada implícita** é realizada por intermédio de um comando da própria linguagem de programação. Quando este comando é traduzido para uma instrução de mais baixo nível, há uma conversão do comando em uma chamada da função.

Chamada a Rotinas do Sistema Operacional

- As rotinas do sistema podem ser divididas por grupos de função:

Tabela 4.1 **Funções das system calls**

Funções	System calls
Gerência de processos e threads	Criação e eliminação de processos e threads
	Alteração das características de processos e threads
	Sincronização e comunicação entre processos e threads
	Obtenção de informações sobre processos e threads
Gerência de memória	Alocação e desalocação de memória
Gerência do sistema de arquivos	Criação e eliminação de arquivos e diretórios
	Alteração das características de arquivos e diretórios
	Abertura e fechamento de arquivos
	Leitura e gravação em arquivos
	Obtenção de informações sobre arquivos e diretórios
Gerência de dispositivos	Alocação e desalocação de dispositivos
	Operações de entrada/saída em dispositivos
	Obtenção de informações sobre dispositivos

Linguagem de Comandos

- A linguagem de comandos, ou linguagem de controle, permite que o usuário se comunique de uma forma simples com o sistema operacional, capacitando-o a executar diversas tarefas específicas do sistema como criar, ler ou eliminar arquivos, consultar diretórios ou verificar a data e a hora armazenadas no sistema.

Tabela 4.2 Exemplos de comandos do MS Windows

Comando	Descrição
dir	Lista o conteúdo de um diretório
cd	Altera o diretório default
type	Exibe o conteúdo de um arquivo
del	Elimina arquivos
mkdir	Cria um diretório
ver	Mostra a versão do Windows

Linguagem de Comandos

- Cada comando, depois de digitado pelo usuário, é interpretado pelo shell ou interpretador de comandos, que verifica a sintaxe do comando, faz chamadas a rotinas do sistema e apresenta um resultado ou uma mensagem informativa.

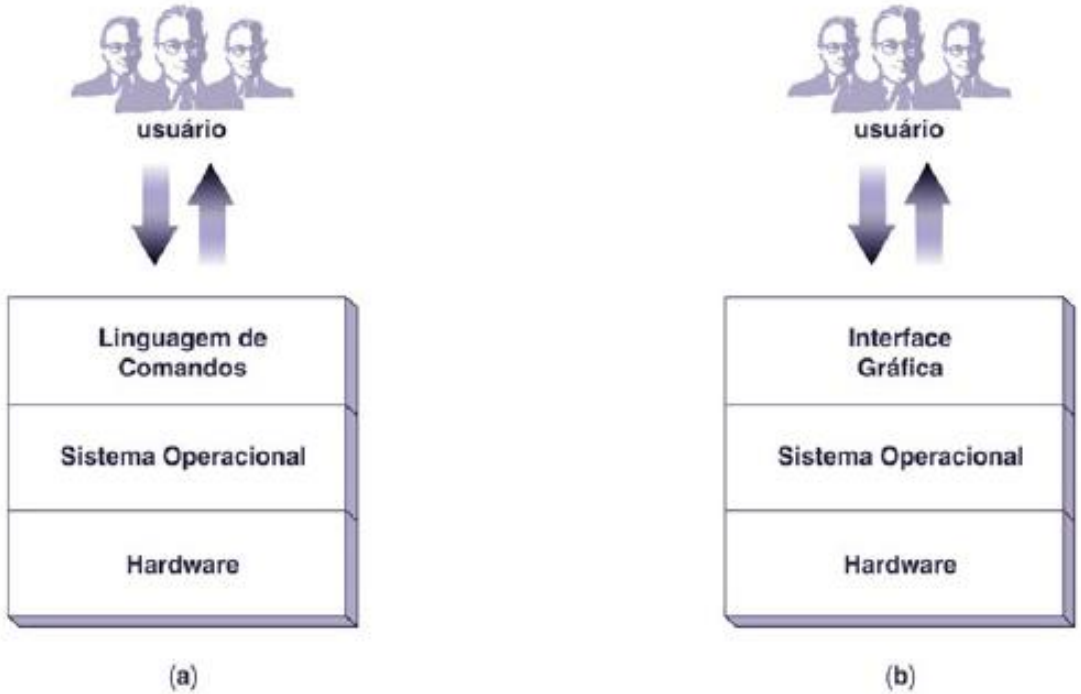

Fig. 4.4 Interface do usuário com o sistema operacional.

Ativação/Desativação do Sistema

- Inicialmente, quando um computador é ligado não há sistema operacional carregado na memória da máquina. Em geral, o sistema operacional reside em um disco rígido, podendo também estar armazenado em outros dispositivos de memória secundária, como CD ou DVD.

- Os componentes do sistema operacional devem ser carregados para a memória principal toda vez que o computador é ligado por intermédio de um procedimento denominado **ativação do sistema ou boot.**

Ativação/Desativação do Sistema

- O procedimento de ativação se inicia com a execução de um programa chamado boot loader, que se localiza em um endereço fixo de uma memória ROM da máquina.

- Este programa chama a execução de outro programa conhecido como POST (Power-On Self Test), que identifica possíveis problemas de hardware no equipamento.

Ativação/Desativação do Sistema

- Se um dispositivo com o sistema operacional é encontrado, um conjunto de instruções é carregado para memória e localizado em um bloco específico do dispositivo conhecido como setor de boot (boot sector).

- A partir da execução deste código, o sistema operacional é finalmente carregado para a memória principal.

Ativação/Desativação do Sistema

- Além da carga, a ativação do sistema também consiste na execução de arquivos de inicialização onde são especificados procedimentos de customização e configuração de hardware e software específicos para cada ambiente.

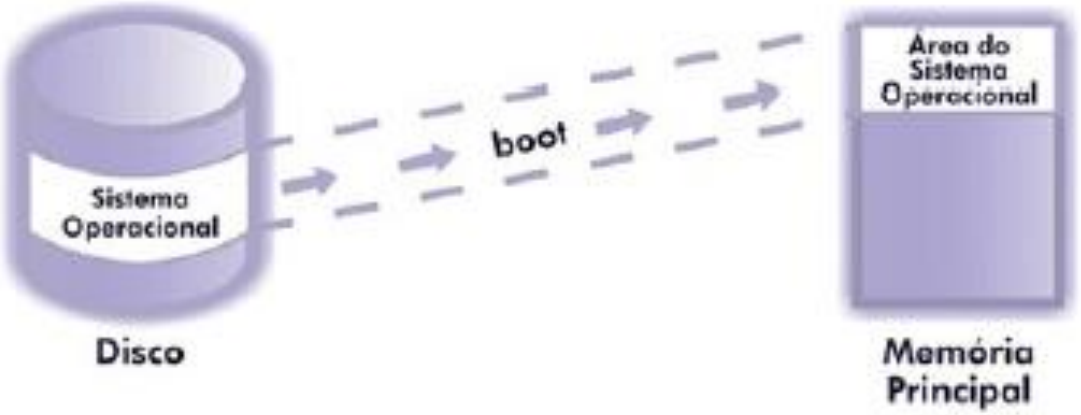

Fig. 4.5 Ativação do sistema.

Ativação/Desativação do Sistema

- Na maioria dos sistemas também existe o **processo de desativação ou shutdown**.

- Este procedimento permite que as aplicações e componentes do sistema operacional sejam desativados ordenadamente, garantindo, desta forma, sua integridade.

Arquiteturas do Núcleo

- Uma tendência no projeto de sistemas operacionais modernos é a utilização de técnicas de orientação por objetos, o que leva para o projeto do núcleo do sistema todas as vantagens deste modelo de desenvolvimento de software.

- Existe uma série de vantagens na utilização de programação por objetos no projeto e na implementação de sistemas operacionais.

Arquiteturas do Núcleo

- Os principais benefícios são:
  - melhoria na organização das funções e recursos do sistema;
  - redução no tempo de desenvolvimento;
  - maior facilidade na manutenção e extensão do sistema;
  - facilidade de implementação do modelo de computação distribuída.

Arquiteturas do Núcleo

- A estrutura do núcleo do sistema operacional, ou seja, a maneira como o código do sistema é organizado e o inter-relacionamento de seus diversos componentes, pode variar conforme a concepção do projeto.

- As principais arquiteturas dos sistemas operacionais são:
  - arquitetura monolítica,
  - arquitetura de camadas,
  - máquina virtual e
  - arquitetura microkernel.

Arquitetura Monolítica

- A **arquitetura monolítica** pode ser comparada com uma aplicação formada por vários módulos que são compilados separadamente e depois linkados, formando um grande e único programa executável, onde os módulos podem interagir livremente.

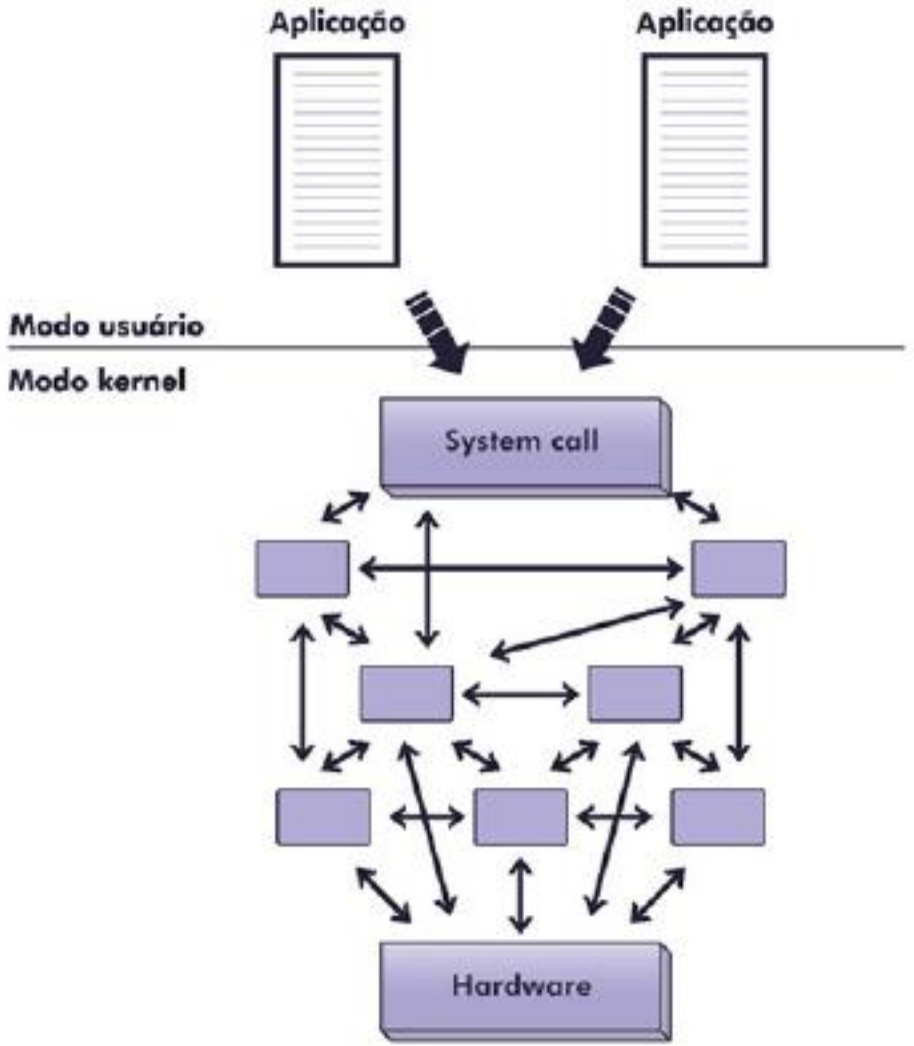

Fig. 4.6 Arquitetura monolítica.

Arquitetura Monolítica

- Os primeiros sistemas operacionais foram desenvolvidos com base neste modelo, o que tornava seu desenvolvimento e, principalmente, sua manutenção bastante difíceis.

- Devido a sua simplicidade e bom desempenho, a estrutura monolítica foi adotada no projeto do **MS-DOS** e nos primeiros sistemas **Unix**.

Arquitetura de Camadas

- Com o aumento da complexidade e do tamanho do código dos sistemas operacionais, técnicas de programação estruturada e modular foram incorporadas ao seu projeto.

- Na **arquitetura de camadas**, o sistema é dividido em níveis sobrepostos. Cada camada oferece um conjunto de funções que podem ser utilizadas apenas pelas camadas superiores.

Arquitetura de Camadas

- A vantagem da estruturação em camadas é isolar as funções do sistema operacional, facilitando sua manutenção e depuração, além de criar uma hierarquia de níveis de modos de acesso, protegendo as camadas mais internas.

- Uma desvantagem para o modelo de camadas é o desempenho. Cada nova camada implica uma mudança no modo de acesso.

Fig. 4.7 Arquitetura em camadas

Arquitetura de Camadas

- Atualmente, a maioria dos sistemas comerciais utiliza o modelo de duas camadas, onde existem os modos de acesso usuário (não privilegiado) e kernel (privilegiado).

- A maioria das versões do Unix e o Windows da Microsoft está baseada neste modelo.

Máquina Virtual

- Um sistema computacional é formado por níveis, onde a camada de nível mais baixo é o hardware.
- O modelo de máquina virtual, ou virtual machine (VM), cria um nível intermediário entre o hardware e o sistema operacional, denominado gerência de máquinas virtuais.
- Este nível cria diversas máquinas virtuais independentes, onde cada uma oferece uma cópia virtual do hardware, incluindo os modos de acesso, interrupções, dispositivos de E/S etc.

Máquina Virtual

- Este modelo cria o isolamento total entre cada VM, oferecendo grande segurança para cada máquina virtual.

- Se, por exemplo, uma VM executar uma aplicação que comprometa o funcionamento do seu sistema operacional, as demais máquinas virtuais não sofrerão qualquer problema.

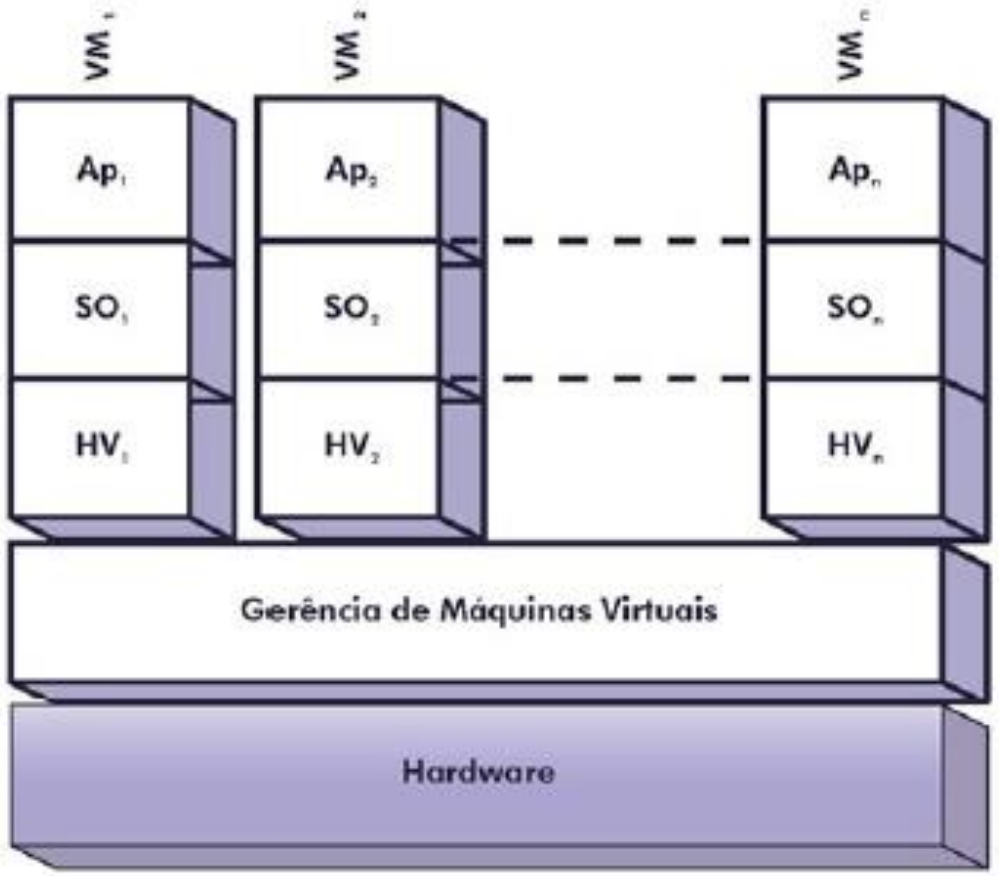

Fig. 4.8 Máquina virtual.

Máquina Virtual

- Existem diversas aplicações para a utilização de máquinas virtuais:
  - **Portabilidade de código:** A máquina virtual Java (Java Virtual Machine – JVM) funciona dessa forma.
  - **Consolidação de servidores:** é muito comum que os servidores de uma empresa funcionem com apenas parte de sua capacidade total de processamento, ou seja, existe a subutilização da UCP, memória principal e dispositivos de E/S dos sistemas computacionais.

Máquina Virtual

- Existem diversas aplicações para a utilização de máquinas virtuais:
  - **Aumento da disponibilidade:** uma vez que cada VM pode ser copiada facilmente para a memória secundária, como um disco, caso haja algum problema com uma VM ou com o próprio sistema onde está sendo processada, basta restaurar a cópia da VM em outro servidor e rapidamente o ambiente pode ser novamente disponibilizado.

Máquina Virtual

- Existem diversas aplicações para a utilização de máquinas virtuais:
  - **Facilidade de escalabilidade e balanceamento de carga:** caso o sistema no qual a VM está sendo processada fique sobrecarregado e apresente problemas de desempenho, a VM pode ser facilmente migrada para um novo ambiente que possua maior capacidade de processamento.

Máquina Virtual

- Existem diversas aplicações para a utilização de máquinas virtuais:
  - **Facilidade no desenvolvimento de software:** as VMs permitem a criação de um ambiente independente para o desenvolvimento e teste de software sem o comprometimento das máquinas de produção. Além disso, é possível o teste do software em diferentes sistemas operacionais e suas versões sem a necessidade de um hardware dedicado.

Arquitetura Microkernel

- Uma tendência nos sistemas operacionais modernos é tornar o núcleo do sistema operacional o menor e mais simples possível.

- Para implementar esta ideia, os serviços do sistema são disponibilizados através de processos, onde cada um é responsável por oferecer um conjunto específico de funções, como gerência de arquivos, gerência de processos, gerência de memória e escalonamento.

Arquitetura Microkernel

- Sempre que uma aplicação deseja algum serviço, é realizada uma solicitação ao processo responsável.
- A aplicação que solicita o serviço é chamada de **cliente**, enquanto o processo que responde à solicitação é chamado de **servidor**.
- A principal função do núcleo é realizar a comunicação, ou seja, a troca de mensagens entre cliente e servidor.
- A maioria das iniciativas nesta área está relacionada ao desenvolvimento de sistemas operacionais distribuídos.

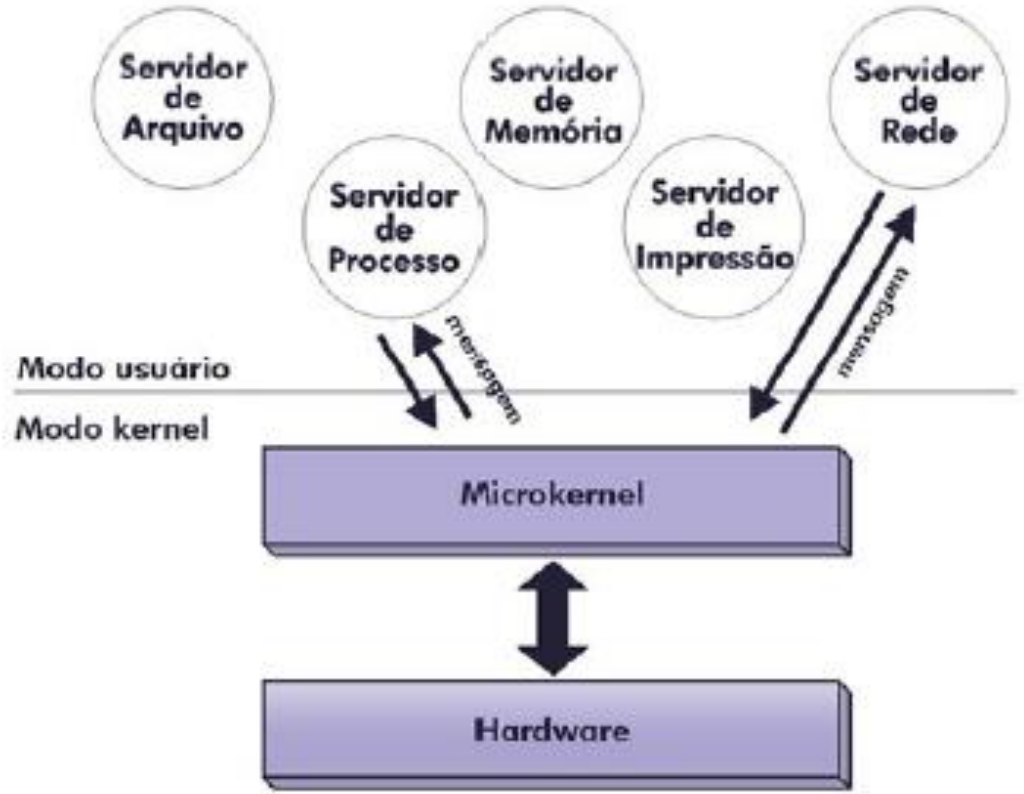

Fig. 4.9 Arquitetura microkernel.

Exercícios

1) O que é o núcleo do sistema e quais são suas principais funções?
2) O que são instruções privilegiadas e não privilegiadas? Qual a relação dessas instruções com os modos de acesso?
3) Por que as rotinas do sistema operacional possuem instruções privilegiadas?
4) O que é uma system call e qual sua importância para a segurança do sistema? Como as system calls são utilizadas por um programa?

Exercícios

5) Compare as arquiteturas monolítica e de camadas. Quais as vantagens e desvantagens de cada arquitetura?
6) Quais as vantagens do modelo de máquina virtual?
7) Como funciona o modelo cliente-servidor na arquitetura microkernel? Quais as vantagens e desvantagens dessa arquitetura?
8) Por que a utilização da programação orientada a objetos é um caminho natural para o projeto de sistemas operacionais?

Referências desta Aula

- TANENBAUM, A. S. Sistemas Operacionais Modernos, Prentice-Hall do Brasil, 4ª edição, 2016.

- MACHADO F. B., MAIA L. P. Arquitetura de Sistemas Operacionais, LTC, 5ª edição, 2014.

Fim
Obrigado

Rodrigo