INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PIAUÍ

TADS

Sistemas Operacionais

Prof. Ricardo Ramos

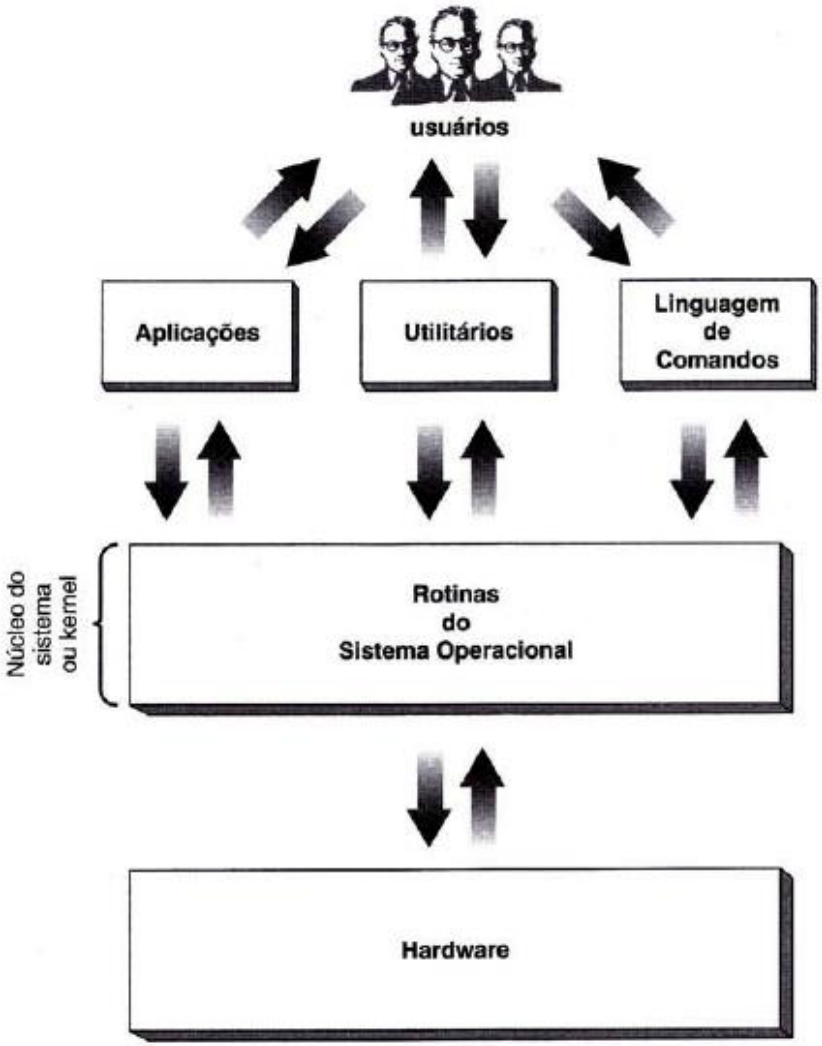
Conceitos Básicos - Capítulo 04

Estrutura do Sistema Operacional

4.1 Introdução

- O *kernel* (núcleo do sistema operacional) é um conjunto de rotinas que oferecem serviços aos usuários e às suas aplicações.

4.1 Introdução



Utilitários: compiladores e editores de texto.

4.2 Funções do Núcleo

As principais funções do *kernel* são:

- gerência de processos e threads;
- gerência de memória;
- gerência do sistema de arquivos;
- gerência de dispositivos de E/S;
- e outras.

4.3 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 a seus serviços.

Modos de acesso (mecanismo de segurança presente no hardware dos processadores)

4.3 Modo de Acesso

Modo usuário:

Uma aplicação só pode executar instruções não-privilegiadas (não oferecem riscos ao sistema), tendo acesso a um número reduzido de instruções.

Modo kernel:

A aplicação pode ter acesso ao conjunto total de instruções do processador, inclusive as privilegiadas (podem ocasionar sérios problemas à integridade do sistema se usadas de maneira indiscriminada).

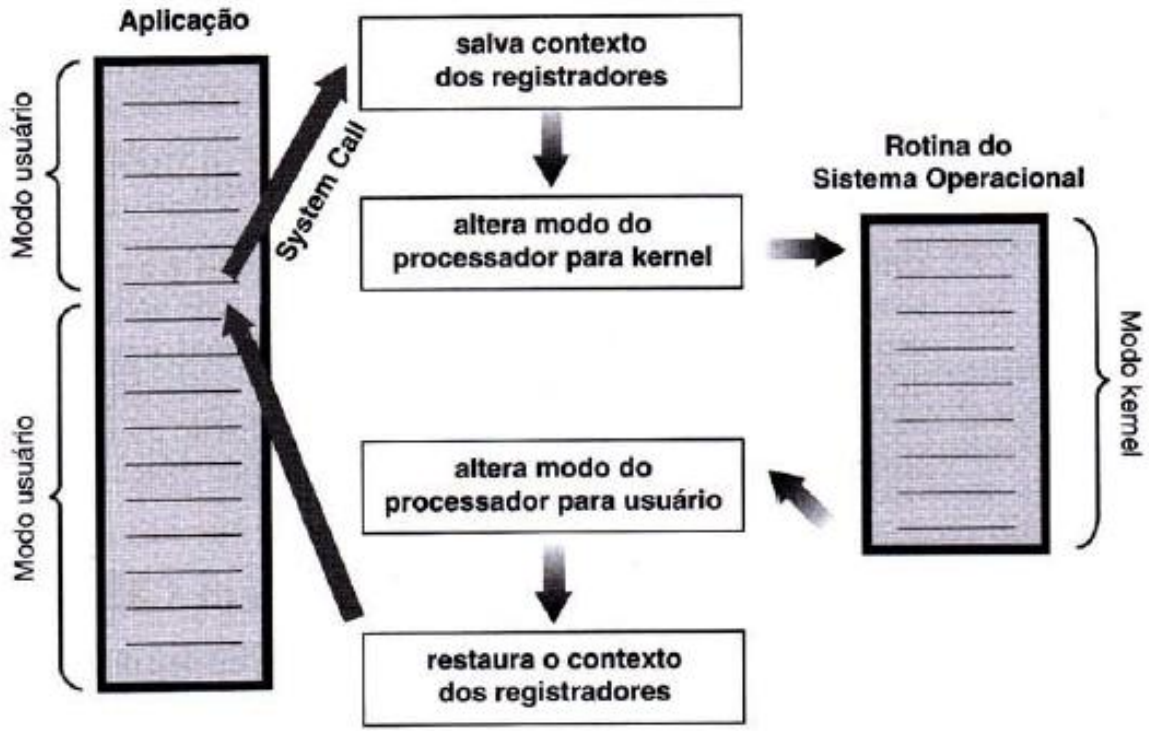
4.4 Rotinas do SO e *System Calls*

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

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

4.4 Rotinas do SO e *System Calls*

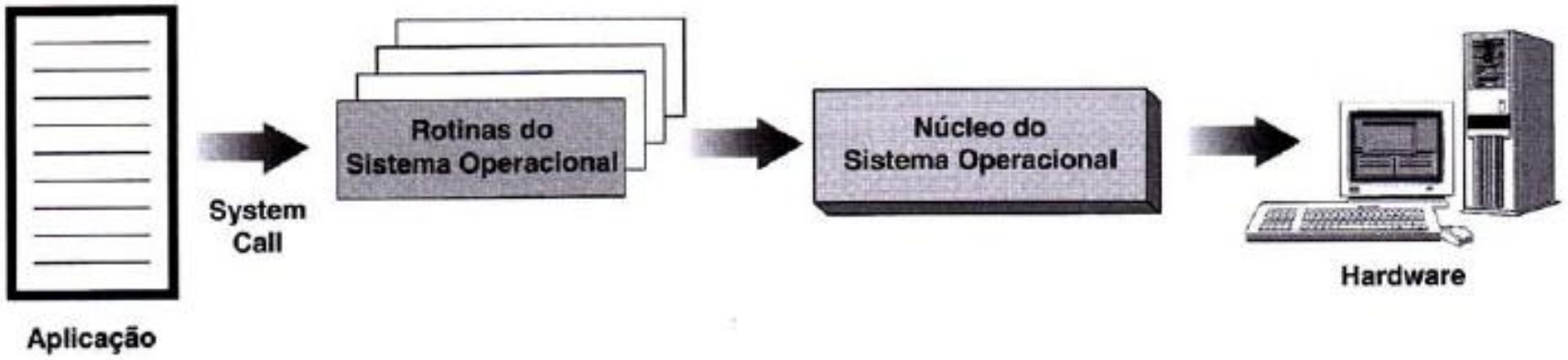
(porta de entrada para o *kernel*)



Ex: Acesso ao disco rígido (operação de E/S).

4.5 Chamada a Rotinas do SO

Sempre que uma aplicação desejar algum serviço do sistema, deve ser realizada uma chamada a uma de suas rotinas através de uma *system call*.



Semelhante a chamada de uma sub-rotina por um programa.

4.5 Chamada a Rotinas do SO

Cada SO possui seu próprio conjunto de rotinas, com nomes, parâmetros e formas de ativação específicos.

As rotinas do sistema podem ser divididas por grupos de função.

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 Abrir e fechar 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

4.6 Linguagem de Comandos

(ou linguagem de controle)

Permite comunicação simples com o SO.

O usuário dispõe de uma interface direta com o SO.

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

4.6 Linguagem de Comandos

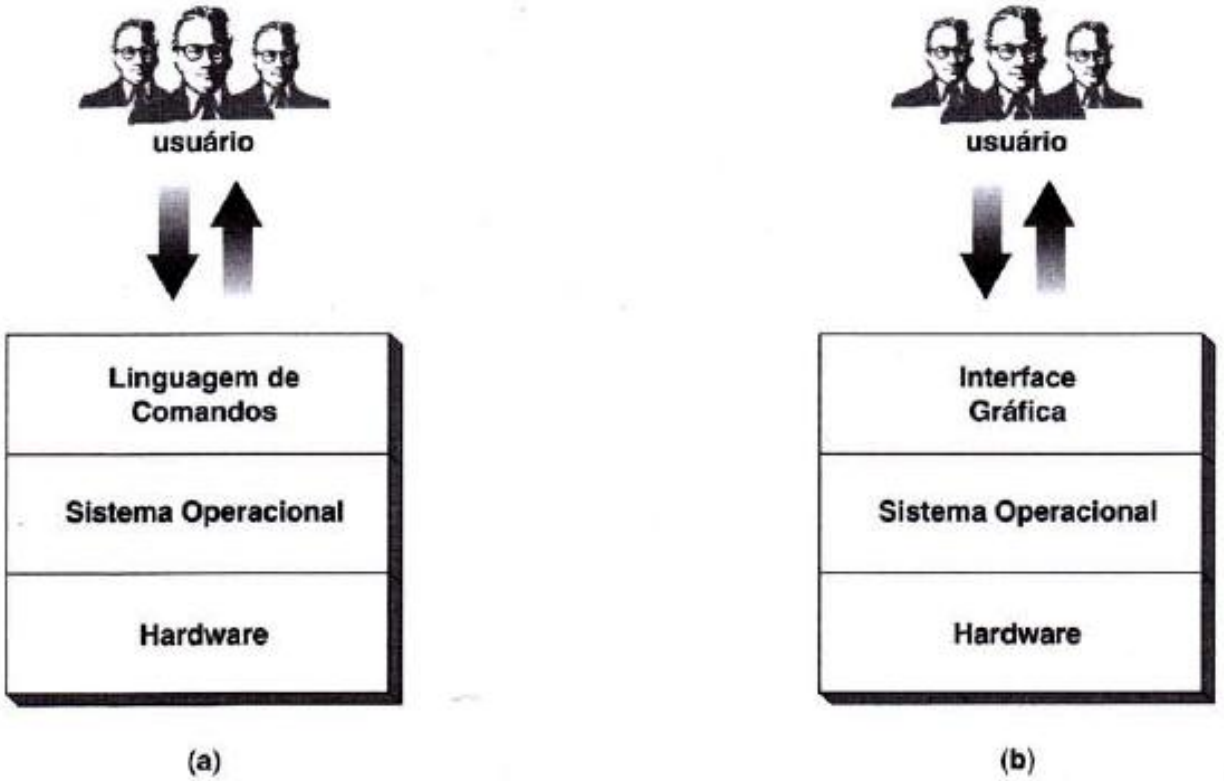
(ou linguagem de controle)

Cada comando, depois de digitado pelo usuário, é interpretado pelo *shell* ou **interpretador de comando** (em geral, não faz parte do SO), que verifica a sintaxe do comando, faz chamadas a rotinas do sistema e apresenta um resultado ou uma mensagem informativa.

No Linux, o *shell* padrão é o bash.

4.6 Linguagem de Comandos

(ou linguagem de controle)



4.6 Linguagem de Comandos

(ou linguagem de controle)

Arquivos *batch* (arquivos de comandos ou *shell scripts*) são programas com uma seqüência de comandos armazenados em um arquivo texto.

Prática: Windows 7

4.7 Ativação/Desativação do Sistema

Ativação do sistema ou *boot*.

1º - *boot loader* (programa na memória ROM)

2º - POST (identifica os possíveis problemas de hardware)

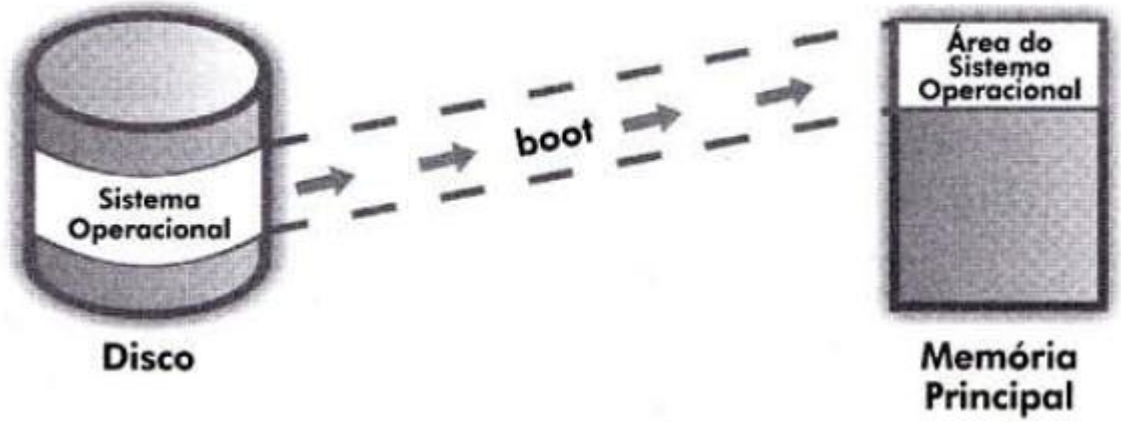
3º - verifica se há um SO residente (erro ou não)

4º - setor de *boot*

5º - carrega o SO para a memória principal

Desativação do sistema ou *shutdown* (desativados ordenadamente).

4.7 Ativação/Desativação do Sistema



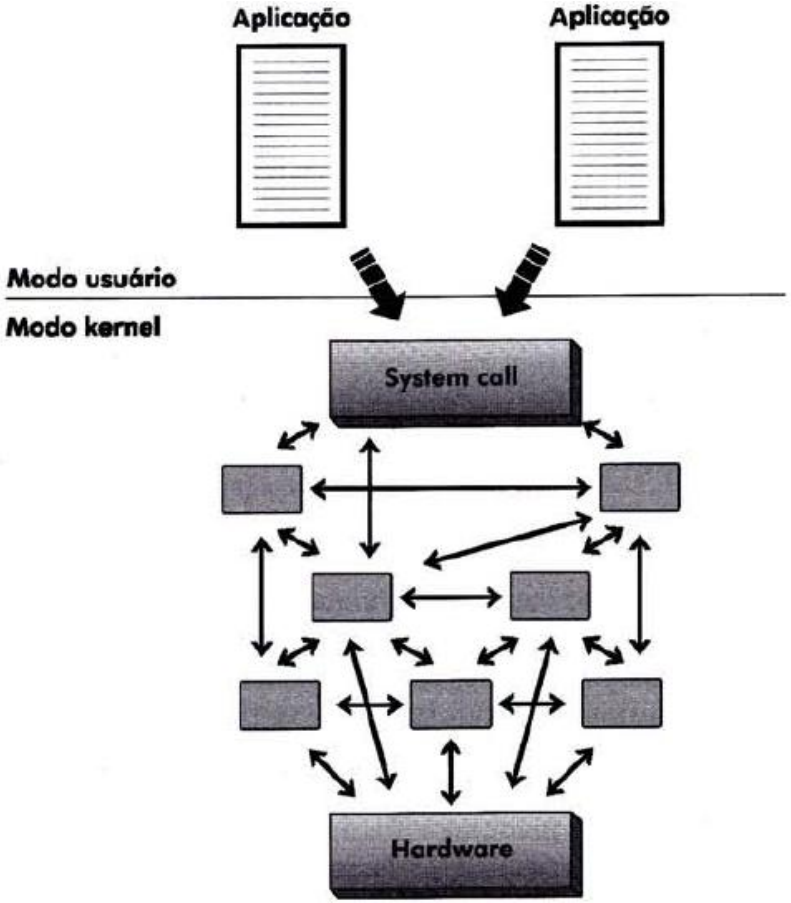
4.8 Arquiteturas do Núcleo

4.8.1 Arquitetura monolítica

- comparada a uma aplicação com vários módulos
- módulos são compilados separadamente e depois linkados
- grande e único programa executável
- desenvolvimento e manutenção difíceis
- bom desempenho
- MS-DOS e nos primeiros sistemas Unix

4.8 Arquiteturas do Núcleo

4.8.1 Arquitetura monolítica



4.8 Arquiteturas do Núcleo

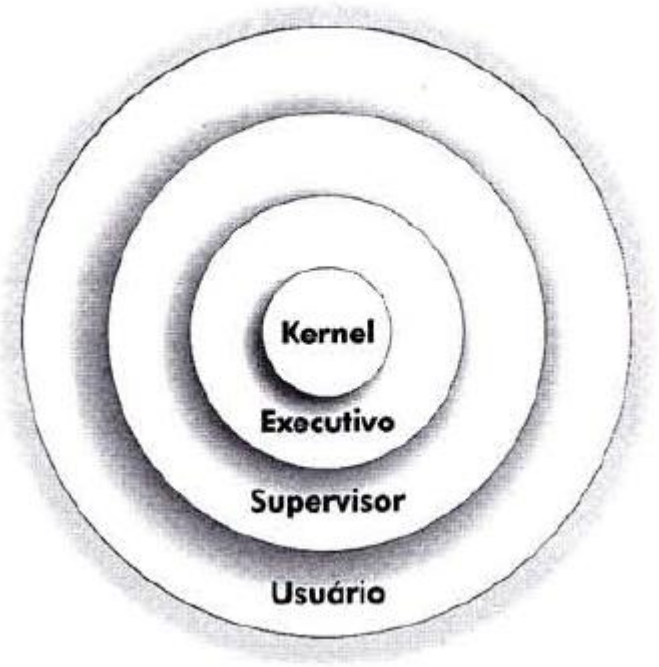
4.8.2 Arquitetura de camadas

O sistema é dividido em níveis (ou camadas) sobrepostos, onde cada camada oferece um conjunto de funções que podem ser utilizadas apenas pelas camadas superiores.

Camadas mais internas são mais privilegiadas que as mais externas.

4.8 Arquiteturas do Núcleo

4.8.2 Arquitetura de camadas



4.8 Arquiteturas do Núcleo

4.8.2 Arquitetura de camadas

Vantagens: isola as funções do SO, facilitando sua manutenção e depuração, protegendo as camadas mais internas.

Desvantagem: desempenho, devido a mudança de modos de acesso entre as camadas.

Exemplo: atualmente o modelo de duas camadas (modo usuário e *kernel*) está sendo utilizado pelo Unix e Windows da Microsoft.

4.8 Arquiteturas do Núcleo

4.8.3 Máquina Virtual

Este modelo cria um nível intermediário entre o hardware e o SO, denominado gerência de máquinas virtuais.

cópia virtual do hardware

cria diversas máquinas virtuais independentes

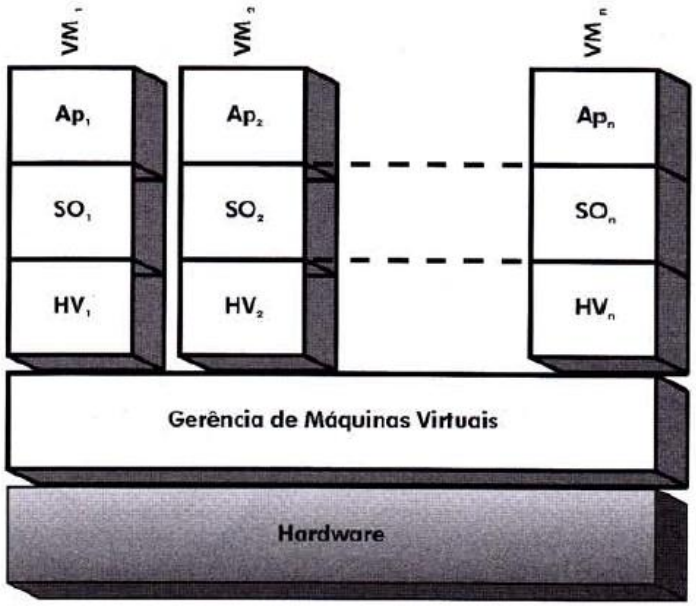


Fig. 4.8 Máquina virtual.

4.8 Arquiteturas do Núcleo

4.8.3 Máquina Virtual

Vantagens: permite a convivência de SOs diferentes no mesmo computador, com segurança para cada máquina virtual.

Desvantagens: grande complexidade devido ao compartilhamento de hardware entre as máquinas virtuais

4.8 Arquiteturas do Núcleo

4.8.4 Arquitetura microkernel

Uma tendência nos SOs modernos é tornar o *kernel* o menor e mais simples possível.

Serviços do sistema são disponibilizados através de processos.

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PIAUÍ

4.8 Arquiteturas do Núcleo

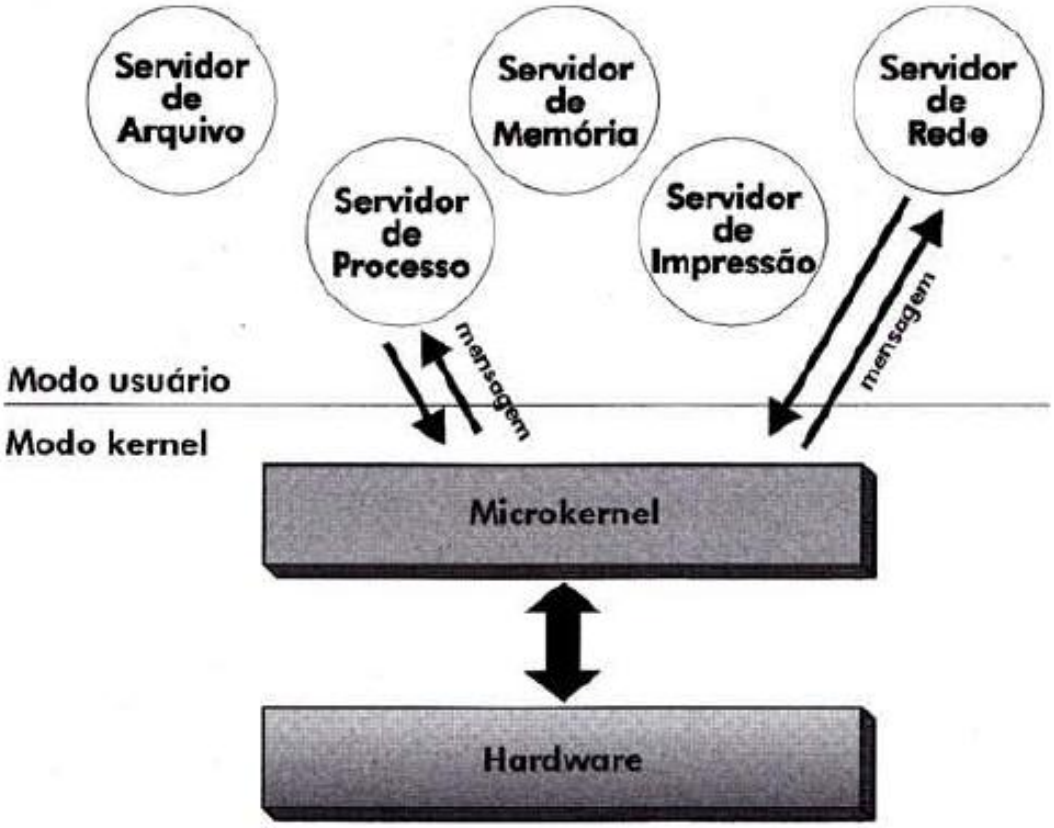
4.8.4 Arquitetura microkernel

Aplicação que solicita - cliente

Processo que responde - servidor

Cliente solicita um serviço enviando uma mensagem para o servidor

Função: realizar a comunicação entre cliente e servidor



4.8 Arquiteturas do Núcleo

4.8.4 Arquitetura microkernel

Surgiu com o SO Mach na década de 80

Servidores (modo usuário)

Microkernel (modo *kernel*)

Aumento da disponibilidade e escalabilidade

Núcleo menor, mais fácil de depurar e mais confiável.

Implementação na prática é difícil.