unex

Arquitetura de Computadores

Introdução aos Sistemas Operacionais

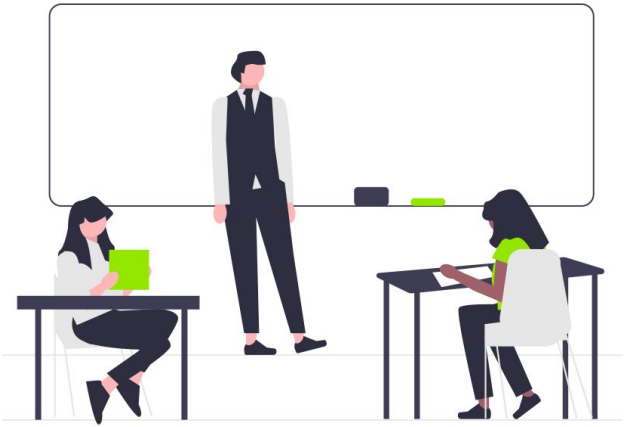
Quem sou eu

Júlio César Andrade

Bacharel em Engenharia de Computação - UEFS
Especialista em User Experience - UNIFACS
Mestrando em Ciências da Computação - UEFS

Objetivo da aula

Capacitar os alunos a compreender os princípios fundamentais dos Sistemas Operacionais.



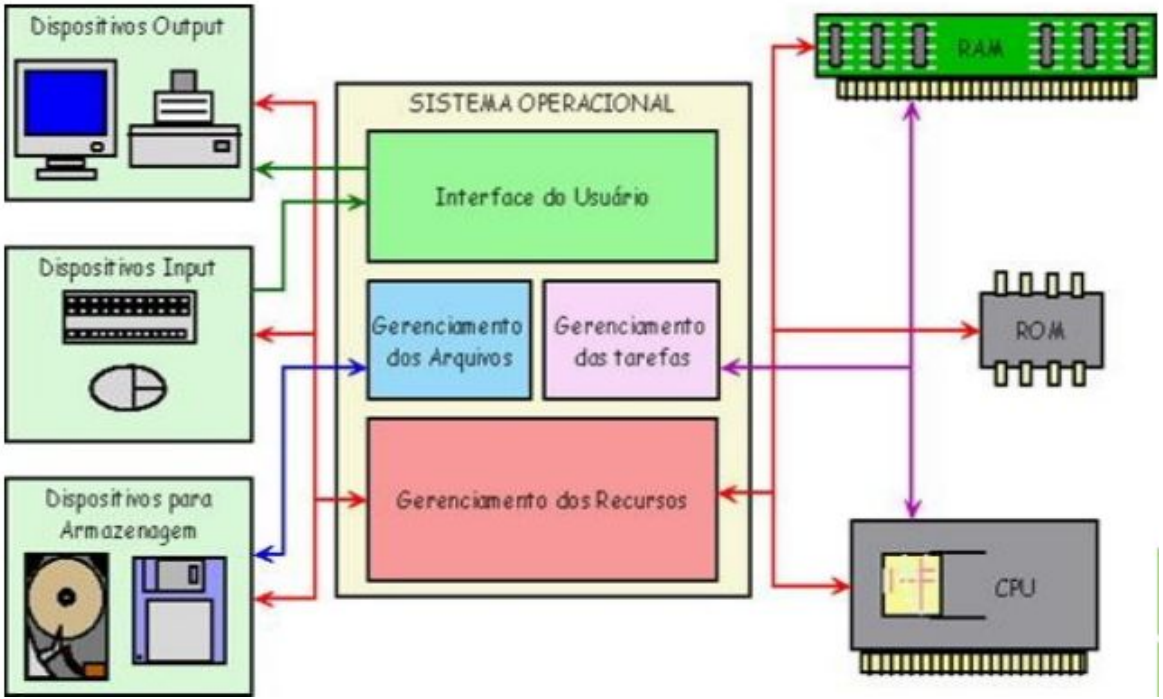
Conceitos básicos de sistemas operacionais

Introdução

Imagine se precisássemos alocar um espaço na memória específico para determinada variável?

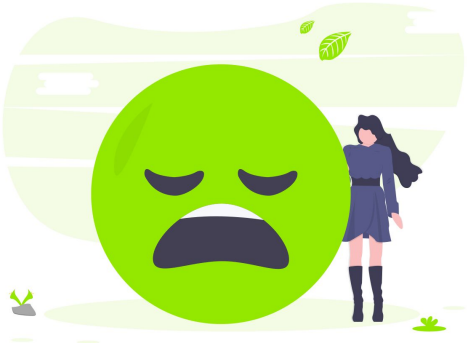
O que é um sistema operacional?

- Programa que realiza a interface entre os softwares aplicativos e o hardware;
- É uma camada de Software que controla o acesso a todos os recursos de hardware e software.


Dispositivos Output
Dispositivos Input
Dispositivos para Armazenagem
SISTEMA OPERACIONAL
Interface do Usuário
Gerenciamento dos Arquivos
Gerenciamento das tarefas
Gerenciamento dos Recursos
RAM
ROM
CPU

Sistema sem S.O.

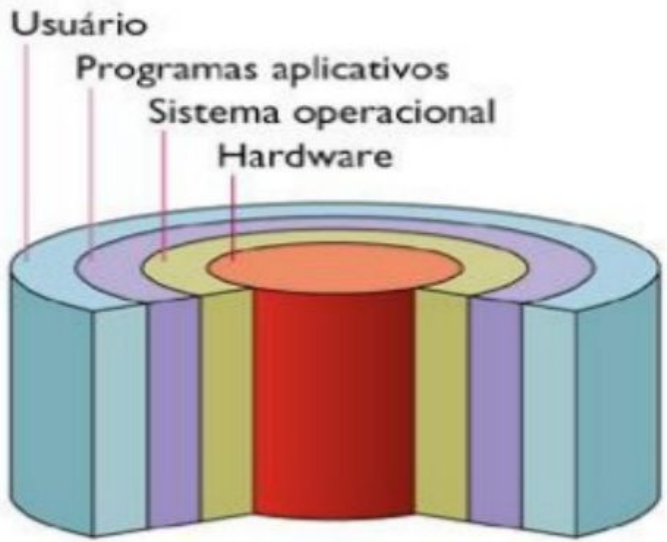
- Gasto maior de tempo de programação;
- Aumento da dificuldade de implementação;
- Usuário preocupado com detalhes de hardware.



Máquina Multinível

Máquina multinível em computação se refere à organização hierárquica de diferentes camadas ou níveis de abstração, como hardware, sistema operacional e aplicativos, que colaboram para realizar operações e processar informações.

Conceito de camadas



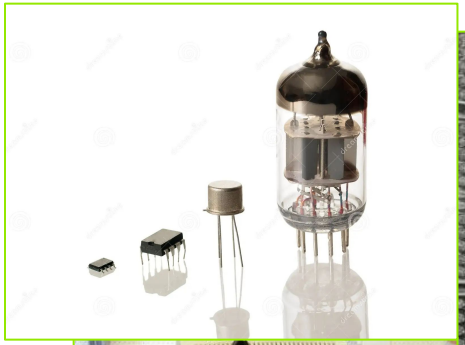
Funções dos sistemas operacionais

Gerenciador de recursos: o trabalho do sistema operacional é gerenciar as diferentes partes do sistema de maneira eficiente.

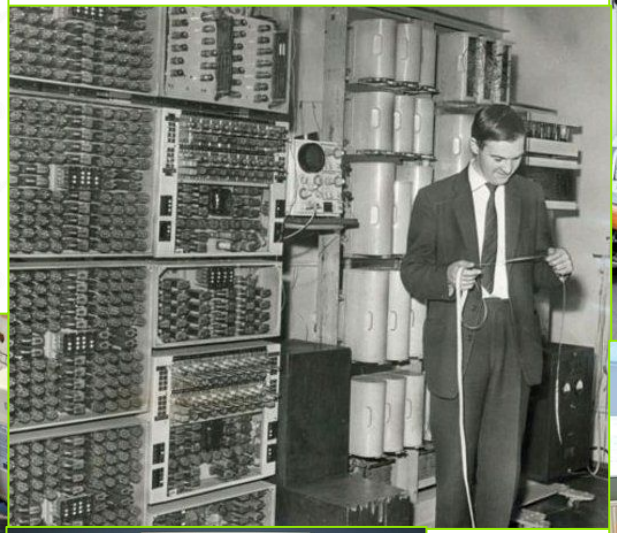
Como máquina estendida: o trabalho do sistema é proporcionar aos usuários abstrações que sejam mais convenientes para usar do que a máquina real. Essas incluem processos, espaços de endereçamento e arquivos.

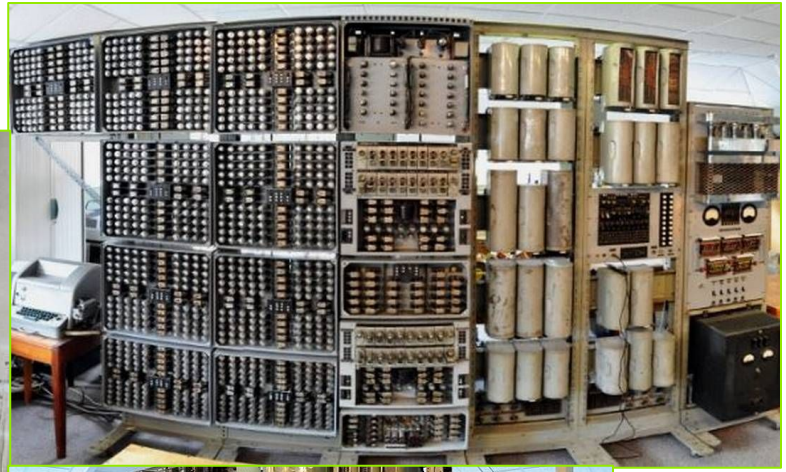
Histórico

Evolução diretamente ligada ao hardware.




WANG





Histórico

1ª Fase (1945-1955)

Características

- Não existia o conceito de SO.
- Máquinas enormes, lentas e imprecisas (milhares de válvulas).
- Necessário profundo conhecimento de hardware.
- Programação em linguagem de máquina através de painéis.

2ª Fase (1956-1965)

Características

- Transistor (velocidade e confiabilidade).
- Memórias magnéticas (maior capacidade, mais compactas).
- Primeiras linguagens (Assembly, Fortran).
- Processamento em batch (maior utilização do processador).
- Rotinas de SO para E/S (IOCS - Input/Output Control System).

Histórico

3ª Fase (1966-1980)

Características

• Circuitos integrados e microprocessadores.
• Diminuição de custos, aumento do poder de processamento.
• Multiprogramação (time-slice - um programa realiza I/O, outro executa).
• Spooling (alteração na ordem de execução das tarefas - não puramente sequencial).
• Time-sharing (tempo compartilhado) programas usam o processador durante pequenos intervalos de tempo.

Histórico

4ª Fase (1981-1990)

Características

- PC's
- Multiprocessamento (mais de uma UCP).
- Processamento Paralelo.
- Redes (heterogeneidade).

5 ª Fase (1991-)

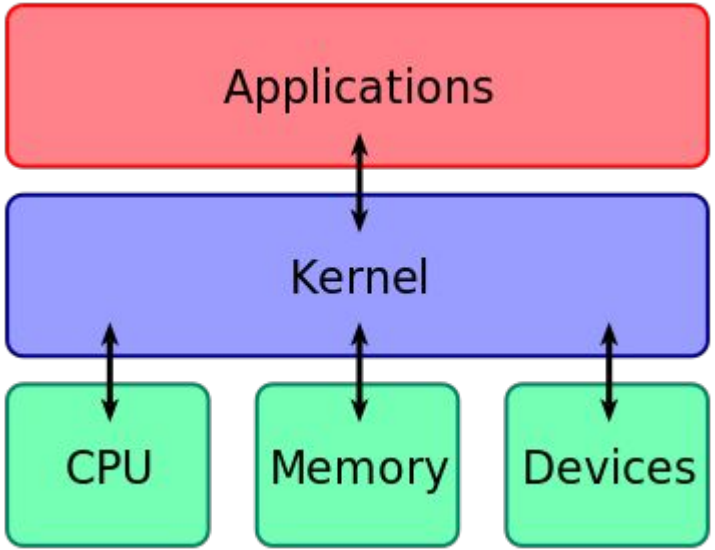
Características

- Maior capacidade de processamento e armazenamento.
- Multimídia, Inteligência Artificial e Bancos de Dados Distribuídos.
- Processamento distribuído (vários processadores e redes).
- Interfaces homem-máquina (linguagem natural, sons e imagens).

Estrutura de uma sistema operacional

Modos de operação

O sistema operacional é dividido em dois modos básicos, sendo estes o modo Kernel e o modo Usuário. Esta divisão permite que os aplicativos sejam executados com segurança.

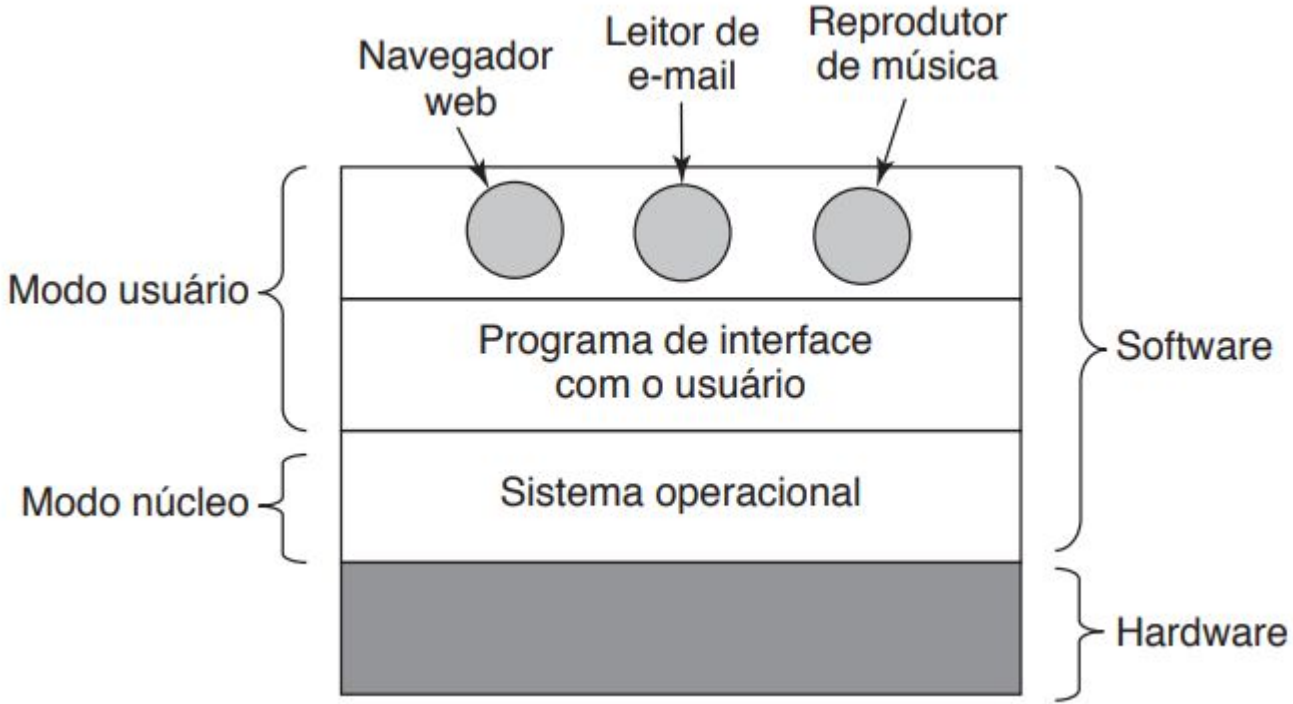


Modo Núcleo (Kernel Mode):

- Privilégios elevados.
- Acesso total aos recursos do hardware.
- Execução de instruções privilegiadas.
- Controle do sistema operacional.
- Gerenciamento direto de hardware e memória.

Modo Usuário (User Mode):

- Privilégios limitados.
- Execução com acesso controlado aos recursos do hardware.
- Não pode executar instruções privilegiadas diretamente.
- Isolamento de aplicativos e processos do usuário.


Navegador web
Leitor de e-mail
Reprodutor de música
Modo usuário
Programa de interface com o usuário
Modo núcleo
Sistema operacional
Software
Hardware

O modo núcleo é onde o núcleo (kernel) do sistema operacional é executado. O kernel gerencia recursos de hardware, como CPU, memória e dispositivos de E/S, e é responsável por tarefas essenciais para o funcionamento do sistema.

Quando um programa precisa realizar operações mais privilegiadas, **ele faz uma chamada ao sistema (system call)**, que transfere a execução para o modo núcleo, onde o kernel pode realizar a operação em nome do programa.

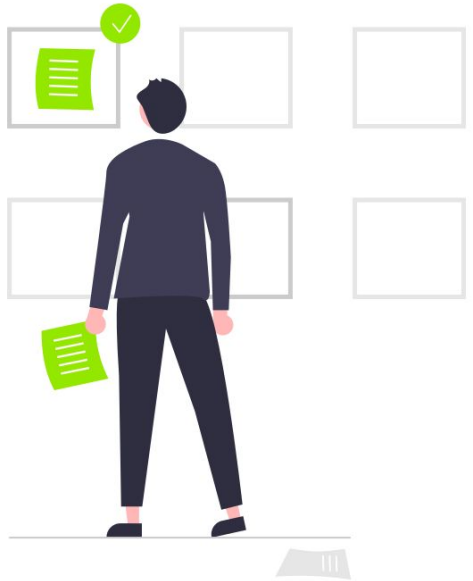
Por que tudo isso?

A separação entre modo núcleo e modo usuário é uma medida de segurança e estabilidade.

Isso **impede que aplicativos normais causem danos inadvertidos ao sistema operacional** ou a outros aplicativos.

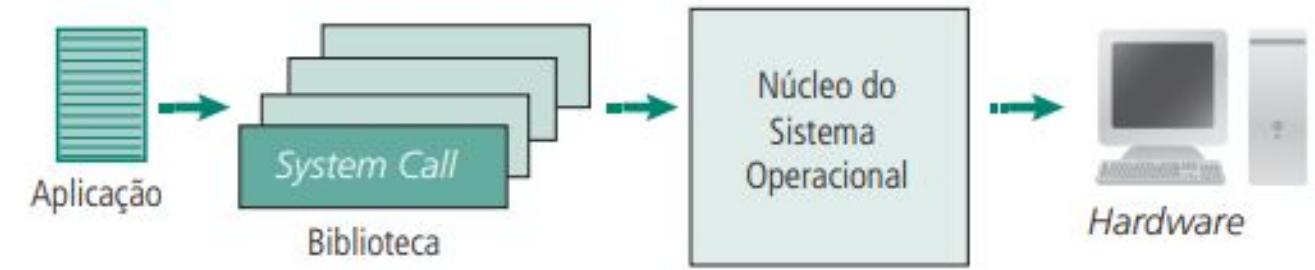
Em resumo…

O modo núcleo é reservado para operações críticas do sistema operacional, enquanto o modo usuário é usado para a execução de aplicativos e processos de usuário comuns.



System calls (Chamadas ao sistema)

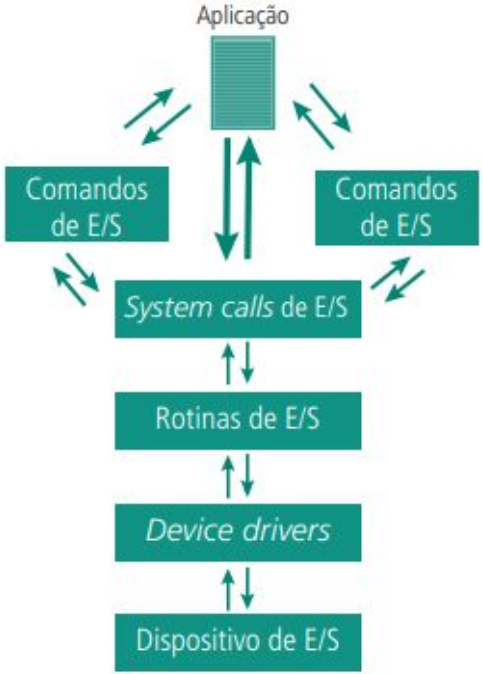
As chamadas ao sistema (system calls) fornecem a interface entre um processo e o sistema operacional. Essas chamadas estão disponíveis como instruções em **linguagem assembly**.



System calls (Chamadas ao sistema)

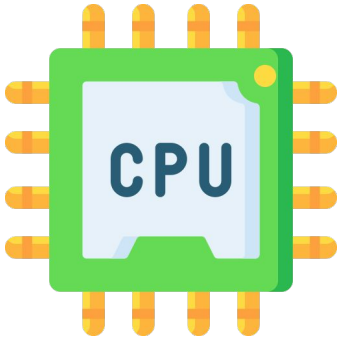
Exemplo:

Consideremos escrever um programa simples para ler dados de um arquivo e copiá-los para outro arquivo. Cada passo dado pelo programa praticamente efetua uma chamada ao sistema.

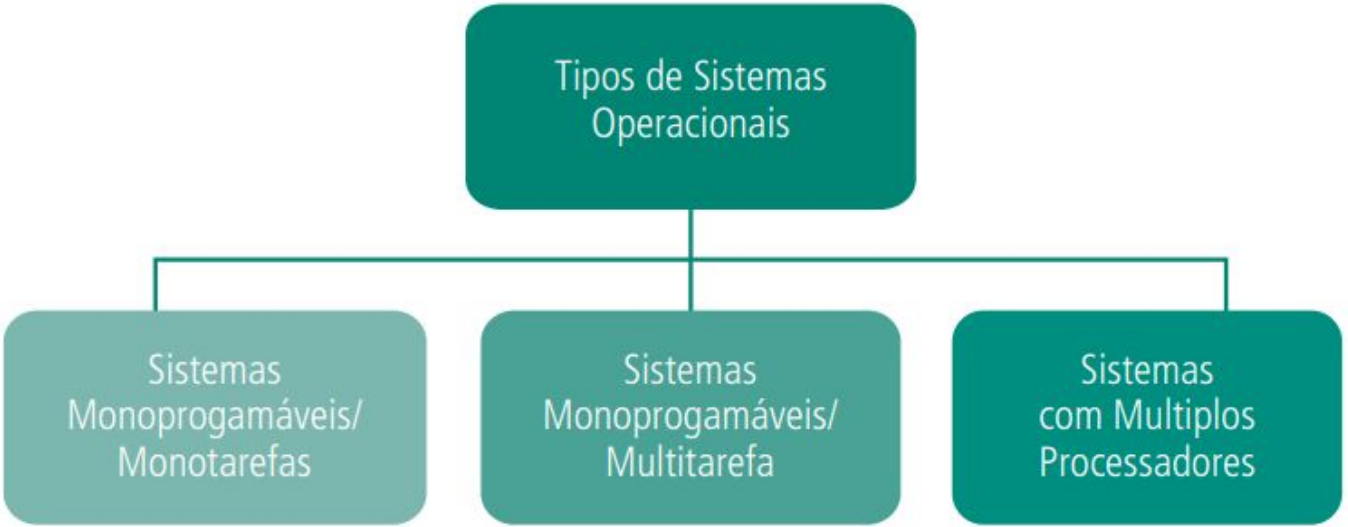


Tipos de Sistemas Operacionais

Os tipos de sistemas operacionais e sua evolução estão relacionados diretamente com a evolução do hardware e das aplicações por ele suportadas.



Considerando o processamento, podemos classificar os sistemas operacionais de acordo com a quantidade de tarefas que podem ser executadas simultaneamente.



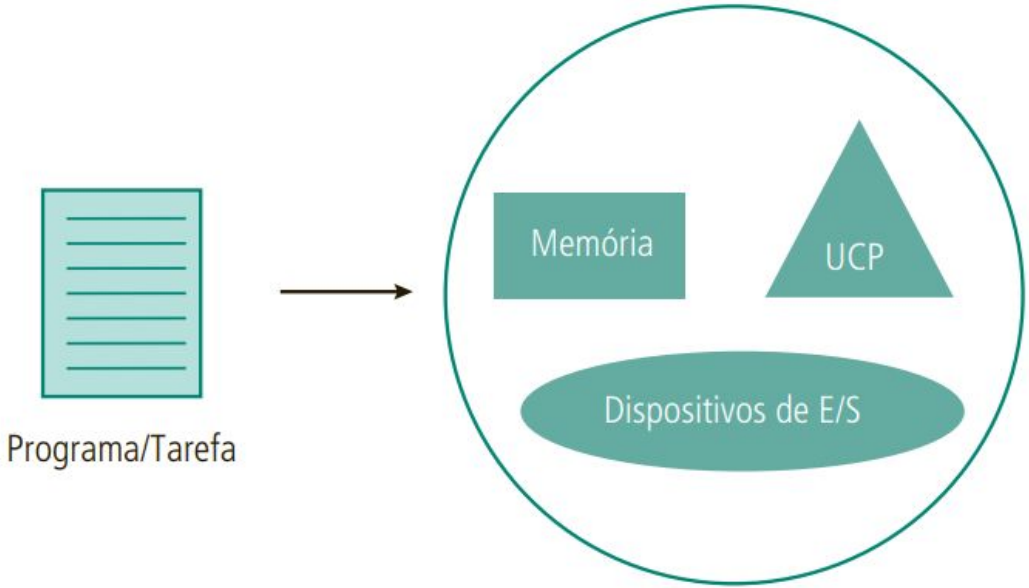
Monoprogramáveis ou Monotarefa

- Dedicado à um único usuário.
- Somente um programa residente na memória.
- Processador ocioso enquanto o programa espera interação do usuário.
- Periféricos e memória, geralmente, subutilizados.
- Implementação relativamente simples (sem preocupações com proteção).

Monoprogramáveis ou Monotarefa

- Dedicado à um único usuário.
- Somente um programa residente na memória.
- Processador ocioso enquanto o programa espera interação do usuário.
- Periféricos e memória, geralmente, subutilizados.
- Implementação relativamente simples (sem preocupações com proteção).

Monoprogramáveis ou Monotarefa



Para que um usuário possa executar outro programa, deverá aguardar a finalização do programa corrente.

Monoprogramáveis ou Monotarefa: Vantagens

Simplicidade:

São mais simples de projetar, implementar e entender, uma vez que só executam um programa por vez.

Menor Overhead:

Há menos sobrecarga de sistema, pois o SO concentra seus recursos em um único programa.

Estabilidade:

Como há apenas um programa em execução, é mais fácil garantir a estabilidade do sistema.

Requisitos de Hardware Menores:

Geralmente, requerem menos recursos de hardware, já que não precisam lidar com a complexidade de multitarefa.

Monoprogramáveis ou Monotarefa: Desvantagens

Baixa Utilização de Recursos:

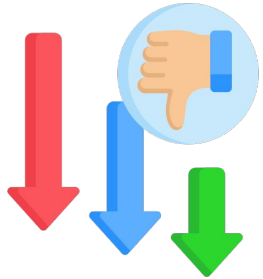
Pode haver períodos de ociosidade do sistema, especialmente quando o programa está aguardando entrada/saída.

Ineficiência:

Em sistemas monoprogramáveis, o processador pode ficar ocioso enquanto aguarda operações de entrada/saída, resultando em eficiência reduzida.

Resposta Lenta:

A resposta do sistema a novas tarefas é geralmente mais lenta, já que é necessário encerrar o programa atual antes de iniciar outro.



Sistemas Multiprogramáveis ou Multitarefa

- Mais complexos e eficientes (divisão de recursos entre usuários).
- Compartilhamento de memória, processador e periféricos (menor custo de utilização).
- Gerenciamento de acesso concorrente aos recursos (ordenado e protegido).
- Suportam mais de um tipo de processamento (batch, time-sharing, real-time).

Sistemas Multiprogramáveis: Vantagens

Alta Utilização de Recursos:

Permitem a execução simultânea de vários programas, maximizando a utilização do processador.

Eficiência:

Reduzem o tempo de resposta ao permitir a comutação rápida entre tarefas.

Concorrência:

Vários programas podem competir por recursos, otimizando o desempenho do sistema.

Adaptabilidade:

São mais adequados para ambientes dinâmicos e complexos.

Sistemas Multiprogramáveis: Desvantagens

Complexidade:

Maior complexidade no design e implementação.

Estabilidade Desafiadora:

Gerenciar múltiplos programas pode tornar a estabilidade mais desafiadora.

Overhead de Troca de Contexto:

O interruptor entre programas pode introduzir overhead de troca de contexto.

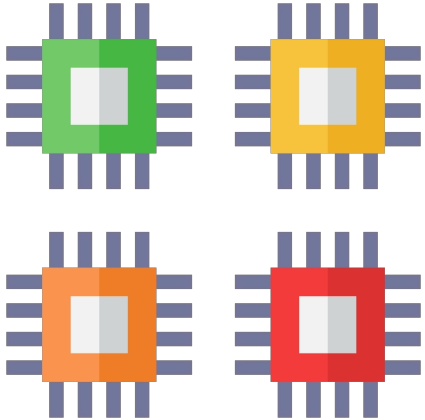
Requerem Hardware Mais Potente:

Podem exigir hardware mais robusto para suportar a execução simultânea de múltiplos programas.

Sistemas com múltiplos processadores

Tarefas distribuídas entre dois ou mais processadores.

Vantagem: permitir que mais de um programa possa ser executado simultaneamente ou que um mesmo programa seja dividido em várias partes e executado simultaneamente nos vários processadores.



Sistemas com múltiplos processadores

Desvantagens:

Complexidade de Programação:

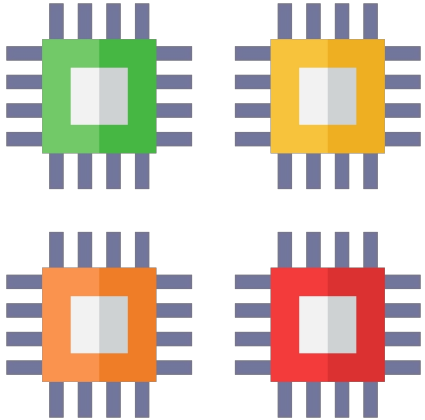
Programar para ambientes multiprocessadores pode ser mais complexo devido à necessidade de gerenciar a concorrência e a sincronização.

Custo:

Hardware e software para sistemas multiprocessadores podem ser mais caros devido à complexidade envolvida.

Sincronização e Concorrência:

Gerenciar a sincronização entre os processadores e lidar com problemas de concorrência pode ser desafiador.



Sistemas com múltiplos processadores

Esse tipo de sistema surgiu da necessidade de aplicações que requeriam um grande poder computacional, como sistemas de previsão do tempo, modelagens, simulações, desenvolvimento aeroespacial, entre outros.

Arquiteturas do núcleo (kernel)

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

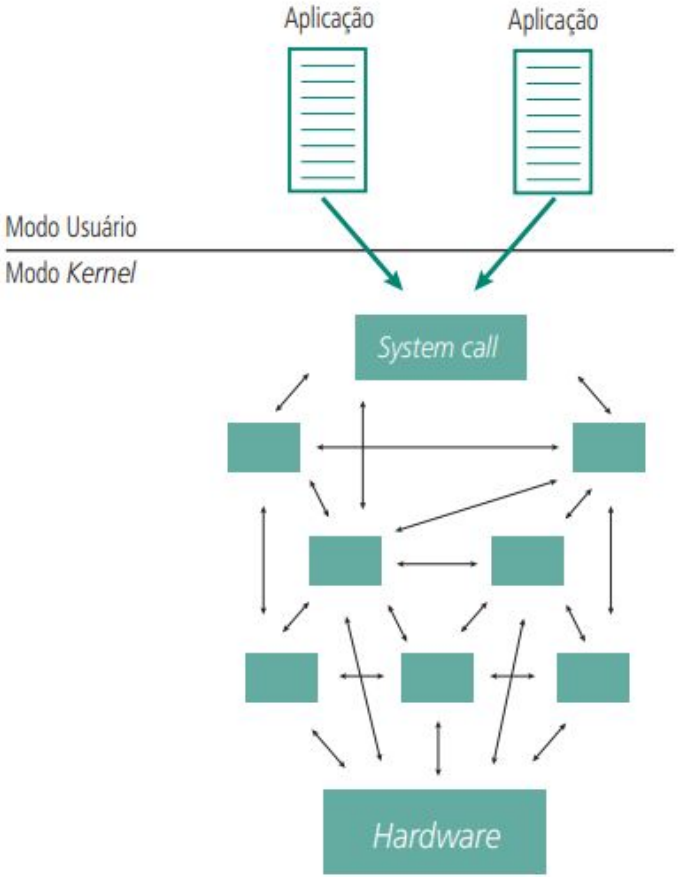
Arquiteturas do núcleo (kernel)

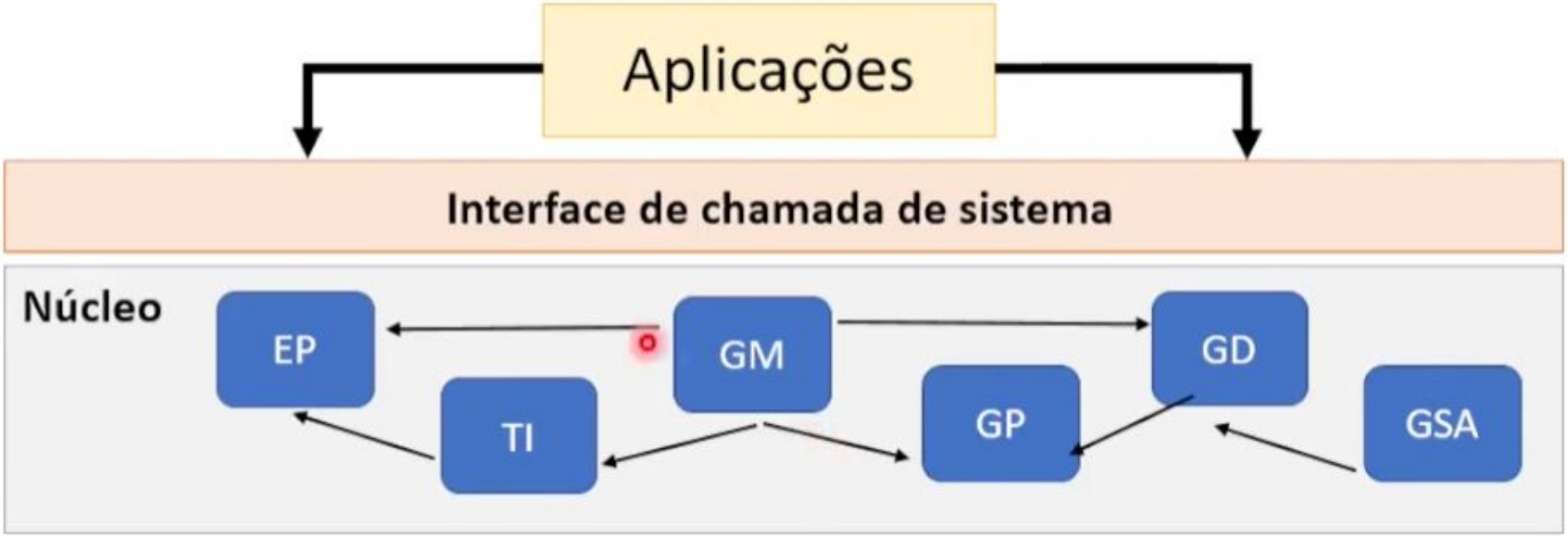
Existem diversos tipos de arquiteturas do kernel:

- Arquitetura monolítica
- Arquitetura de camadas
- Arquitetura de máquina virtual
- Arquitetura microkernel

Arquitetura monolítica

A estrutura monolítica pode ser comparada com uma aplicação formada por vários procedimentos que são compilados separadamente e depois linkados, formando um grande e único programa executável.




Aplicações
Interface de chamada de sistema
Núcleo
EP
TI
GM
GP
GD
GSA

Alguns exemplos de SOs com arquitetura monolítica:

MS-DOS: O Microsoft Disk Operating System, embora antigo, é um exemplo clássico de um sistema operacional monolítico. Ele era usado em computadores pessoais durante a década de 1980.

Linux (em sua maioria): Embora o kernel Linux seja monolítico, é importante notar que o Linux tem uma modularidade considerável, permitindo que você adicione ou remova módulos de kernel conforme necessário.

Unix (em alguns casos): Alguns sistemas Unix também adotam uma abordagem monolítica, embora muitos Unix modernos tenham características de sistemas híbridos ou camadas.

Windows (até certo ponto): O Windows é geralmente considerado um sistema operacional monolítico, embora tenha componentes distintos que desempenham funções específicas.

Arquitetura em Camadas

A arquitetura em camadas consiste em níveis sobrepostos, onde cada camada tem um conjunto de funções que podem ser utilizadas pelas camadas superiores.

Arquitetura em Camadas

Estrutura do sistema operacional THE. (1968)

Camada	Função
5	O operador
4	Programas de usuário
3	Gerenciamento de entrada/saída
2	Comunicação operador–processo
1	Memória e gerenciamento de tambor
0	Alocação do processador e multiprogramação

NOTA: Como mencionado anteriormente, Windows não segue uma arquitetura estritamente em camadas da mesma forma que alguns sistemas Unix/Linux, mas possui componentes distintos que operam em diferentes níveis.

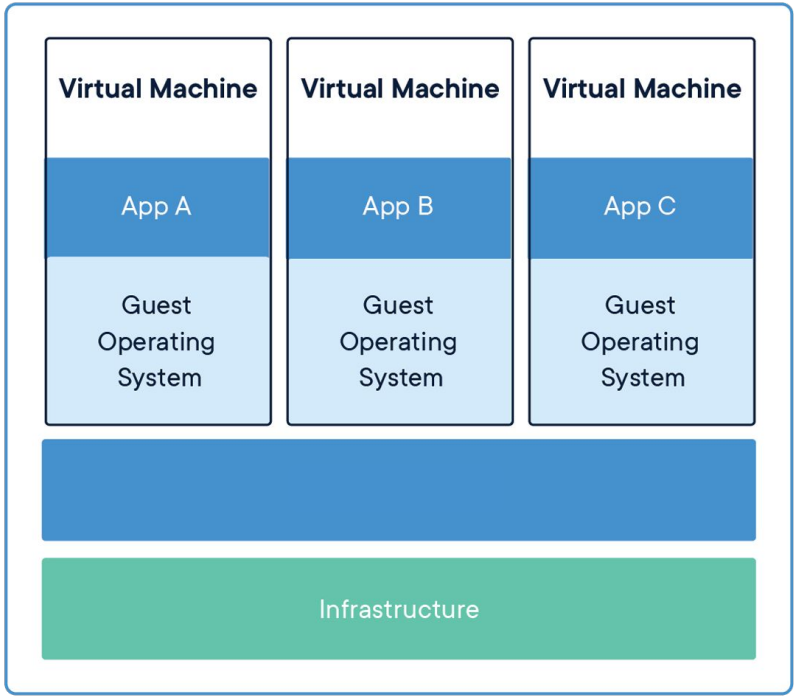
Arquitetura em Camadas

Vantagem: Segurança e proteção às camadas mais internas.

Desvantagem: Pode haver uma pequena sobrecarga de desempenho devido à comunicação entre as camadas.

Arquitetura de máquina virtual

Na arquitetura de máquina virtual cria-se um nível intermediário entre hardware e sistema operacional, que cria diversas máquina virtuais independentes com uma cópia virtual do hardware.



Arquitetura de máquina virtual

- Portabilidade de código
- Consolidação de servidores (executar diversas aplicações em uma única máquina ao invés de diversos servidores separados)
- Aumento da disponibilidade (qualquer problema com uma VM basta restaurar a cópia da VM em outro servidor)
- Facilidade de escalabilidade e balanceamento de carga (se o sistema ficar sobrecarregado a VM pode ser migrada para outro ambiente com facilidade)
- Facilidade no desenvolvimento de software (mais fácil de testar software em dois sistemas operacionais diferentes)

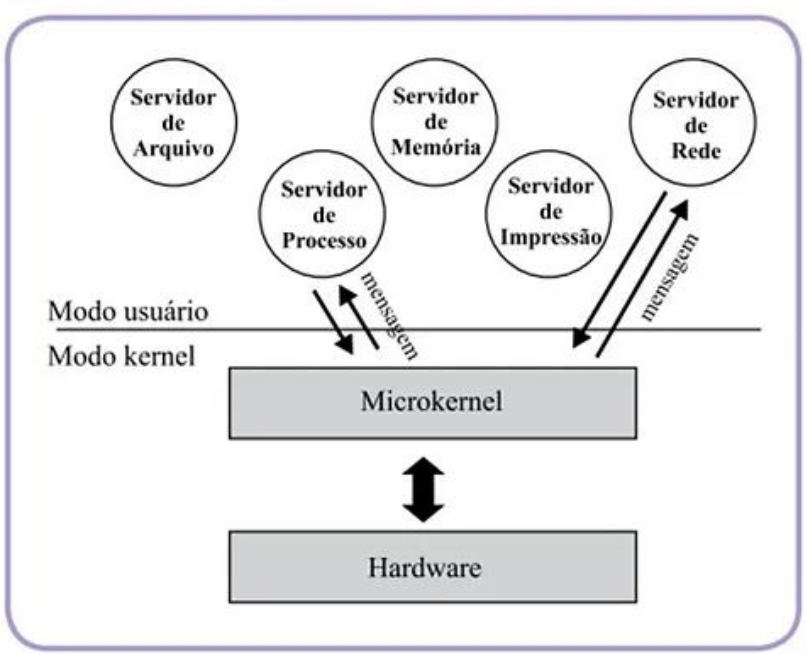
Arquitetura microkernel

Mantém o núcleo mínimo (microkernel) com funcionalidades essenciais e move serviços mais complexos para fora do kernel.

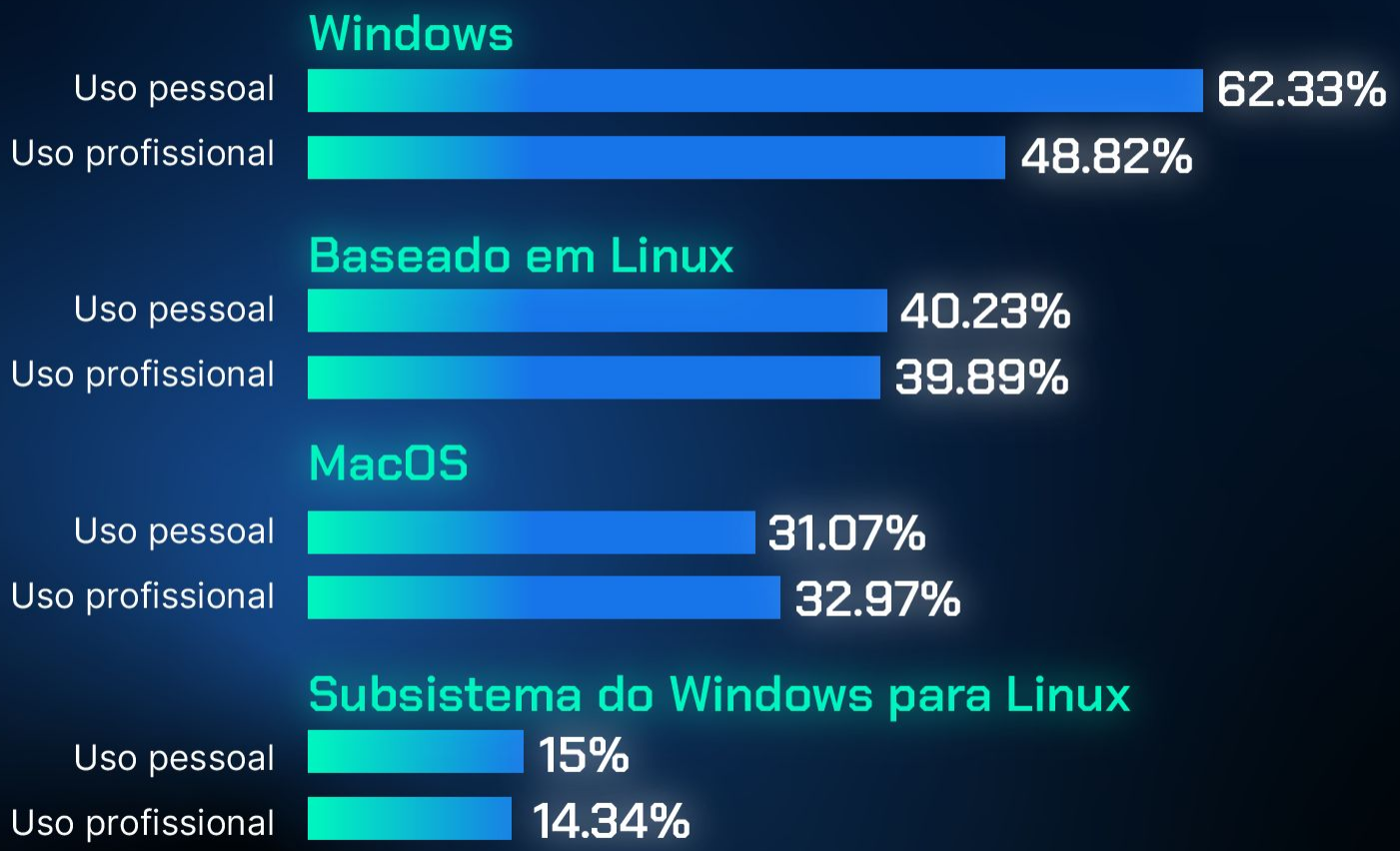
Características:

- Flexibilidade e extensibilidade, facilitando a adição/remoção de serviços.
- Menor complexidade no núcleo, o que pode resultar em maior estabilidade.

Arquitetura microkernel



Quais os sistemas operacionais mais usados?


Windows
Uso pessoal
62.33%
Uso profissional
48.82%
Baseado em Linux
Uso pessoal
40.23%
Uso profissional
39.89%
MacOS
Uso pessoal
31.07%
Uso profissional
32.97%
Subsistema do Windows para Linux
Uso pessoal
15%
Uso profissional
14.34%

Bibliografia

TANENBAUM, A. S. , **Sistemas Operacionais Modernos.** Segunda Edição, Prentice Hall, 2003.
Bibliografias Complementares. GALVIN, S., Operating System Concepts.