

---

# EMB22109 - Sistemas Embarcados:

## Introdução a Sistemas Operacionais e Tipos de Kernel

Hugo Marcondes

Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina  
Departamento Acadêmico de Eletrônica  
hugo.marcondes@ifsc.edu.br

28 de Maio de 2025

*\* Créditos ao Prof. Emerson Ribeiro de Mello, o qual criou e disponibilizou o template aqui usado, via ShareLaTeX*



# Na aula de hoje veremos...

---

**1** Introdução a Sistemas Operacionais

**2** Tipos de Kernel

**3** Referências

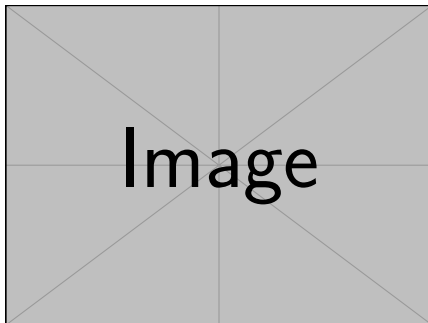


# Introdução a Sistemas Operacionais

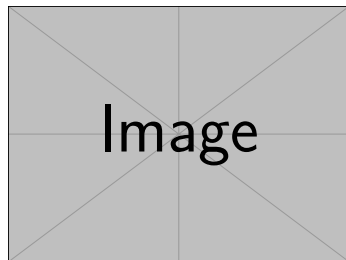


## Definição

**Sistema Operacional** é uma camada de software que gerencia os recursos de hardware e fornece serviços para programas de aplicação.



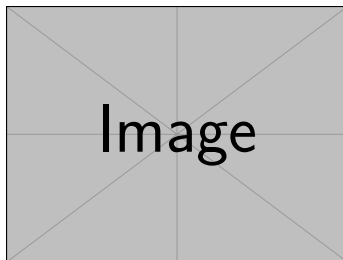
- 1 Visam FACILITAR o uso de sistemas computacionais
- 2 Sistemas Computacionais
  - 2.1 CPU + Memória + I/O
- 3 O sistema operacional pode ser visto sob duas perspectivas:
  - 3.1 Máquina Virtual
  - 3.2 Gerenciador de Recursos



# SO como Máquina Virtual

---

- 1 O sistema operacional "estende" o hardware para implementar uma interface de mais alto nível para as aplicações
- 2 Importante característica para prover portabilidade
- 3 Abstrai detalhes complexos do hardware
- 4 Fornece uma interface uniforme para programas de aplicação



# SO como Gerenciador de Recursos

---

## 1 Gerencia recursos de hardware:

1.1 CPU

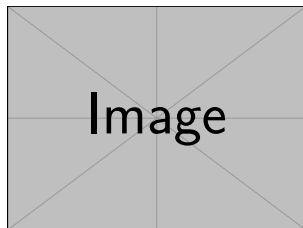
1.2 Memória

1.3 Dispositivos de I/O

## 2 Controla a execução de programas

## 3 Previne erros e uso indevido

## 4 Garante o uso eficiente dos recursos



## 1 Kernel = Núcleo

1.1 Parte central e fundamental de um programa/algoritmo

## 2 Kernel = Modo Kernel (Supervisor)

2.1 Parte de um programa que executa em modo Kernel

2.2 Modo Kernel -> Espaço/Modo de Execução

2.3 Suporte do Processador a diferentes espaços ou níveis de execução (modo Kernel e modo usuário)



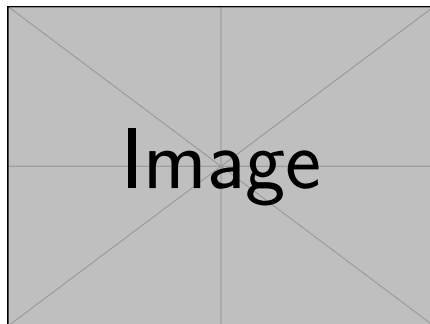


# Kernel de um Sistema Operacional

---

- 1 Núcleo do Sistema Operacional
- 2 Provê o Gerenciamento do Sistema e Funcionalidades Básicas
- 3 Executa em modo privilegiado (modo kernel)
- 4 Controla acesso direto ao hardware
- 5 Implementa serviços fundamentais:
  - 5.1 Gerenciamento de memória
  - 5.2 Gerenciamento de processos
  - 5.3 Gerenciamento de dispositivos
  - 5.4 Comunicação entre processos





- 1 Diagramas de camadas de Abstração e Anéis de Privilégio de Execução
- 2 Diferentes níveis de acesso ao hardware
- 3 Proteção contra operações indevidas

# Tipos de Kernel



## 1 Kernel Monolítico

## 2 Microkernel

### 2.1 Nanokernel

## 3 Kernel Híbrido

## 4 Exokernel

*\*Kernel Híbrido é tido como um termo controverso dentre a comunidade científica da área.*



# Kernel Monolítico

---

1 Todo o sistema operacional executa em modo kernel

2 Abordagem tradicional

3 Exemplos:

3.1 Unix tradicional

3.2 Linux

3.3 FreeBSD

4 Vantagens:

4.1 Desempenho

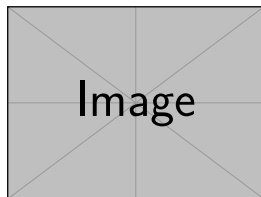
4.2 Acesso direto ao hardware

5 Desvantagens:

5.1 Menor segurança

5.2 Falhas afetam todo o sistema

5.3 Código mais complexo



# Microkernel

---

- 1 Apenas funções essenciais no kernel
- 2 Serviços executam como processos em modo usuário
- 3 Comunicação via IPC (Comunicação Entre Processos)

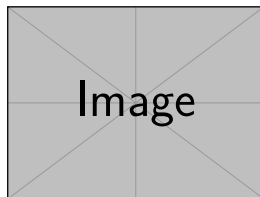
- 4 Exemplos:

- 4.1 MINIX
  - 4.2 QNX
  - 4.3 L4

- 5 Vantagens:

- 5.1 Maior segurança
  - 5.2 Falhas isoladas
  - 5.3 Código mais modular

- 6 Desvantagens:



1

## Segurança

1.1

Menor código possível executando em modo Kernel

1.2

Menos tempo de execução do sistema em modo privilegiado

2

## Separação de Serviços em Servidores

2.1

Serviços executam como processos independentes

2.2

Isolamento de falhas

3

## Comunicação via IPC

3.1

Baseada em troca de mensagens

3.2

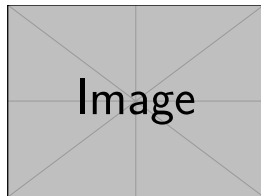
Mecanismo fundamental para operação do sistema



# Kernel Híbrido

---

- 1 Combina características de kernel monolítico e microkernel
- 2 Alguns serviços no espaço do kernel, outros como processos de usuário
- 3 Exemplos:
  - 3.1 Windows NT (Windows 10/11)
  - 3.2 macOS (XNU)
  - 3.3 DragonFly BSD
- 4 Vantagens:
  - 4.1 Equilíbrio entre desempenho e modularidade
  - 4.2 Flexibilidade de design
- 5 Desvantagens:





- 1 Abordagem minimalista e radical
- 2 Kernel fornece apenas proteção e multiplexação de recursos
- 3 Bibliotecas em nível de usuário (LibOS) implementam abstrações tradicionais
- 4 Desenvolvido no MIT nos anos 90
- 5 Vantagens:
  - 5.1 Máxima flexibilidade para aplicações
  - 5.2 Personalização de abstrações
  - 5.3 Potencial para alto desempenho
- 6 Desvantagens:
  - 6.1 Complexidade para desenvolvedores de aplicações
  - 6.2 Menor portabilidade
  - 6.3 Poucos sistemas em produção



## Comparação entre Tipos de Kernel

---

<b>Característica</b>	<b>Monolítico</b>	<b>Microkernel</b>	<b>Híbrido</b>	<b>Exokernel</b>
Tamanho do kernel	Grande	Pequeno	Médio	Mínimo
Desempenho	Alto	Moderado	Alto/Moderado	Potencialmente Alto
Modularidade	Baixa	Alta	Média	Alta
Segurança	Menor	Maior	Média	Varia
Complexidade	Alta	Média	Alta	Baixa (para o usuário) Alta (para o kernel)
Exemplos	Linux, FreeBSD	MINIX, QNX	Windows NT, macOS	MIT Exokernel



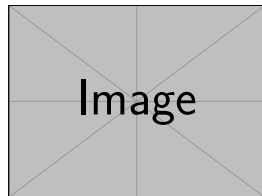
# Interação de um Sistema com SO

---

## 1 Comunicação entre aplicações e kernel:

1.1 Kernel Monolítico:  
System Calls e Sinais de Interrupções

1.2 Microkernel: IPC,  
Kernel Calls e Sinais de Interrupções



## 2 Diferenças fundamentais:

2.1 Monolítico: chamadas  
diretas ao kernel

2.2 Microkernel:  
comunicação via  
mensagens entre  
processos



# Tendências Atuais em Design de Kernel

---

- 1 Sistemas operacionais modernos tendem a adotar abordagens híbridas
- 2 Linux: kernel monolítico com módulos carregáveis
- 3 Windows: kernel híbrido com componentes em modo kernel e modo usuário
- 4 Sistemas embarcados: microkernels ganham popularidade pela segurança
- 5 Virtualização e containers: novas demandas para design de kernel
- 6 Sistemas de tempo real: foco em previsibilidade e determinismo



# Resumo

---

- 1 Sistemas Operacionais são camadas de software que gerenciam recursos de hardware
- 2 Podem ser vistos como Máquinas Virtuais ou Gerenciadores de Recursos
- 3 O Kernel é o núcleo do sistema operacional, executando em modo privilegiado
- 4 Principais tipos de kernel:
  - 4.1 Monolítico: todo o SO em modo kernel (Linux)
  - 4.2 Microkernel: mínimo em modo kernel, serviços em modo usuário (MINIX)
  - 4.3 Híbrido: combinação das abordagens anteriores (Windows NT)
  - 4.4 Exokernel: apenas proteção e multiplexação de recursos
- 5 Cada abordagem apresenta vantagens e desvantagens em termos de desempenho, segurança e modularidade



# Referências



# References I

---

