



O que é um Kernel?

O **Kernel** é o núcleo central de um sistema operacional, que gerencia a comunicação entre o hardware e o software.

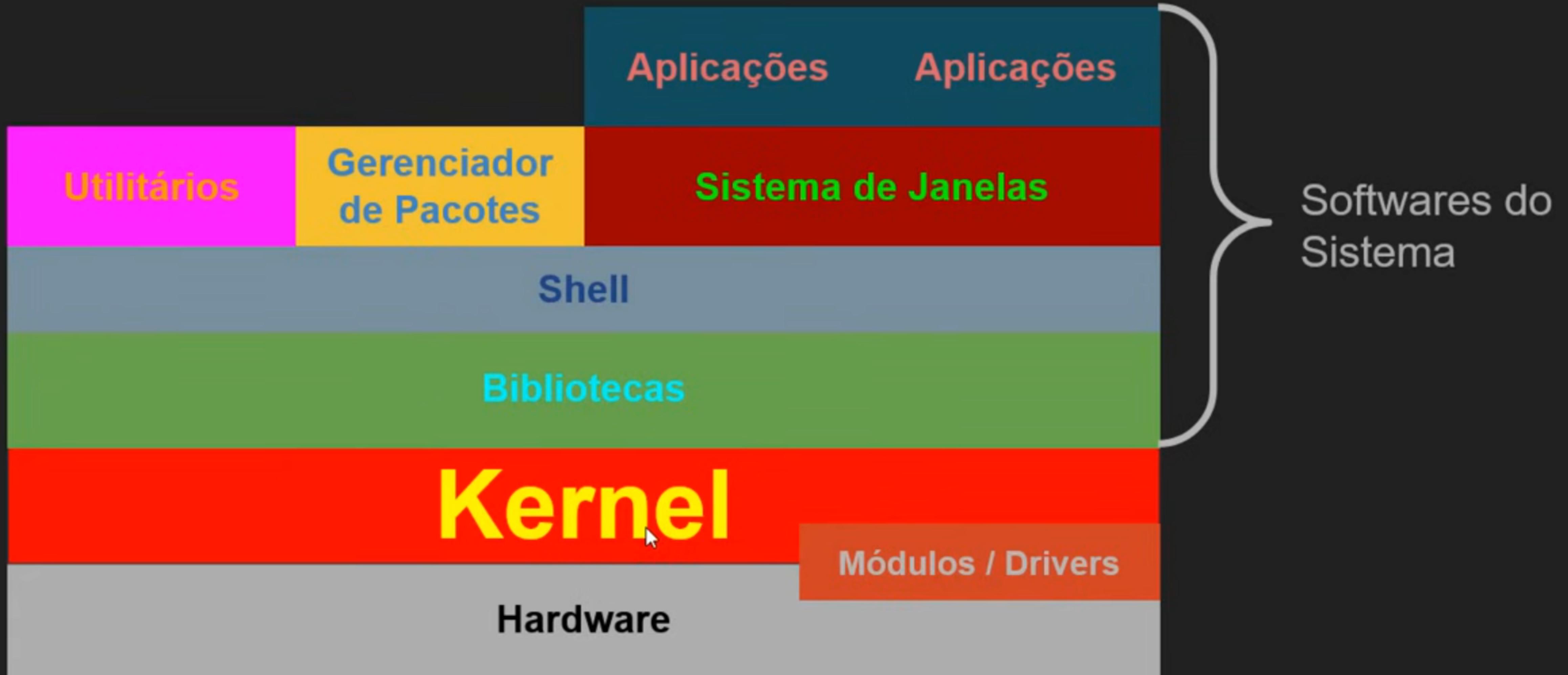
Atua como software intermediário, permitindo que programas utilizem os recursos do sistema de forma eficiente e segura.

Suas funções incluem (entre outras):

- Gerenciamento de recursos: CPU, memória e dispositivos de entrada/saída.
- Comunicação interna: Facilita chamadas de sistema e a execução dos processos.

Sem um kernel, os aplicativos não teriam como interagir com o hardware, e o sistema não conseguiria gerenciar os recursos de forma adequada.

Modelo de um Sistema Linux com seu Kernel



Principais Funções de um Kernel



Principais Funções de um Kernel



- **Gerenciamento de Processos:** Criação, execução, escalonamento.

Principais Funções de um Kernel



- **Gerenciamento de Processos:** Criação, execução, escalonamento.
- **Gerenciamento de Memória:** Controla a alocação dinâmica de memória, prevenindo conflitos entre processos.
- **Gerenciamento de Dispositivos:** Comunicação com os dispositivos de hardware por meio de drivers, como por exemplo periféricos (mouse, teclado, etc.).

Principais Funções de um Kernel



- **Gerenciamento de Processos:** Criação, execução, escalonamento.
- **Gerenciamento de Memória:** Controla a alocação dinâmica de memória, prevenindo conflitos entre processos.
- **Gerenciamento de Dispositivos:** Comunicação com os dispositivos de hardware por meio de drivers, como por exemplo periféricos (mouse, teclado, etc.).
- **Gerenciamento de Arquivos:** Manipulação de arquivos, controle de acesso e integridade.



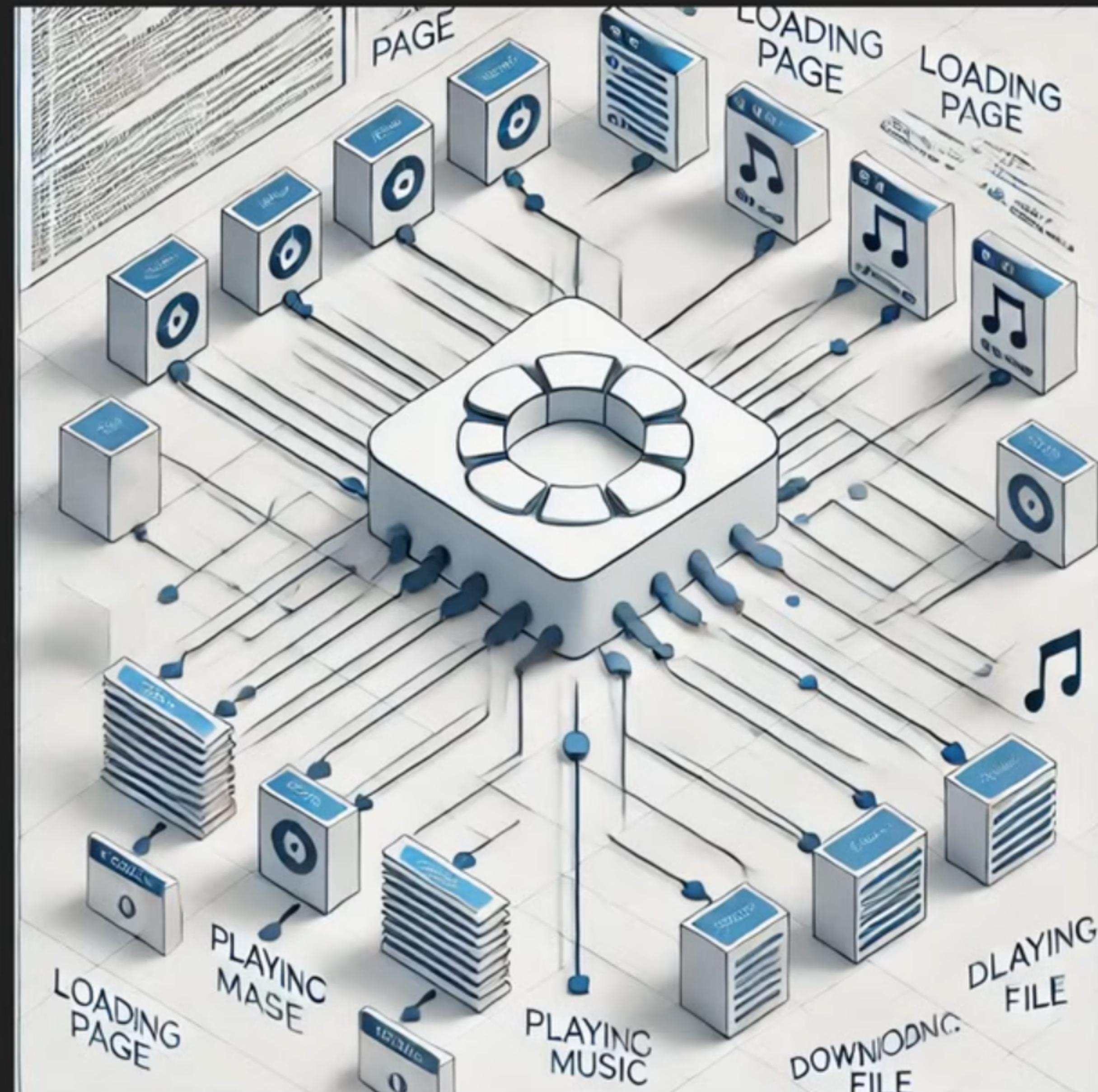
Principais Funções de um Kernel

- **Gerenciamento de Processos:** Criação, execução, escalonamento.
- **Gerenciamento de Memória:** Controla a alocação dinâmica de memória, prevenindo conflitos entre processos.
- **Gerenciamento de Dispositivos:** Comunicação com os dispositivos de hardware por meio de drivers, como por exemplo periféricos (mouse, teclado, etc.).
- **Gerenciamento de Arquivos:** Manipulação de arquivos, controle de acesso e integridade.
- **Gerenciamento de Segurança:** Proteção: isolamento de processos e privilégios de acesso.
- **Chamadas de Sistema:** Solicitação de serviços do SO e interface com os programas de usuário.

Processos e Threads

- **Processo:** Programa em execução (código, dados, recursos).
- **Thread:** Menor unidade de execução dentro de um processo (tarefas).
- Múltiplos threads em um processo = execução simultânea (paralelismo).
- *Exemplo:* Threads de downloads múltiplos em um navegador.

Processo com threads diferentes rodando simultaneamente (carregar página, tocar música, baixar arquivo).



Arquiteturas de Kernel

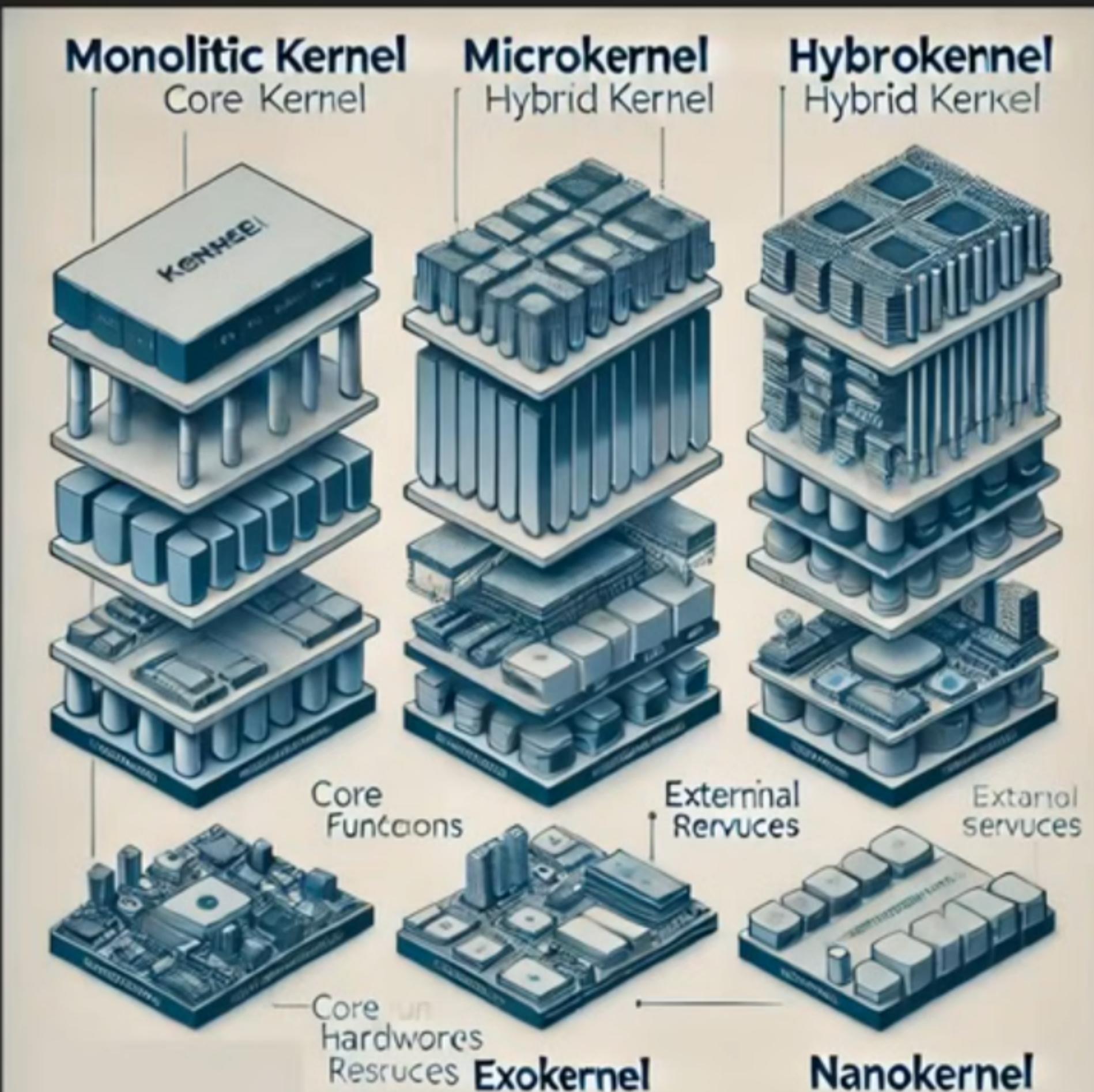


Tipos de Kernel: Uma Visão Geral

Existem diferentes abordagens para a arquitetura de núcleos.

As mais comuns são:

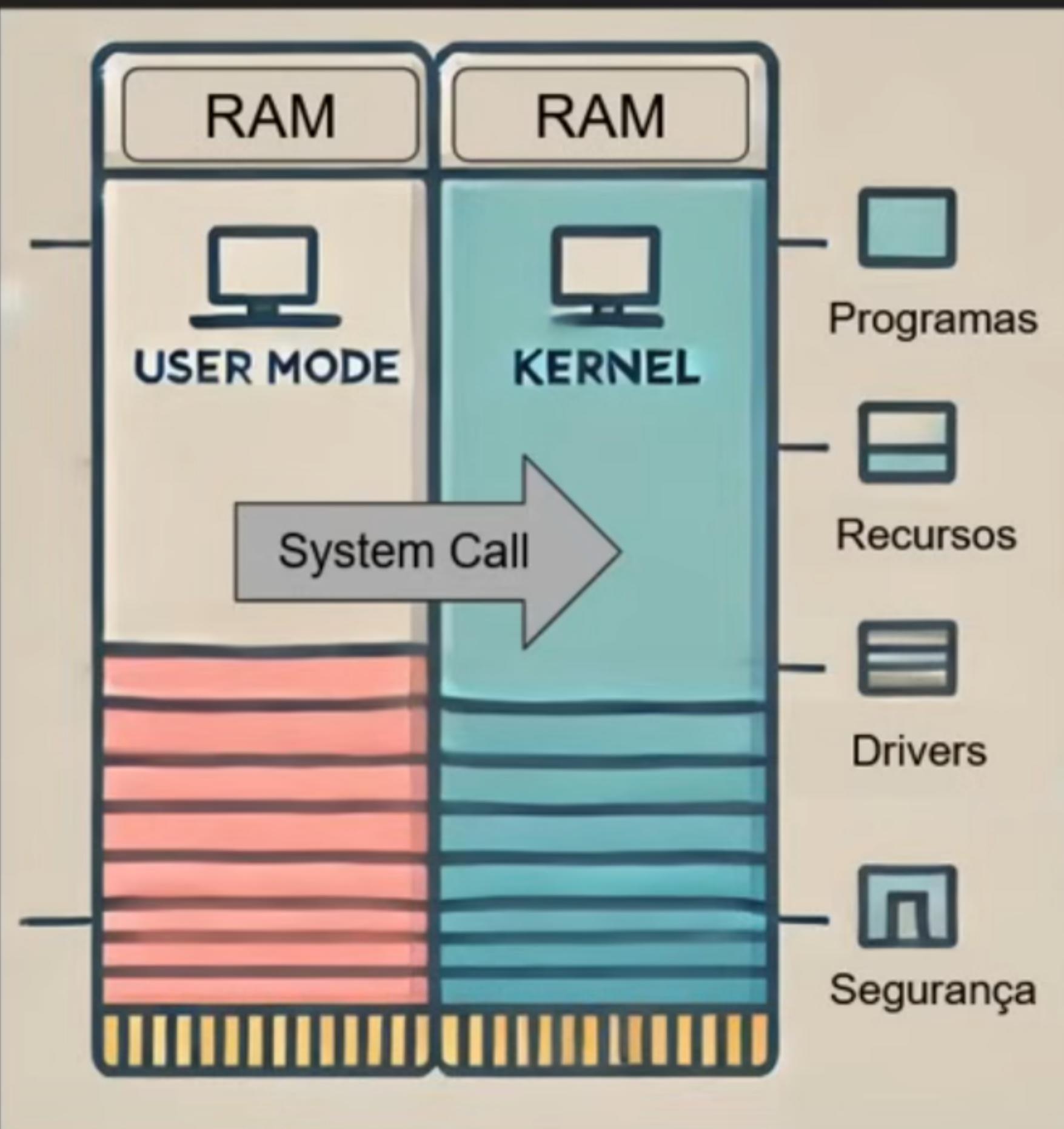
- Kernel Monolítico
- Microkernel
- Kernel Híbrido
- Exokernel
- Nanokernel



Modos de Operação: Modo Kernel vs. Modo Usuário

Modos de Operação são os níveis de privilégio com os quais o sistema executa instruções, divididos em dois principais:

- O kernel opera principalmente no modo privilegiado (modo kernel), onde tem acesso total e direto ao hardware e recursos críticos.
- Os programas de usuário operam com acesso limitado, sem permissão direta para interagir com o hardware ou modificar recursos críticos. As solicitações são feitas via ***chamadas de sistema*** para o kernel.



Modos de Operação: Modo Kernel vs. Modo Usuário

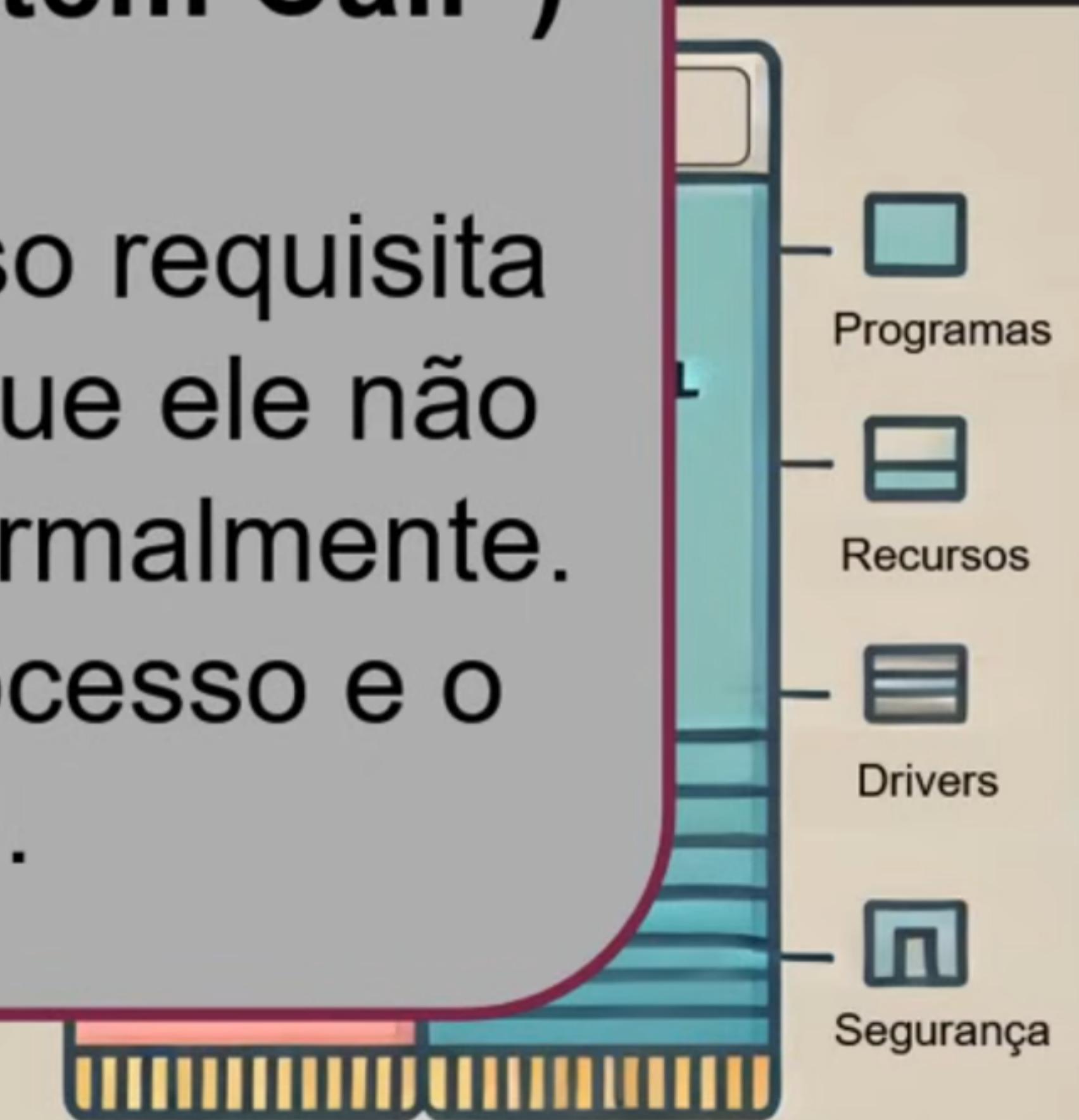
Modos de operação:
os quais
dois prin

- O kernel é o sistema operacional que executa no modo privilegiado, com total acesso a todos os recursos do hardware.
 - Os processos usuários são executados no modo não privilegiado, com limitações de segurança e comuns ao sistema operacional.
- As solicitações das aplicações são feitas via chamadas de sistema para o kernel.

Chamada de Sistema ("System Call")

Maneira pela qual um processo requisita um serviço do kernel do SO que ele não tem permissão de executar normalmente.

É uma interface entre um processo e o sistema operacional.



Kernel Monolítico

- Todos os componentes em um único espaço de memória (modo kernel).
- Desempenho alto, baixa latência.
- Complexo e grande, mais difícil de depurar.
- Falhas podem afetar o sistema todo.

Exemplos: Linux, FreeBSD, AIX, HP-UX.

Software

Kernel

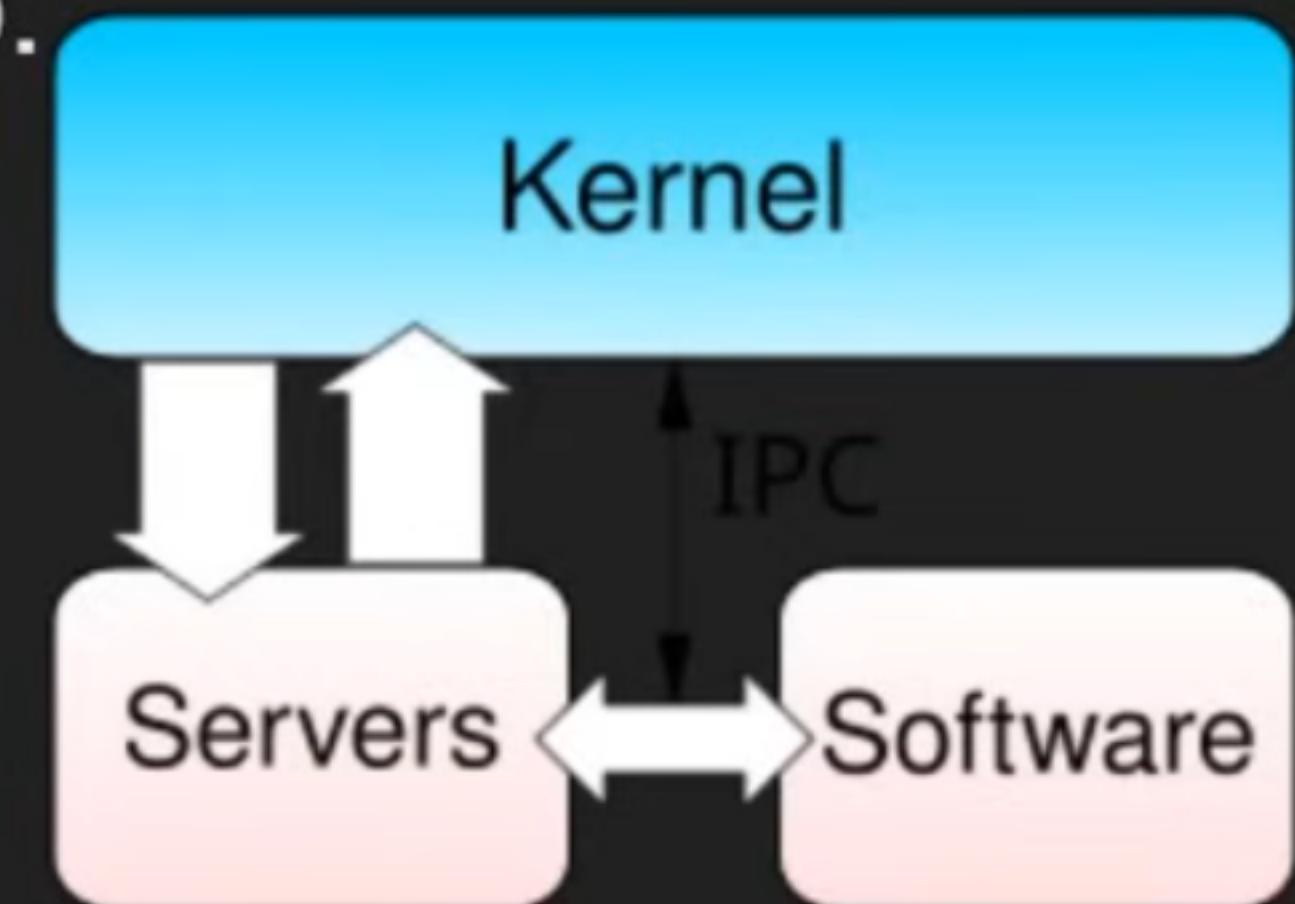
Módulos

Hardware

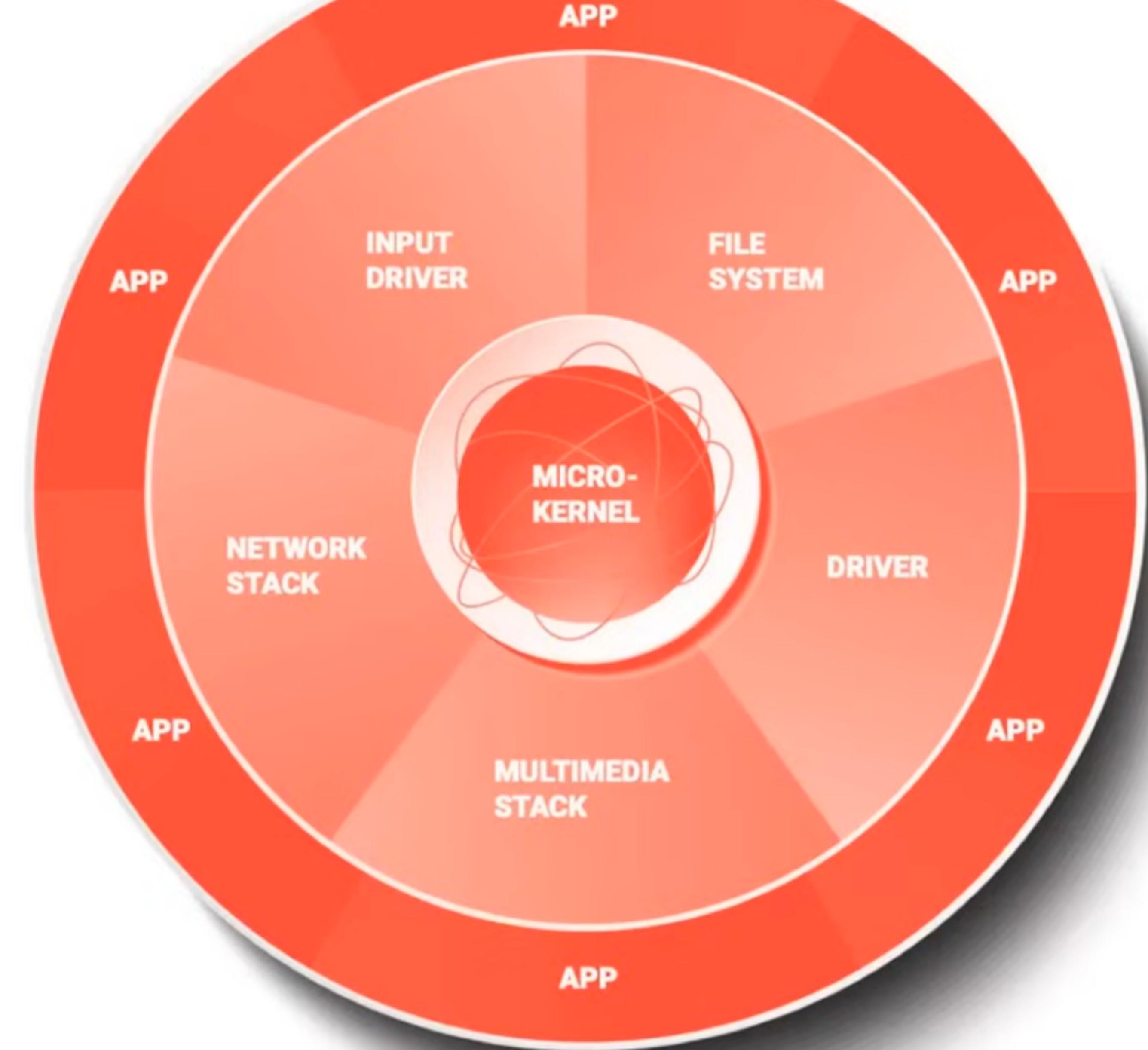
Kernel Monolítico

Microkernel

- Kernel mínimo, funções essenciais no modo kernel.
- Serviços externos (drivers, arquivos) no espaço do usuário.
 - Exemplo: serviços de rede são fornecidos por programas servidores no espaço do usuário.
- Mais seguro e modular (falhas isoladas).
- Mais complexo e com potencial sobrecarga, devido a muitas chamadas de sistema e chaveamentos de contexto.
- Exemplos: Mach, Minix 3, BlackBerry QNX.



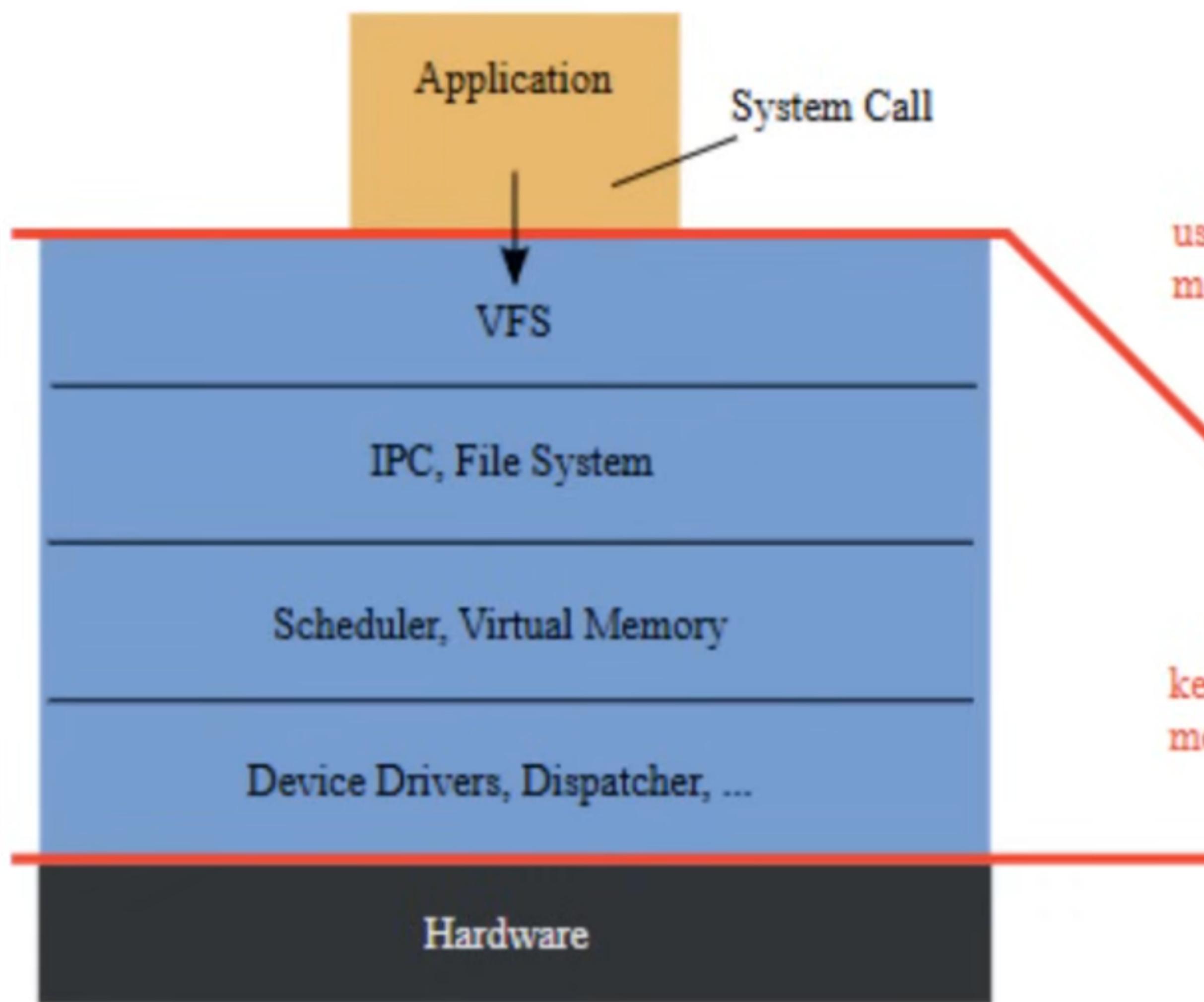
Mi



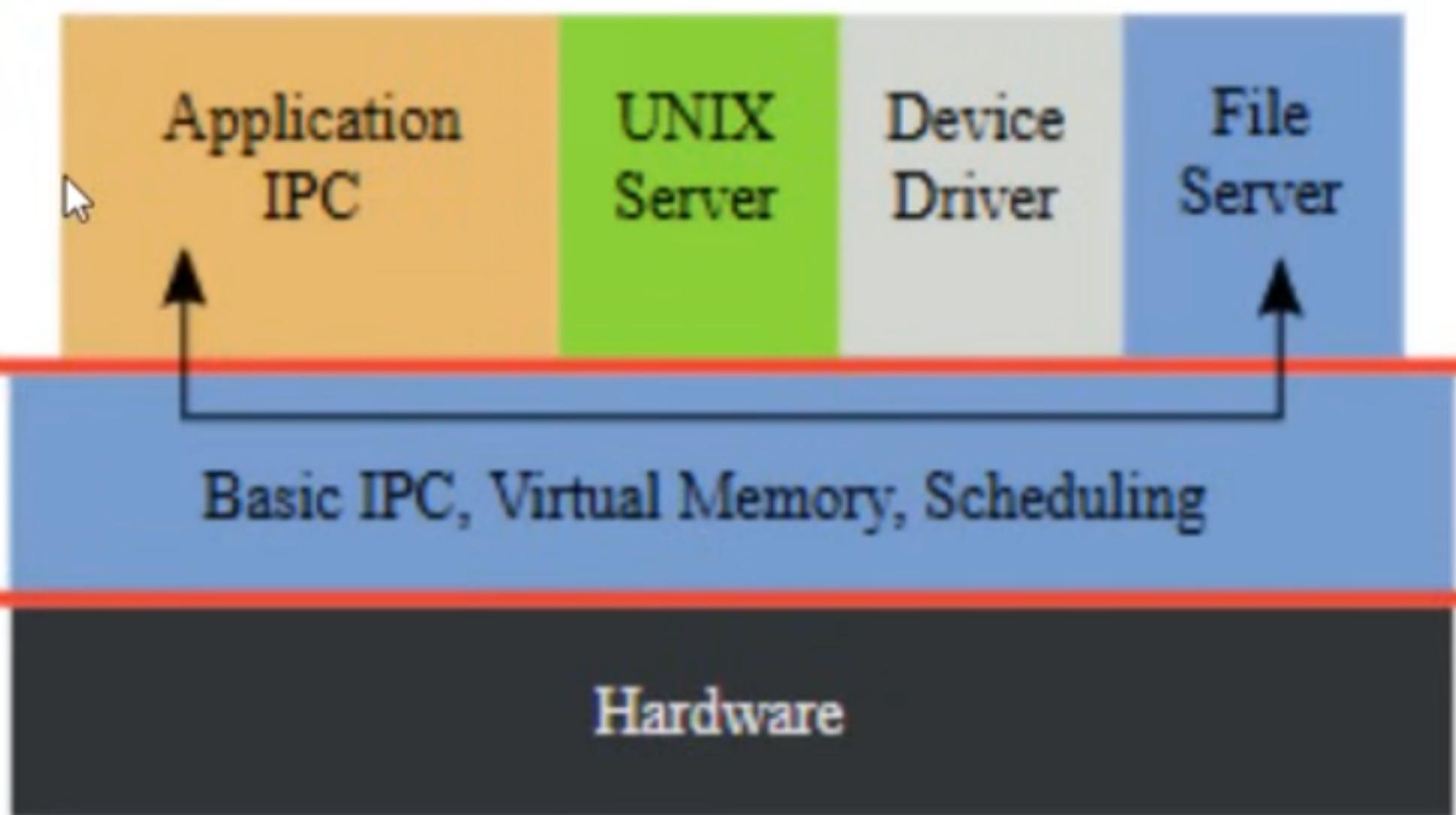
ftware

Sistema BlackBerry QNX Neutrino

Sistema Operacional com Kernel Monolítico

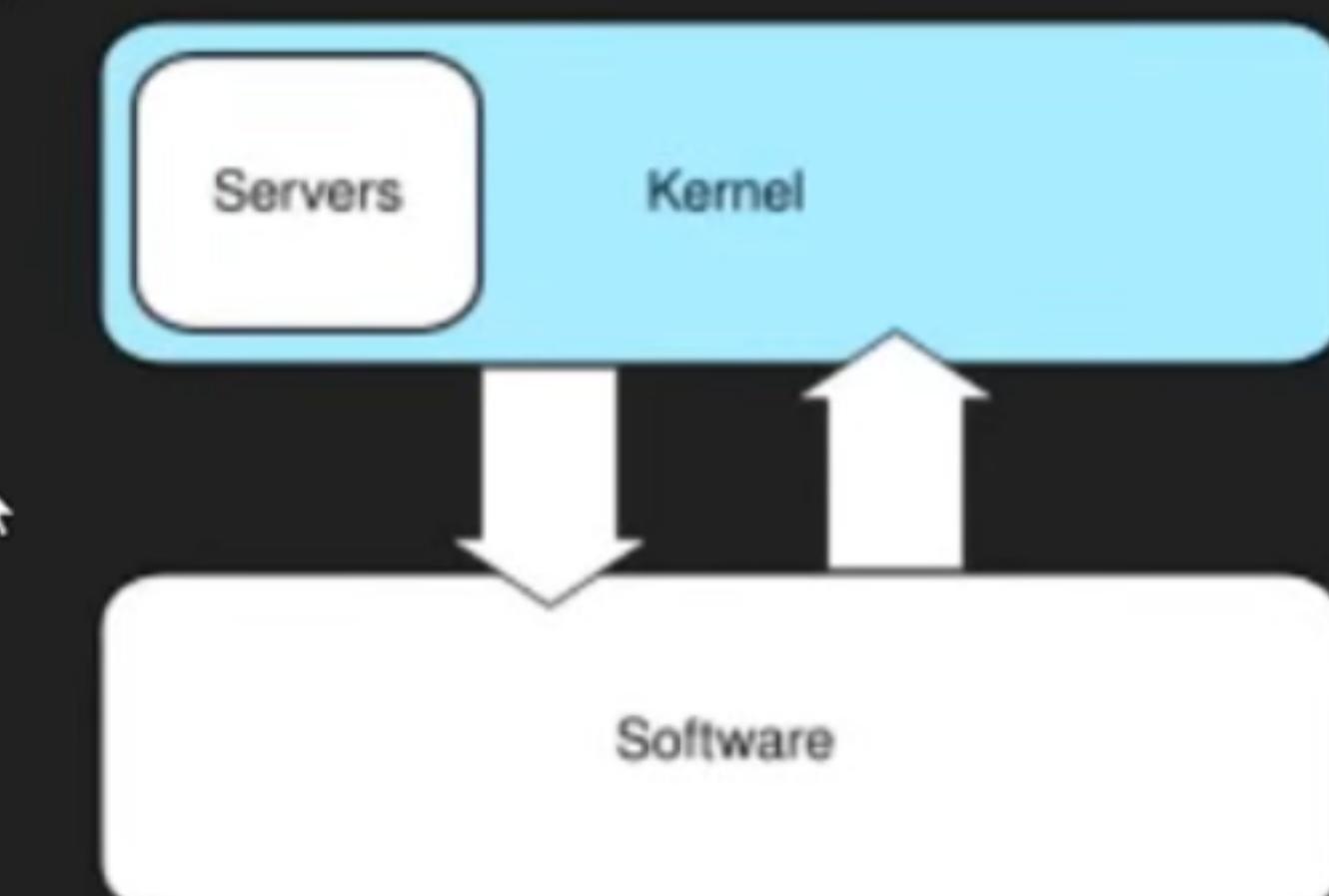


Sistema Operacional com Microkernel

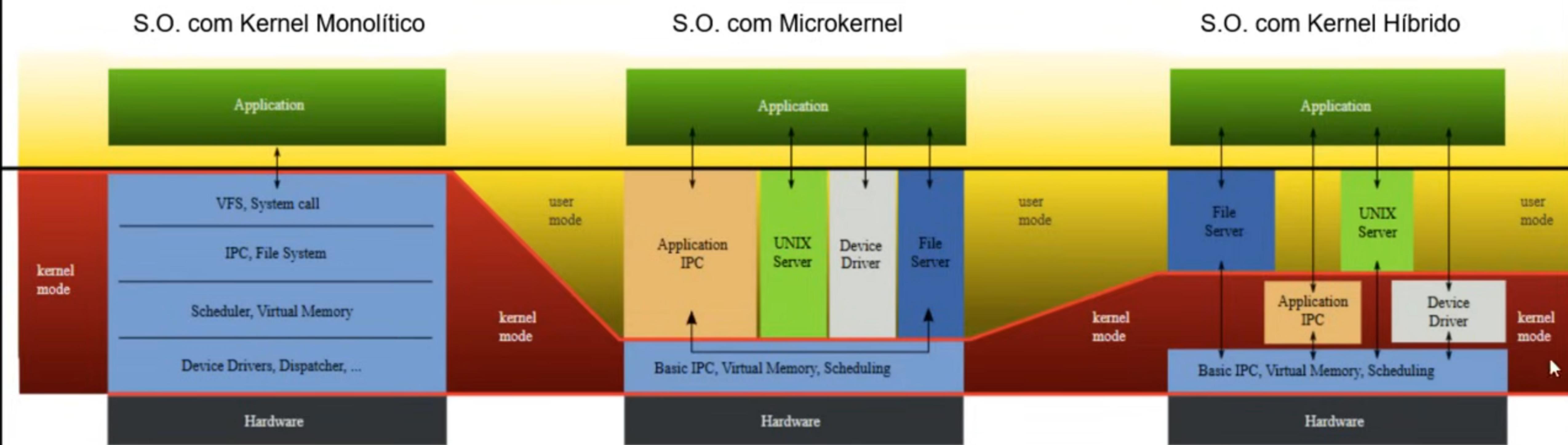


Kernel Híbrido

- Arquitetura de kernel de sistema operacional que combina elementos de kernels monolíticos e microkernels.
- Similar ao microkernel, porém com código adicional rodando no espaço do kernel para aumentar a performance e segurança.
- Tenta unir as vantagens dos kernels monolítico e microkernel.
- Exemplos: NetWare, OS/2, Windows NT, Windows XP / 7 / 10 / 11

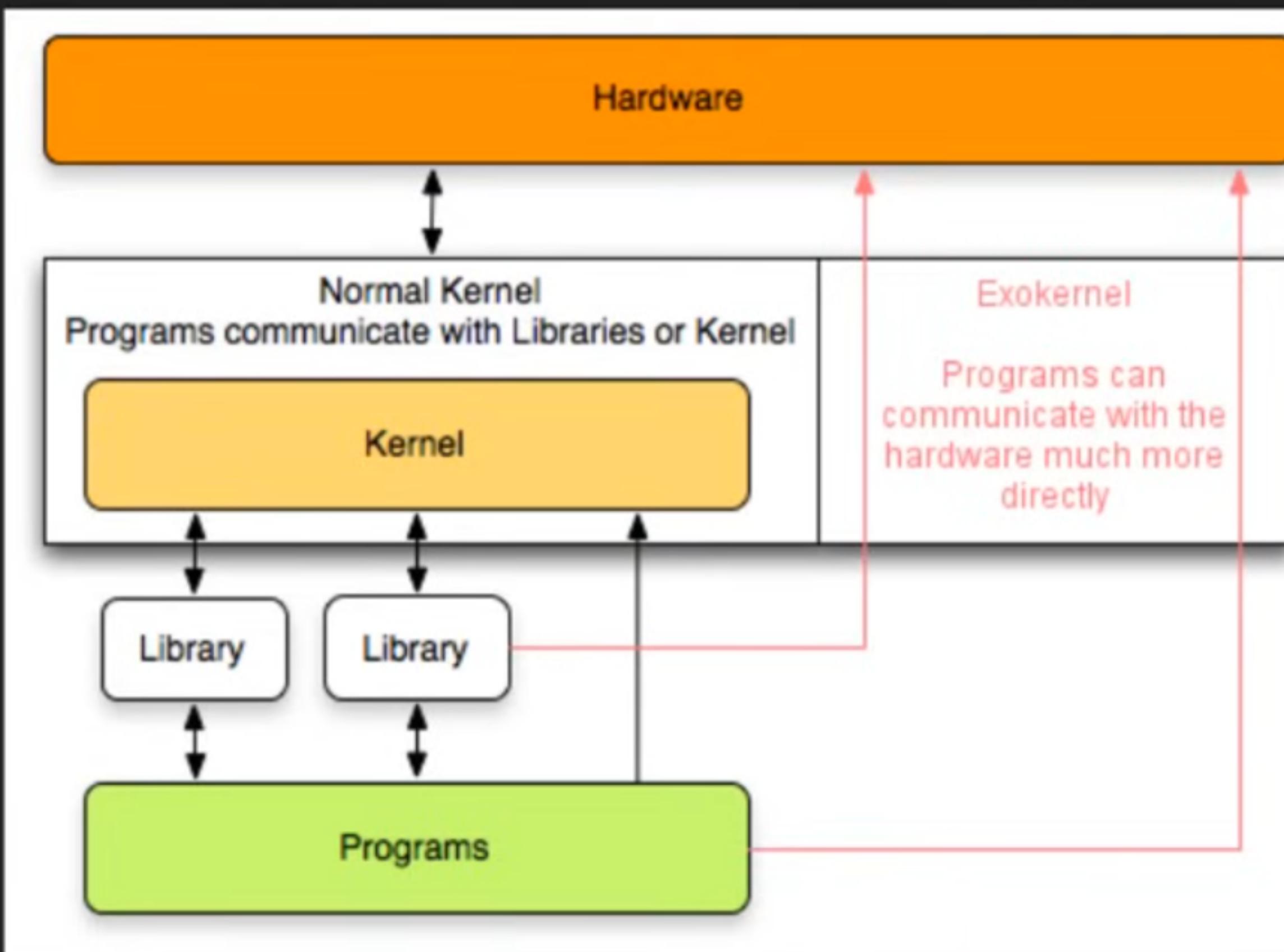


Estrutura de sistemas operacionais baseados em kernel monolítico, microkernel e kernel híbrido:



Exokernel

- Projetado para fornecer abstração mínima entre hardware e aplicativos.
- Expõem recursos de hardware diretamente aos aplicativos.



- Também conhecido como Exonúcleo, nesta abordagem o kernel atua minimamente.
- Exemplos: ExOS, XOK.

Estrutura de um sistema com exokernel.
Imagen: Thorben Bochenek / Wikimedia Commons

Nanokernel



- Abordagem ainda mais minimalista para o design do sistema operacional do que um microkernel.
- Em um nanokernel, o kernel fornece apenas os serviços mais básicos e essenciais, como abstração de hardware e comunicação entre processos (IPC), delegando quase todas as outras funcionalidades para processos de espaço do usuário ou servidores externos.
- Exemplos: KeyKOS, EROS, Mach

Nanokernels e Exokernels são designs mais exóticos, raramente usados em sistemas de produção.

Tabela Comparativa entre Arquiteturas de Kernel

Aspecto	Monolítico	Microkernel	Híbrido	Exokernel	Nanokernel
Estrutura	Serviços rodam no mesmo espaço de memória (modo kernel).	Funções essenciais no modo kernel, serviços no espaço do usuário.	Combina funções monolíticas e externas (microkernel).	Mínima abstração, com aplicativos acessando diretamente o hardware.	Apenas os serviços mais básicos, delegando o resto ao espaço de usuário.
Desempenho	Alto, mas pode sofrer degradação em sistemas grandes.	Pode ser mais lento devido à comunicação entre componentes.	Geralmente otimizado para equilibrar desempenho e modularidade.	Alto, pois os aplicativos acessam o hardware diretamente.	Alto para funções específicas, mas limitado pela simplicidade.
Resiliência (Estabilidade)	Menor, pois falhas podem afetar todo o sistema.	Alta, devido ao isolamento dos serviços.	Moderada, dependendo da implementação dos componentes.	Moderada, depende do controle dos recursos por aplicativos.	Alta, com foco no isolamento mínimo para sistemas críticos.
Modularidade	Baixa, com pouca separação entre os componentes.	Alta, com serviços externos bem definidos.	Moderada, combina modularidade e integração.	Alta, mas exige controle adequado dos aplicativos.	Alta, com foco na simplicidade e escalabilidade.
Exemplos	Linux, FreeBSD	Mach, Minix 3	Windows 10, OS/2	ExOS, XOK	KeyKOS, EROS

Conclusão

- O kernel é base para o funcionamento de qualquer sistema operacional.



Conclusão

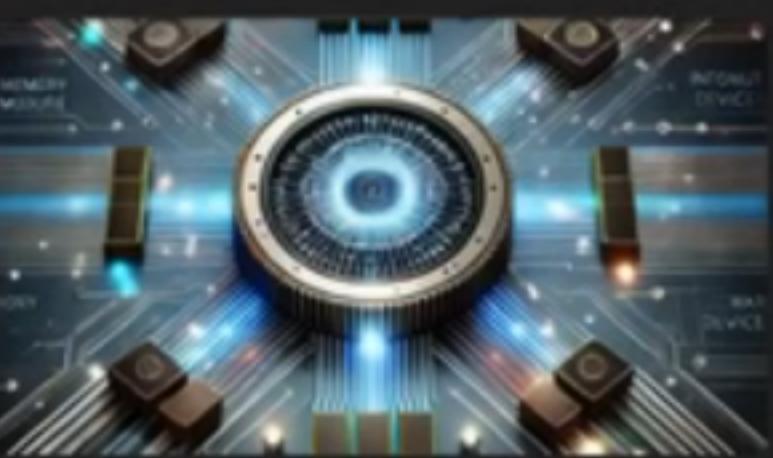
- O kernel é base para o funcionamento de qualquer sistema operacional.
- Diferentes tipos de kernel atendem a diversas necessidades dos sistemas modernos (desempenho, modularidade, segurança, flexibilidade).





Conclusão

- O kernel é base para o funcionamento de qualquer sistema operacional.
- Diferentes tipos de kernel atendem a diversas necessidades dos sistemas modernos (desempenho, modularidade, segurança, flexibilidade).
- Os principais tipos de kernel são:
 - Monolítico: Desempenho elevado, mas menor modularidade.
 - Microkernel: Modularidade e segurança, com serviços externos.
 - Híbrido: Equilibra desempenho e modularidade.



Conclusão

- O kernel é base para o funcionamento de qualquer sistema operacional.
- Diferentes tipos de kernel atendem a diversas necessidades dos sistemas modernos (desempenho, modularidade, segurança, flexibilidade).
- Os principais tipos de kernel são:
 - Monolítico: Desempenho elevado, mas menor modularidade.
 - Microkernel: Modularidade e segurança, com serviços externos.
 - Híbrido: Equilibra desempenho e modularidade.
 - Exokernel: Controle direto do hardware pelos aplicativos.
 - Nanokernel: Estrutura mínima, ideal para sistemas embarcados ou críticos.

Bibliografia Recomendada



- Tanenbaum, A. S.; Bos, H. ***Sistemas Operacionais Modernos***. 4^a Edição. Ed. Pearson. 2016
- Bovet, D. P.; Cesati, M. ***Understanding the Linux Kernel***. 3rd Edition. Ed. O'Reilly Media. 2005.
- Silberschatz, A.; Galvin, P. B.; Gagne, G. ***Fundamentos de Sistemas Operacionais***. 9^a Edição. Ed. LTC. 2018.
- Stallings, W. ***Operating Systems: Internals and Design Principles***. 9th Edition. Ed. Pearson. 2018.