Estrutura do SO

Sistemas Operacionais

Prof. Pedro Ramos
pramos.costar@gmail.com

Pontifícia Universidade Católica de Minas Gerais
ICEI - Departamento de Ciência da Computação

NESTA AULA: ESTRUTURAÇÃO DO SISTEMA OPERACIONAL

- Organização de componentes em um SO

- 4 Exemplos de organização em SO:
  - Kernel monolítico
  - Arquitetura em camadas
  - Micro-kernel
  - Modular

SO's modernos utilizam elementos das 4 abordagens.

Vamos visualizar os *tradeoffs* entre elas!

UMA ESTRUTURA BÁSICA

USUÁRIOS

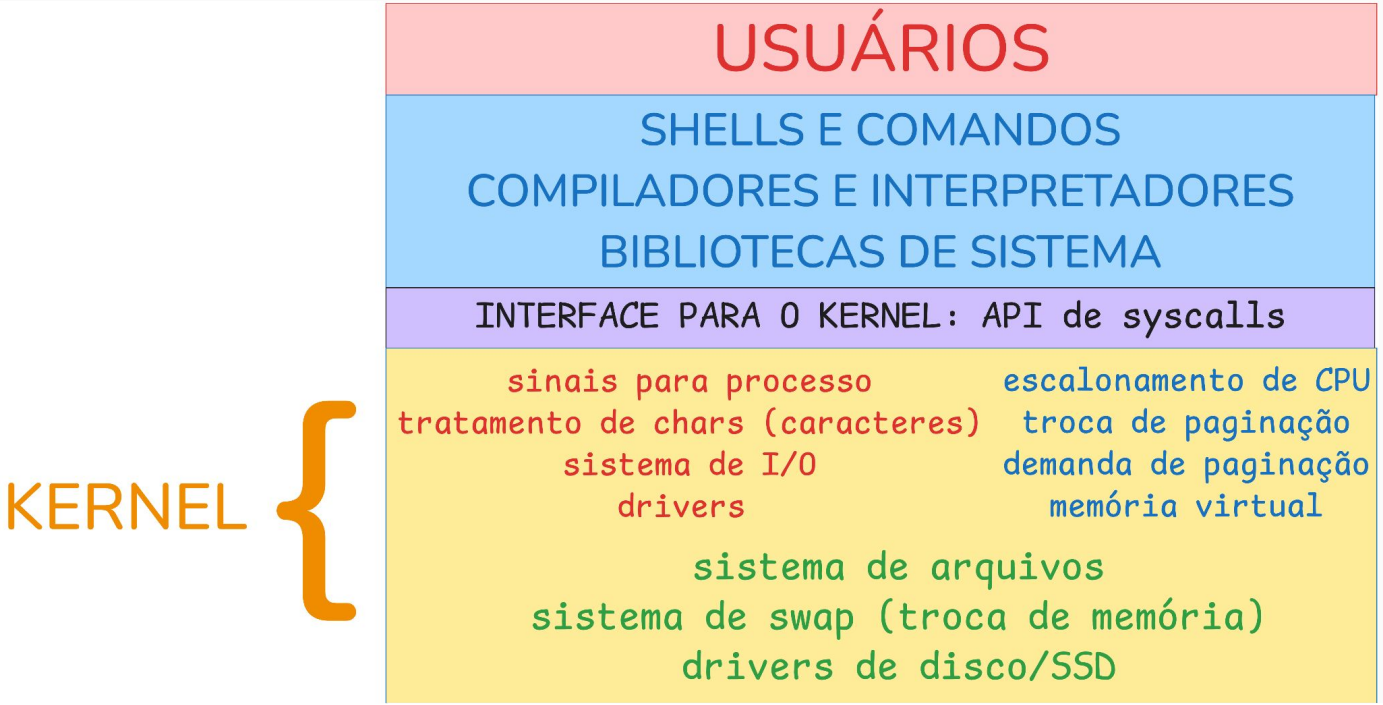
UMA ESTRUTURA BÁSICA

USUÁRIOS

SHELLS E COMANDOS
COMPILADORES E INTERPRETADORES
BIBLIOTECAS DE SISTEMA

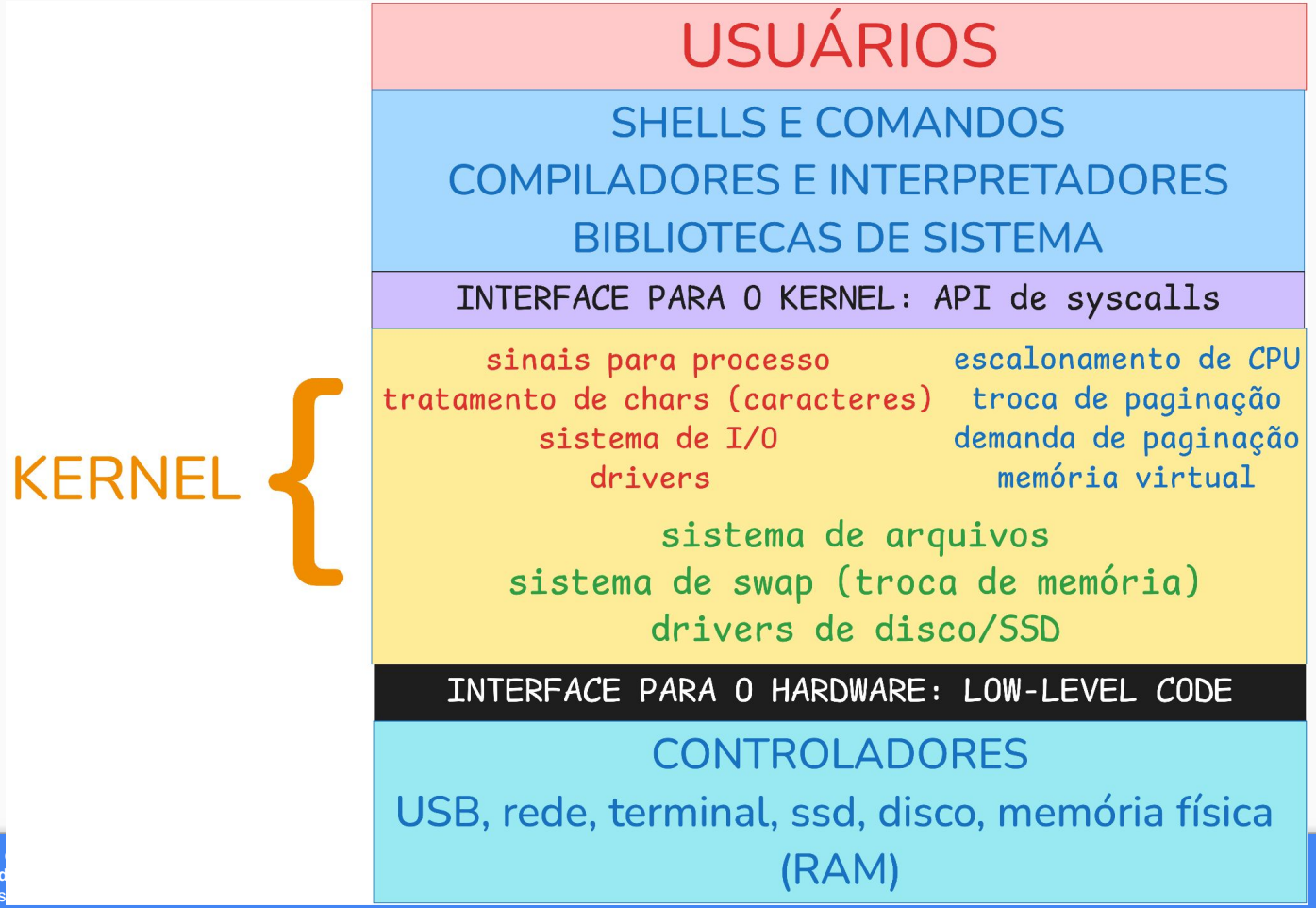
UMA ESTRUTURA BÁSICA

UMA ESTRUTURA BÁSICA



UMA ESTRUTURA BÁSICA

UMA ESTRUTURA BÁSICA



UMA ESTRUTURA BÁSICA

Qual a diferença entre CONTROLADOR-USB e DRIVER-USB?

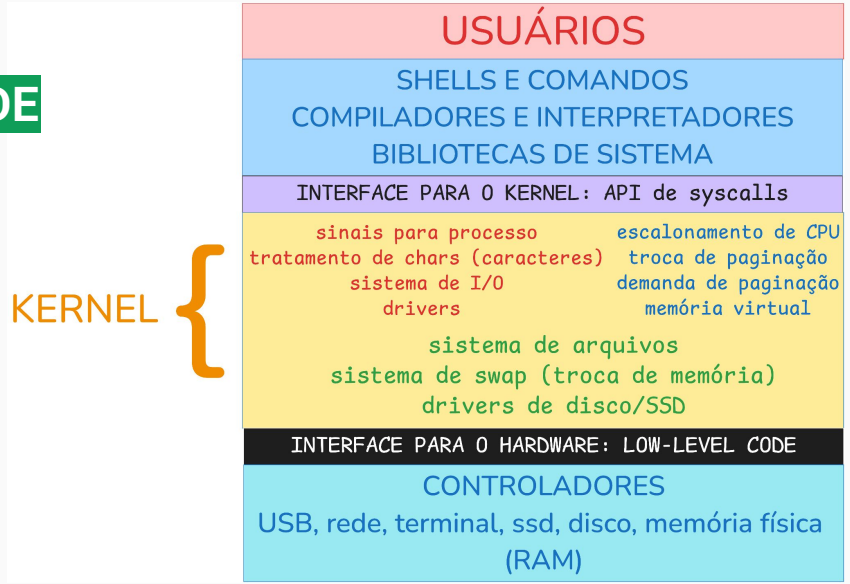
UMA ESTRUTURA BÁSICA

- O KERNEL É A PARTE PROTEGIDA DO SO QUE RODA EM MODO KERNEL,

- LEMBRE-SE: O MODO KERNEL … IMPEDE O USUÁRIO DE ACESSAR AS ESTRUTURAS DE DADOS DO KERNEL, E OS REGISTRADORES DE DISPOSITIVO (I/O)

UMA ESTRUTURA BÁSICA

- A ESTRUTURA REFLETE AS TOMADAS DE DECISÃO.



POR QUE COLOCAR O SHELL FORA DO KERNEL?
E AS BIBLIOTECAS DE SISTEMA? POR QUE NÃO PROTEGÊ-LAS EM MODO KERNEL?

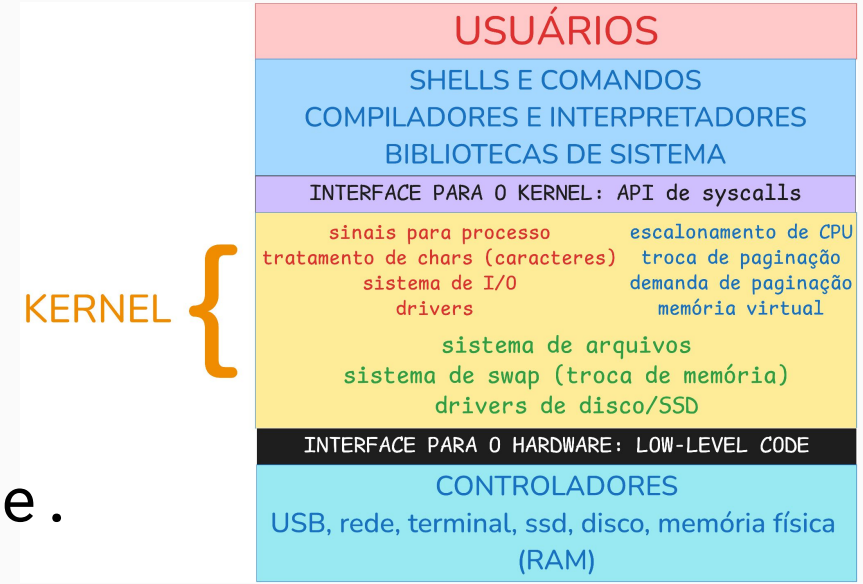
UMA ESTRUTURA BÁSICA

NA IMAGEM AO LADO: UNIX

Kernel **MONOLÍTICO**

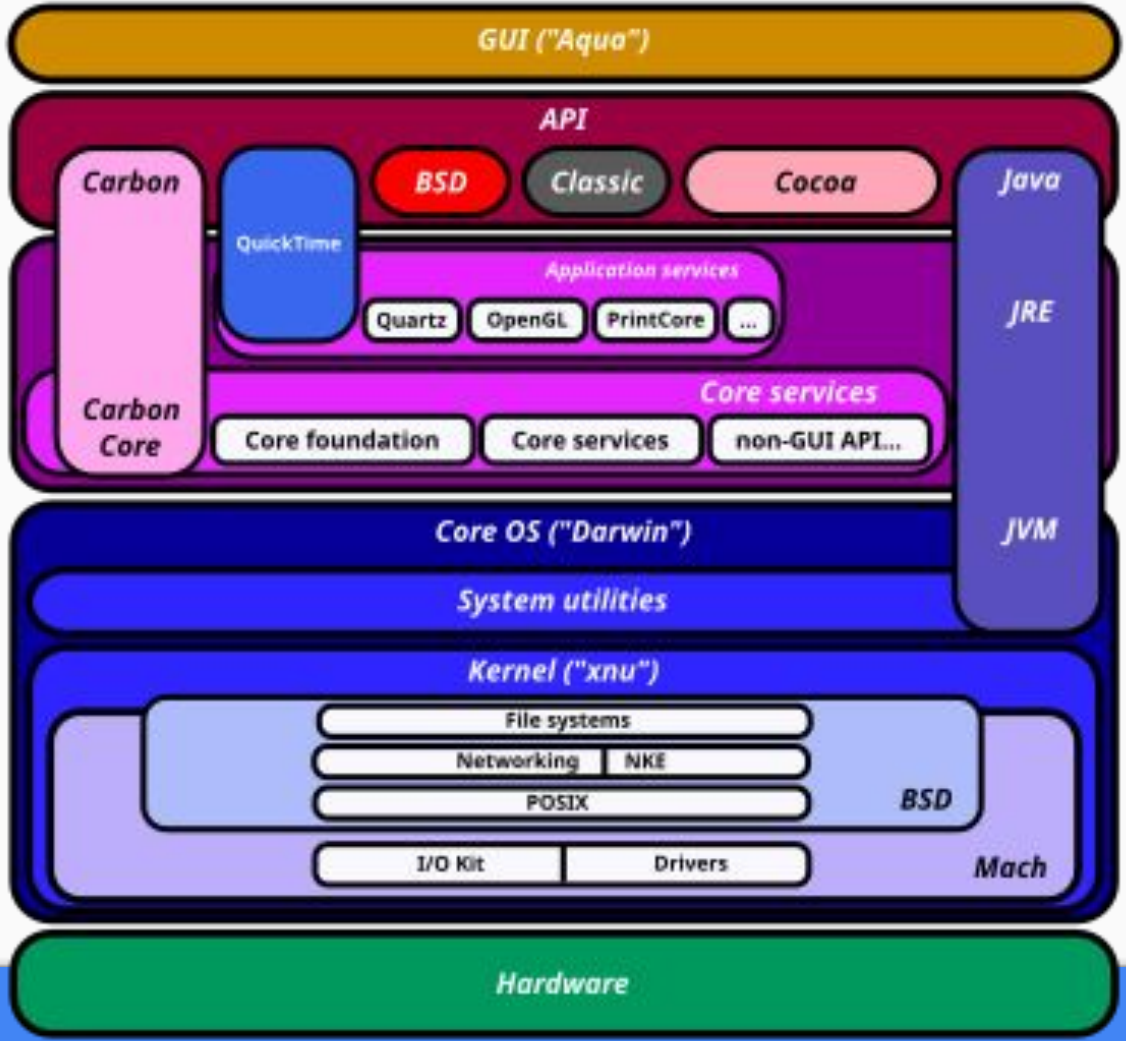
O Kernel é uma ÚNICA unidade.

Tudo no kernel é como se fosse um único processo, executando em um mesmo bloco com o contexto total de todos os serviços.

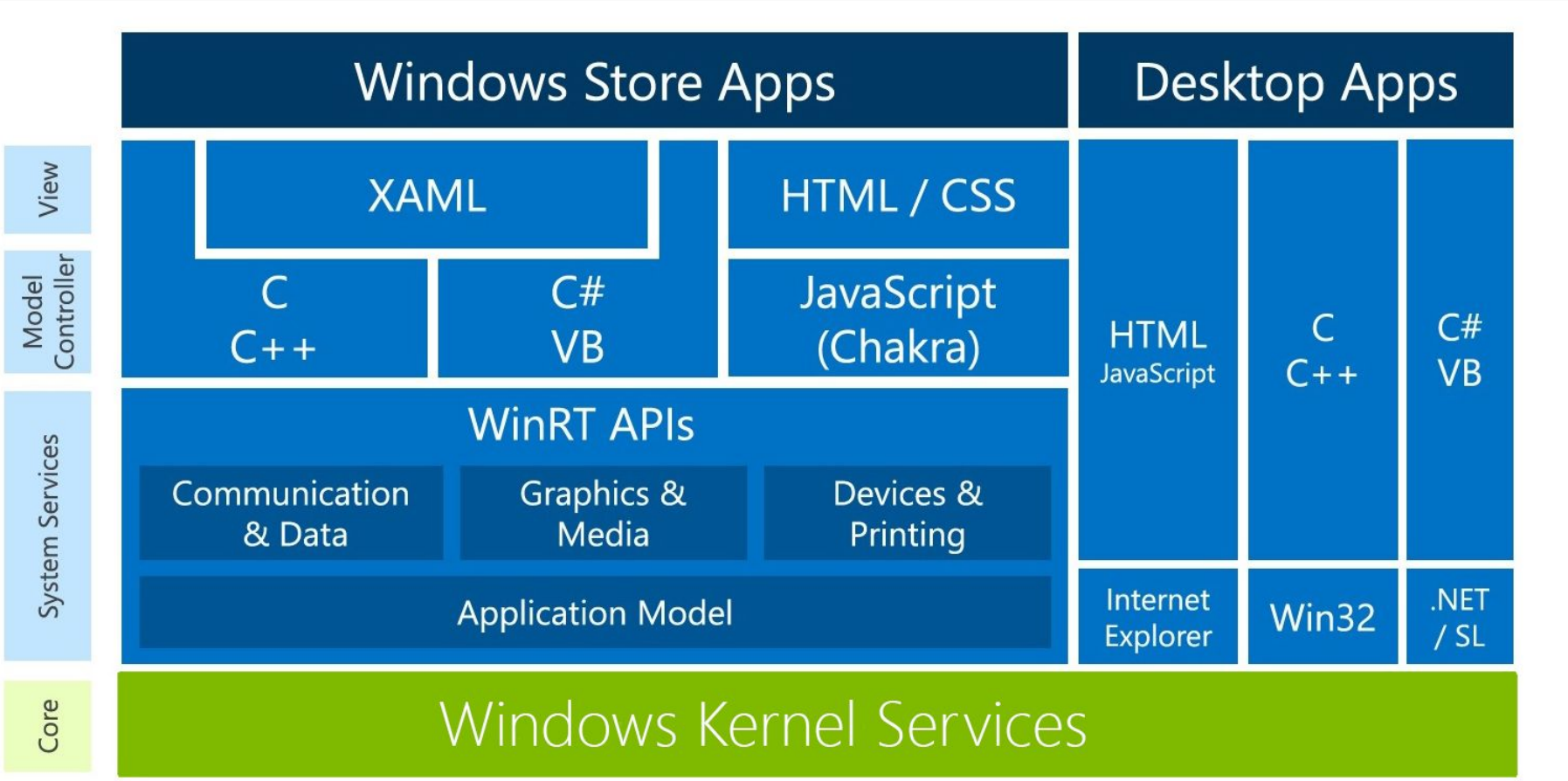


MAC OS X

Para a APPLE, o SO é muito mais que um kernel.



WINDOWS 8


Windows Store Apps
Desktop Apps
View
Model Controller
System Services
Core
XAML
HTML / CSS
C
C++
C#
VB
JavaScript
(Chakra)
WinRT APIs
Communication & Data
Graphics & Media
Devices & Printing
Application Model
HTML
JavaScript
C
C++
C#
VB
Internet Explorer
Win32
.NET / SL
Windows Kernel Services

ESTRUTURA EM CAMADAS

- A CAMADA N USA A CAMADA N-1 E PROVÊ FUNCIONALIDADES PARA A CADAMA N+1

ESTRUTURA EM CAMADAS

Vantagens:

- Modularização
- Manutenção
- Desacoplamento / portabilidade
- Cada camada se preocupa somente em consumir da anterior e produzir uma API para a próxima;
- Debugging

ESTRUTURA EM CAMADAS

Desvantagens:

- Eficiência na comunicação:
  Para disparar uma thread, uma aplicação de usuário precisaria percorrer toda a pilha até o escalonador.
- OVERHEAD (SOBRECARGA DE COMPUTAÇÃO INÚTIL - COMUNICAÇÃO ENTRE CAMADAS)

- Encontrar a **melhor ordem (ÓTIMA)** NÃO É TRIVIAL

ESTRUTURA EM CAMADAS

Desvantagens:

- Eficiência na comunicação:
  Para disparar uma thread, uma aplicação de usuário precisaria percorrer toda a pilha até o escalonador.
- OVERHEAD (SOBRECARGA DE COMPUTAÇÃO INÚTIL - COMUNICAÇÃO ENTRE CAMADAS)

- Encontrar a **melhor ordem (ÓTIMA)** NÃO É TRIVIAL

E SE O ESCALONADOR QUISER USAR A MEMÓRIA VIRTUAL?

ESTRUTURA EM CAMADAS

NEM SEMPRE é possível encontrar uma ordenação LÓGICA entre os SERVIÇOS.

Por causa disso, **Sistemas Operacionais** NÃO são organizados em camadas, exceto…

ESTRUTURA EM CAMADAS

NEM SEMPRE é possível encontrar uma ordenação LÓGICA entre os SERVIÇOS.

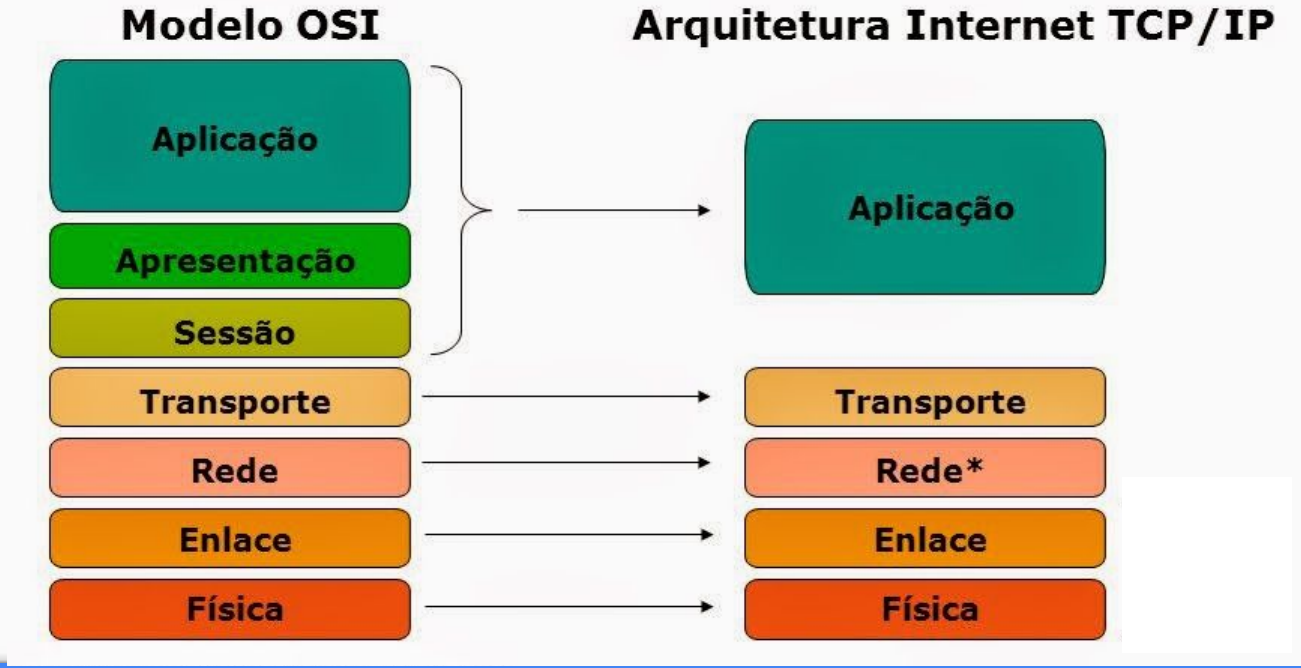
Por causa disso, **Sistemas Operacionais** NÃO são organizados em camadas, exceto…

OS SISTEMAS DE REDE.

ESTRUTURA EM CAMADAS

SISTEMAS DE REDE são organizados em camadas e seus protocolos principais (TCP e UDP) operam somente na camada de transporte.

TCP/IP



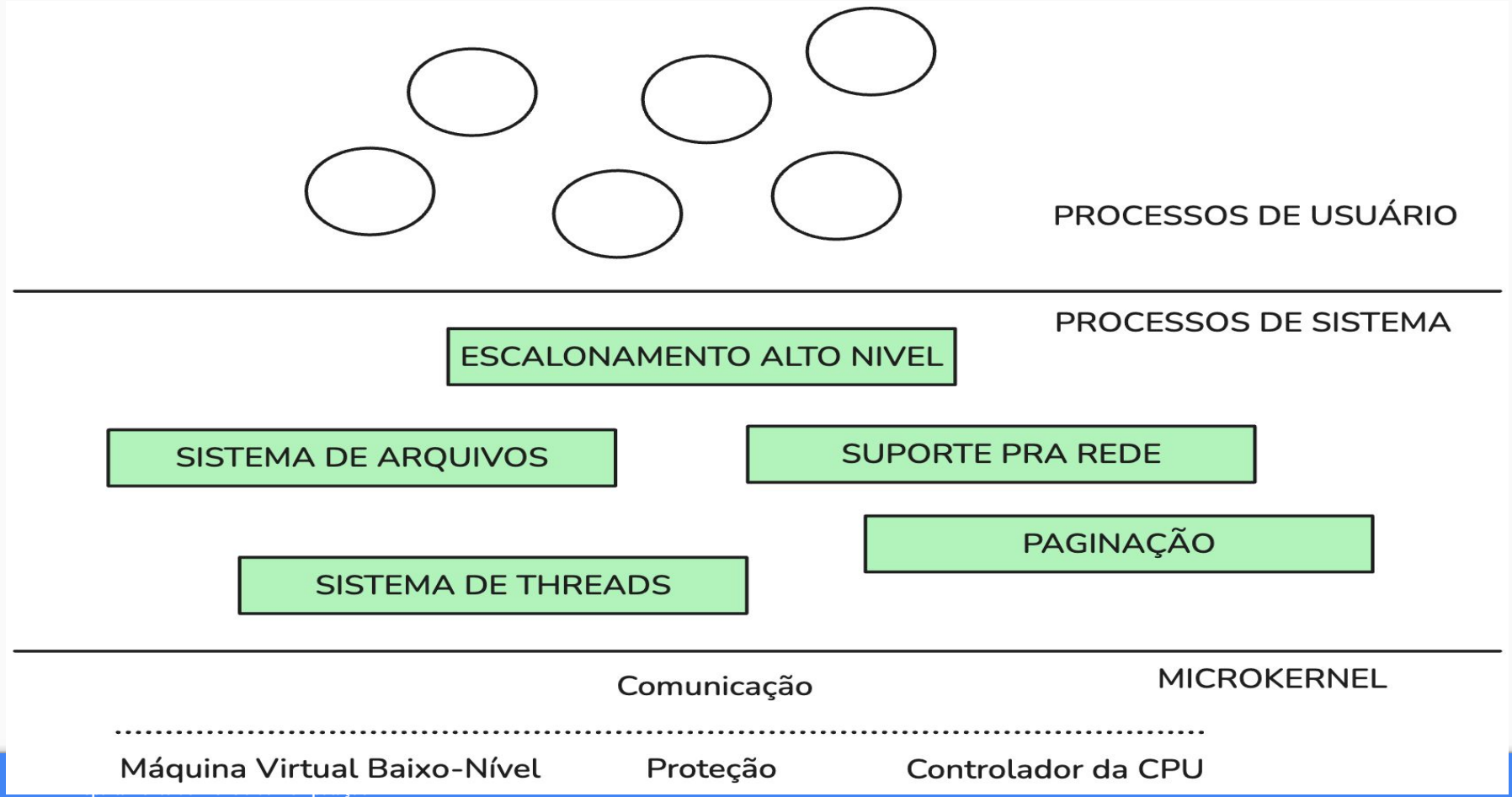
ESTRUTURA MICROKERNEL

O MICROKERNEL É O OPOSTO DO MONOLÍTICO.

- Kernel pequeno que provê comunicação (mensagens) e outras funcionalidades básicas.

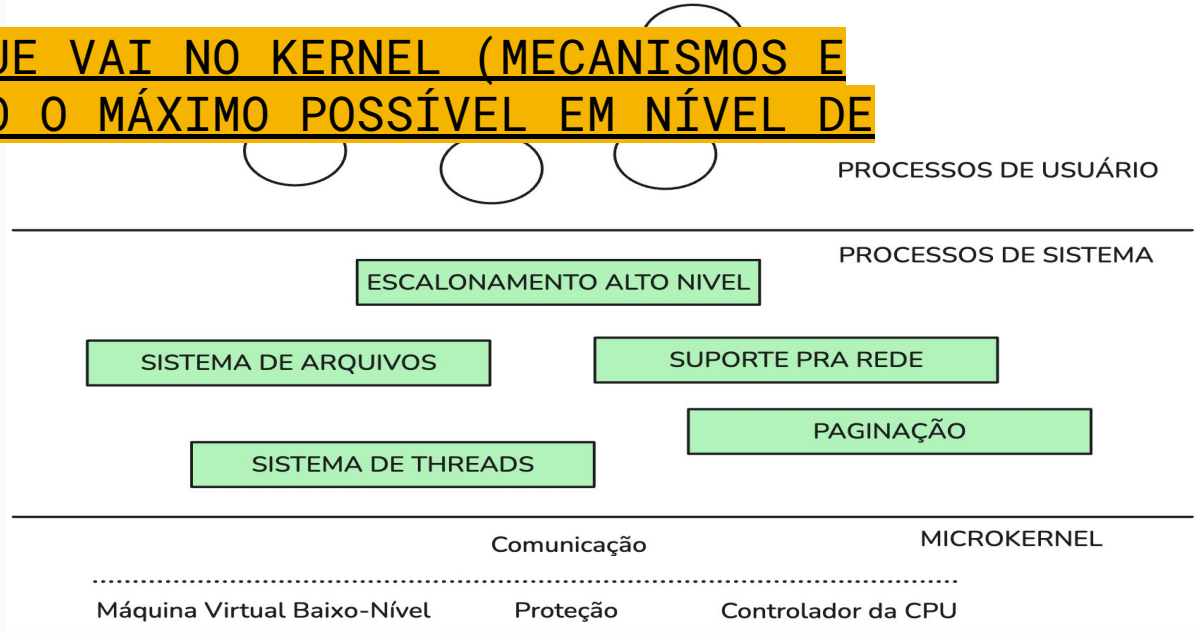
- Outras funcionalidades são **implementadas como processos de usuário.**

ESTRUTURA MICROKERNEL



ESTRUTURA MICROKERNEL

OBJETIVO: MINIMIZAR O QUE VAI NO KERNEL (MECANISMOS E POLÍTICAS) IMPLEMENTANDO O MÁXIMO POSSÍVEL EM NÍVEL DE USUÁRIO.



Vantagens:

Fácil extensão / customização
Micro-kernel simples
Segurança

Desvantagens

Comunicação entre processos do sistema ocorre somente no kernel.

SOLUÇÃO HÍBRIDA

- Para resolver os problemas de performance do microkernel, a solução é **aumentar** o kernel.

- Se há um processo de usuário se comunicando com um processo do sistema, e isso está gerando gargalo no kernel, **então colocamos processo de sistema dentro do kernel.**

SOLUÇÃO HÍBRIDA

- Para resolver os problemas de performance do microkernel, a solução é **aumentar** o kernel.

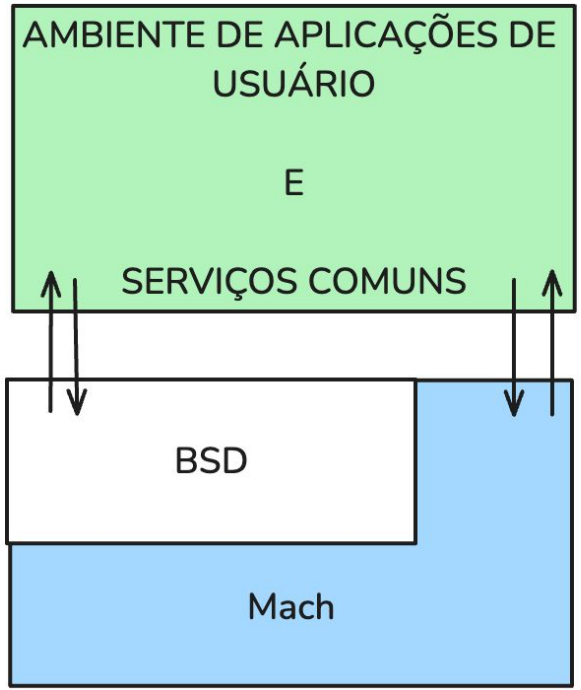
- Se há um processo de usuário se comunicando com um processo do sistema, e isso está gerando gargalo no kernel, **então colocamos processo de sistema dentro do kernel.**

- O **meio do caminho** entre MICROKERNEL e CAMADAS é o **KERNEL HÍBRIDO**

SOLUÇÃO HÍBRIDA: MAC OS X

BSD:

Baseado em UNIX
Threads, CLI, redes, arquivos



Mach:

Memória e comunicação entre processos

MÓDULOS - SO's MODERNOS

- O padrão híbrido é basicamente implementado através de MÓDULOS

- Utiliza **OOP** (Orientação a Objeto)
- Cada módulo é **separado/isolado** e implementado separadamente
- Módulos conversam entre si **através de interfaces públicas**
- Módulos são carregados na memória à **medida que o kernel precisa deles**

- Parecido com a organização em camadas, mas mais flexível.

MÓDULOS - SO’s MODERNOS

- Kernel monolítico: pequena mudança, recompilar o kernel inteiro
- Kernel modular: pequena mudança, recompilar apenas o módulo afetado

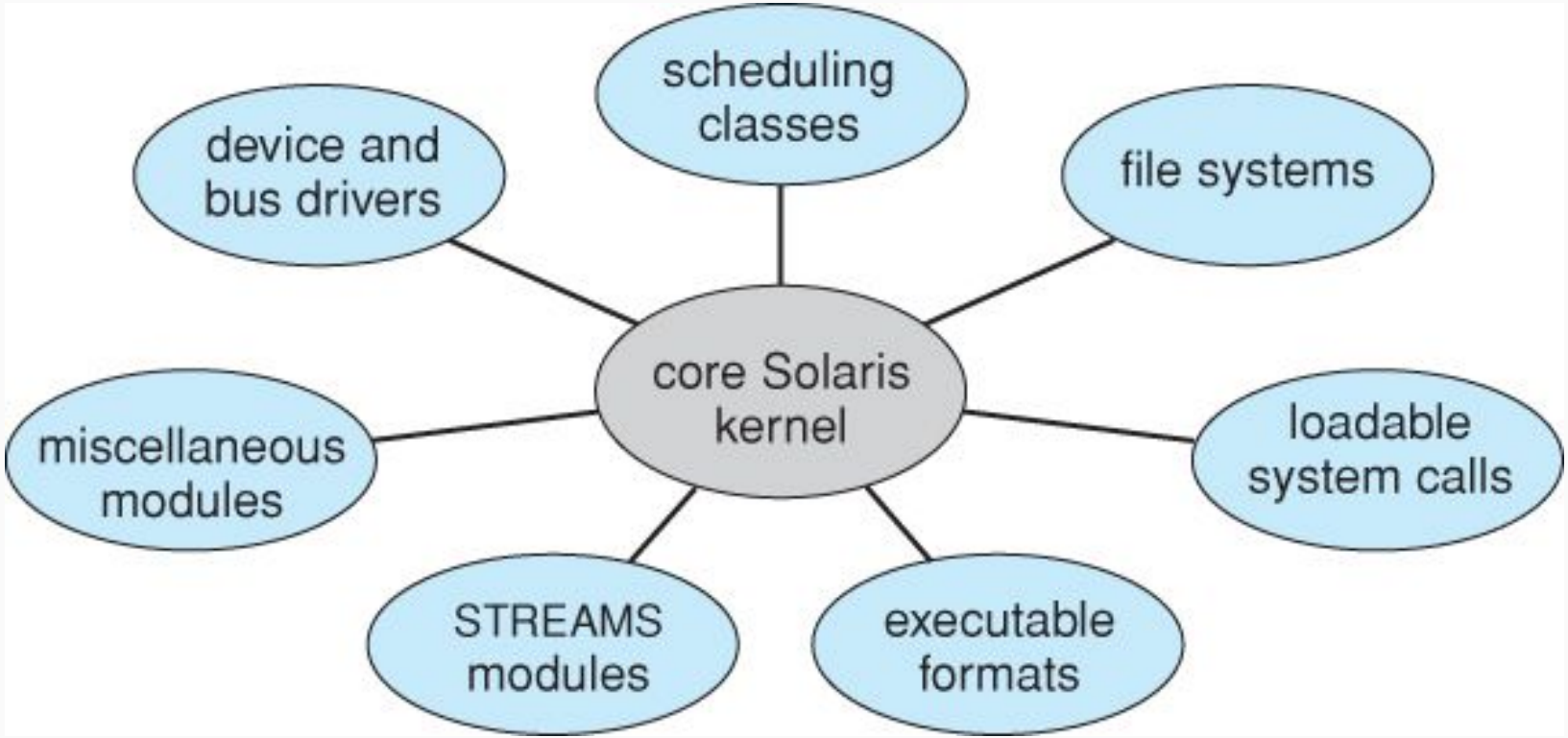
MÓDULOS - SO’s MODERNOS

- Kernel monolítico: pequena mudança, recompilar o kernel inteiro
- Kernel modular: pequena mudança, recompilar apenas o módulo afetado

Exemplo: Solaris (Sistema Operacional)

MÓDULOS - SO's MODERNOS

Exemplo: Solaris (Sistema Operacional)



SEGURANÇA - MONOLÍTICO VS CAMADAS VS HÍBRIDO (MODULAR)

Discussão: qual o método mais seguro de se estruturar um SO? **Monolítico, híbrido ou microkernel ?**

SEGURANÇA - MONOLÍTICO VS CAMADAS VS HÍBRIDO (MODULAR)

Discussão: qual o método mais seguro de se estruturar um SO? **Monolítico, híbrido ou microkernel ?**

Tradeoff: segurança X eficiência

Comunicação ocorre somente no microkernel:
+ seguro, - eficiente

Comunicação ocorre livre entre processos de usuário e sistema:
- seguro, + eficiente

RESUMO

Problema: Como tornar o SO eficiente, confiável e extensível?

Filosofia: A projeção de um SO envolve constante *tradeoff* entre simplicidade e performance.

MÚLTIPLA ESCOLHA - ANALISTA DE SISTEMAS (Cesgranrio)

Um sistema operacional é formado por um conjunto de rotinas que oferecem serviços aos usuários, às suas aplicações e também ao próprio sistema. Esse conjunto de rotinas é denominado núcleo do sistema operacional ou kernel. A estrutura do sistema operacional, ou seja, a maneira como o código do sistema é organizado, pode variar conforme a concepção do projeto. A arquitetura monolítica pode ser definida como um programa:

a) composto por vários módulos que são compilados separadamente e depois linkados, formando um único programa executável.
b) composto por vários níveis sobrepostos, onde cada camada fornece um conjunto de funções que podem ser utilizadas apenas pelas camadas superiores.
c) composto por várias camadas, onde cada camada isola 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.
d) formado por vários níveis, sendo que a camada de nível mais baixo é o hardware, e cada um dos níveis acima cria uma máquina virtual independente, em que cada uma oferece uma cópia virtual do hardware.
e) cujo núcleo do sistema é o menor e o mais simples possível e, neste caso, os servidores do sistema são responsáveis por oferecer um conjunto específico de funções, como gerência de arquivos, por exemplo.

MÚLTIPLA ESCOLHA - ANALISTA DE TI (UFPE)

O sistema operacional desempenha um papel importante no tratamento da E/S, atuando como interface entre o hardware e o software que solicita a E/S. Neste contexto é correto afirmar que:

a) os sistemas de E/S normalmente usam interrupções para comunicar informações sobre operações de E/S. Como essas interrupções causam uma transferência ao modo kernel ou supervisor, elas precisam ser tratadas pelo sistema operacional (SO).
b) não é responsabilidade do sistema operacional fornecer abstrações para acessar dispositivos nem fornecer rotinas que tratam as operações de baixo nível dos dispositivos.
c) o sistema operacional tenta oferecer acesso equilibrado aos recursos de E/S, mas não é responsabilidade do SO escalonar acessos a fim de melhorar a vazão do sistema.
d) o sistema operacional precisa ser capaz de dar comandos aos dispositivos E/S. Esses comandos incluem apenas operações como ler e escrever.
e) o sistema operacional precisa ser capaz de comunicar-se com os dispositivos de E/S mas não pode impedir que o programa do usuário se comunique com os dispositivos de E/S diretamente.

MÚLTIPLA ESCOLHA - ANALISTA DE SEGURANÇA (FGV)

Sistema Operacional é, por definição, um conjunto otimizado de programas que tem por objetivo gerenciar recursos dos computadores.
Nesse sentido, as funções de gerência desempenhadas pelos sistemas operacionais, incluem os seguintes componentes:

a) registradores, unidade de controle, unidade lógica e aritmética e barramentos de interconexão.
b) microprocessador, barramentos USB, slots de memória e controladoras de armazenamento.
c) floppy disk, disco rígido SATA, memória DDR e periféricos de input / output.
d) processamento, memória, dispositivos de entrada/saída e dados.
e) usuários, firewalls, equipamentos de segurança e software.

PERGUNTAS?

REFERÊNCIAS

- **TANENBAUM, Andrew.** Sistemas operacionais modernos.
- **SILBERSCHATZ, Abraham et al.** Fundamentos de sistemas operacionais: princípios básicos.
- **MACHADO, Francis; MAIA, Luiz Paulo.** Arquitetura de Sistemas Operacionais.
- **CARISSIMI, Alexandre et al.** Sistemas operacionais.