

Sistemas Operacionais

Introdução - Arquiteturas de SOs

Prof. Carlos Maziero

DInf UFPR, Curitiba PR

Julho de 2020

Conteúdo

- 1 Sistemas monolíticos
- 2 Sistemas micronúcleo
- 3 Sistemas em camadas
- 4 Sistemas híbridos
- 5 Arquiteturas avançadas
 - Máquinas virtuais
 - Contêineres
 - Sistemas exonúcleo
 - Sistemas uninúcleo

Arquiteturas de SOs

Arquitetura

Forma organizar as várias partes do sistema operacional

Aspectos a considerar:

- Isolamento do núcleo
- Modularização
- Desempenho
- Segurança



Sistemas monolíticos

Mónos (único) + *Líthos* (pedra) = um bloco único

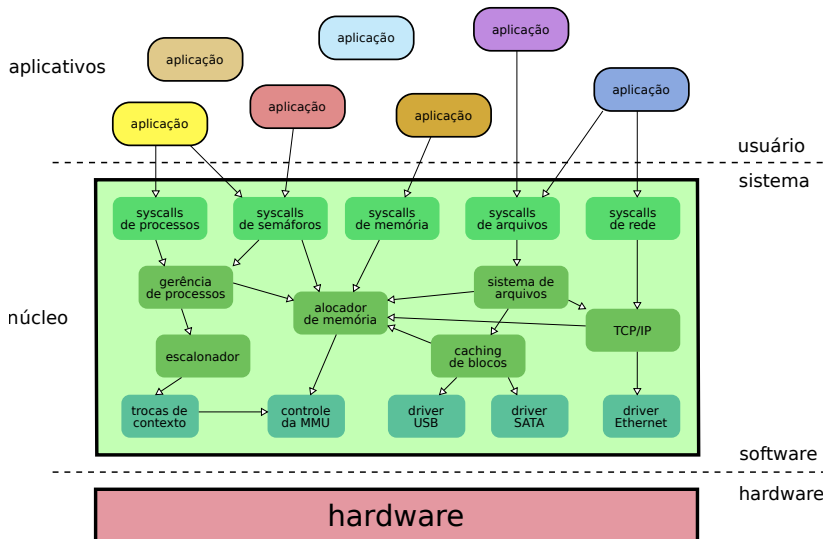
Todo o núcleo roda em modo privilegiado
Sem restrições de acesso entre componentes

Vantagens: desempenho, tamanho

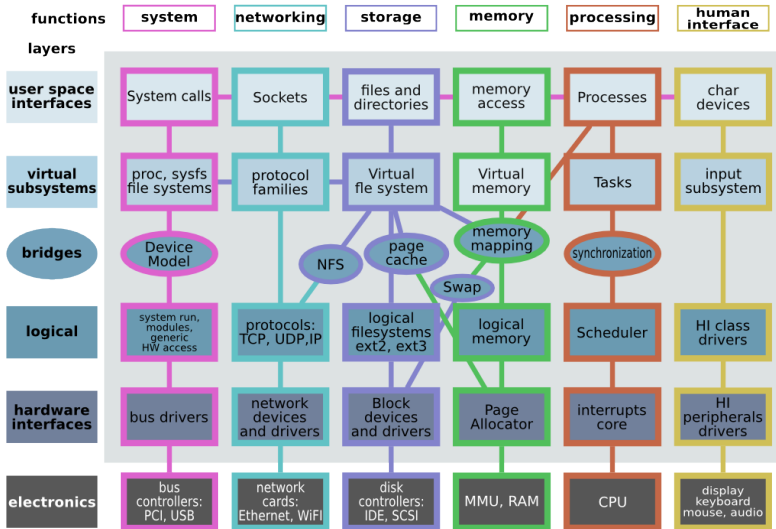
Desvantagens: complexidade, fragilidade



Sistemas monolíticos



Exemplo: núcleo Linux



© 2007-2009 Constantine Shulyupin <http://www.MakeLinux.net/kernel/diagram>

Sistemas micronúcleo

O núcleo implementa:

- espaços de memória protegidos
- tarefa (thread, processo, ...)
- comunicação entre tarefas

Ficam fora do núcleo:

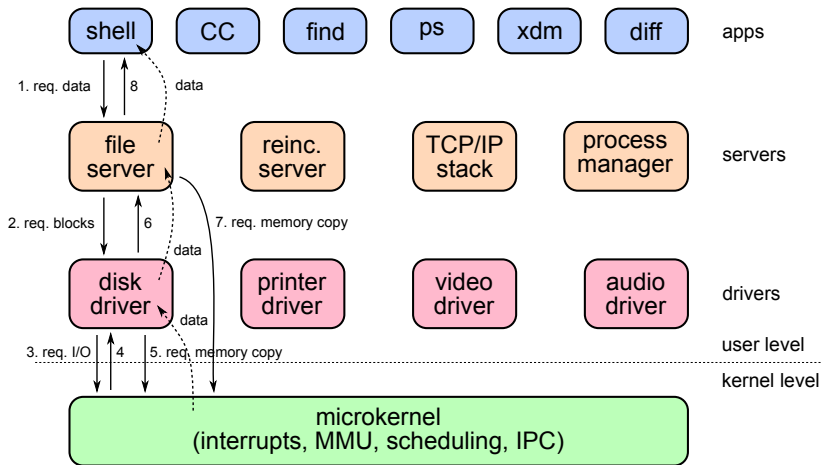
- políticas de escalonamento
- políticas de uso de memória
- sistemas de arquivos
- protocolos de rede

Vantagens: estabilidade, modularidade

Desvantagens: baixo desempenho



Sistemas micro-núcleo - Minix 3



Sistemas em camadas

Princípio: Organizar o núcleo em camadas de abstração

Características gerais:

- Camada inferior: interface com o hardware
- Camadas intermediárias: abstração e gerência
- Camada superior: define as chamadas de sistema

Parcialmente usado na prática:

- *HAL – Hardware Abstraction Layer* do Windows
- Sub-sistemas de arquivos e de rede (modelo OSI)

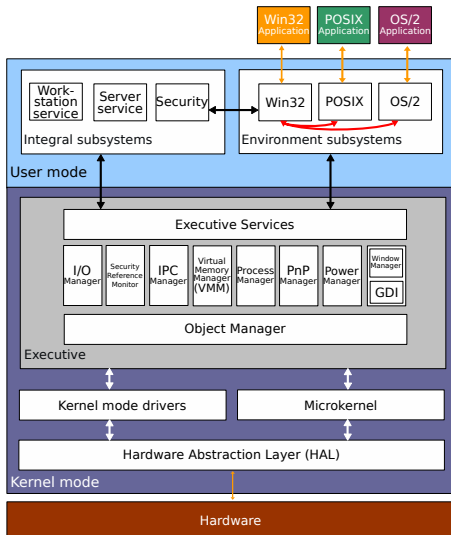
Sistemas híbridos

Misturam características dos anteriores:

- Monolítico
- Micronúcleo
- em camadas

A maioria dos sistemas atuais é híbrida.

Sistemas híbridos - Windows 2000



Máquinas virtuais

Virtualização

Simular em software um sistema computacional sobre outro sistema.

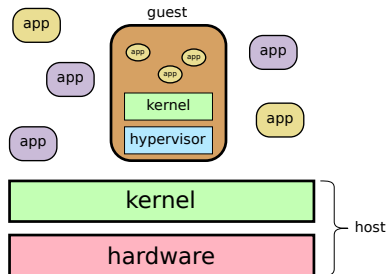
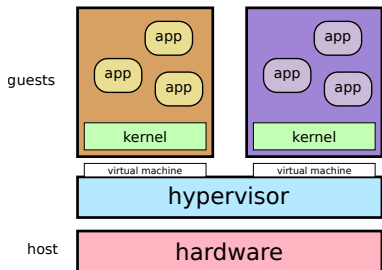
Um ambiente de máquina virtual consiste de três partes:

Host: contém os recursos reais de hardware e software

Hypervisor: constrói o sistema computacional virtual

Guest: executa sobre o sistema virtual

Máquinas virtuais



Famílias de hipervisores

Quanto ao ambiente virtual provido:

HV de aplicação : suporta aplicação convidada (Java, C#)

HV de sistema : suporta SOs convidados (*VMWare*, *VirtualBox*)

Quanto ao suporte de execução:

HV nativo : executa diretamente sobre o hardware (*Xen*)

HV convidado : executa sobre um SO hospedeiro (*VirtualBox*)

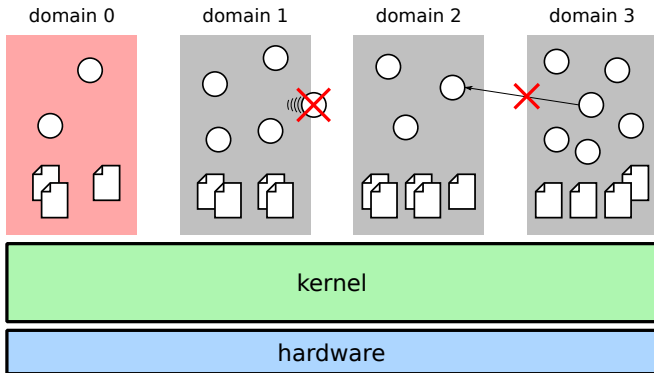
Contêineres

Virtualização do espaço de usuário:

- Espaço de usuário dividido em domínios isolados
- Cada contêiner tem seus próprios recursos
 - usuários, processos, semáforos
 - árvores de diretórios e arquivos
 - interface de rede
- Cada contêiner tem seus próprios *namespaces*
 - UID, PID, IP, ports, ...
- Contêineres compartilham o mesmo núcleo

Exemplos: FreeBSD Jails, Linux Containers (LXC), Docker

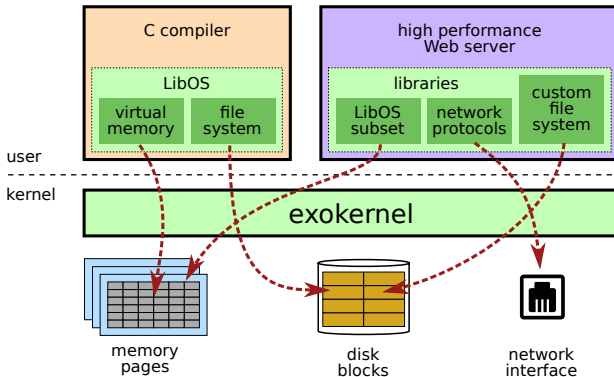
Contêineres



Interações e migrações entre domínios são **proibidas**.

Sistemas exonúcleo

SO dividido em: Micronúcleo + Biblioteca de serviços



Sistemas uninúcleo

Núcleo, serviços e aplicação executam em modo privilegiado

Usado em *appliances* para computação em nuvem (CloudOS)

