

ARQUITETURA DE SISTEMAS OPERACIONAIS

Várias são as estruturas conhecidas de Sistemas Operacionais. Explicaremos de forma resumida algumas das principais estruturas como os *Sistemas Monolíticos*, *Sistemas de Camadas*, *Sistemas Micronúcleo* e *Sistemas Cliente-servidor*.

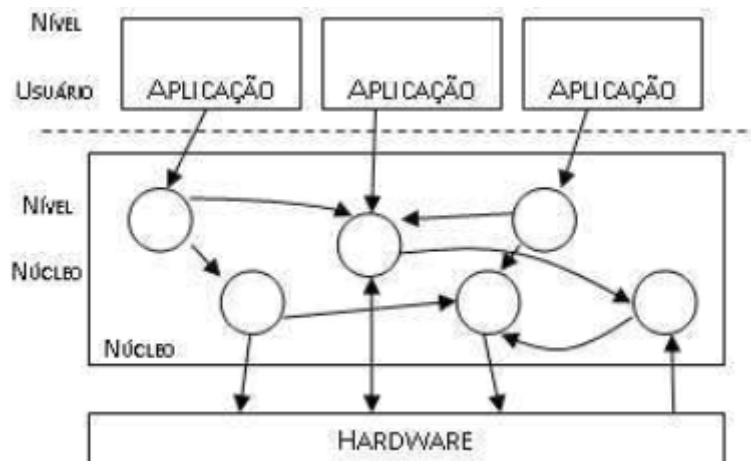
SISTEMAS MONOLÍTICOS

São sistemas que todas as aplicações funcionam a nível núcleo. Na verdade o SO é considerado um emaranhado de rotinas onde o usuário, por meio das chamadas de sistema, operam a máquina.

Monolítico, etimologicamente falando *mono* vem de único e *lítico* de pedra. É como se o núcleo fosse uma "solida pedra" de rotinas que se interagem sem restrições justamente porque o código no nível núcleo tem acesso total a todos os recursos e áreas da memória. Estes sistemas são mais dinâmicos porque cada componente do núcleo pode acessar outros componentes diretamente e com isto, os sistemas podem se tornar mais compactos. Em compensação, o sistema se torna mais robusto, a chance de colapso (travamento, reinicialização ou funcionamento errado) do sistema é maior. A manutenção do sistema é mais complicada visto que todas as partes do núcleo são relacionadas, qualquer mudança em um componente poderá influenciar direta ou indiretamente no funcionamento de outro.

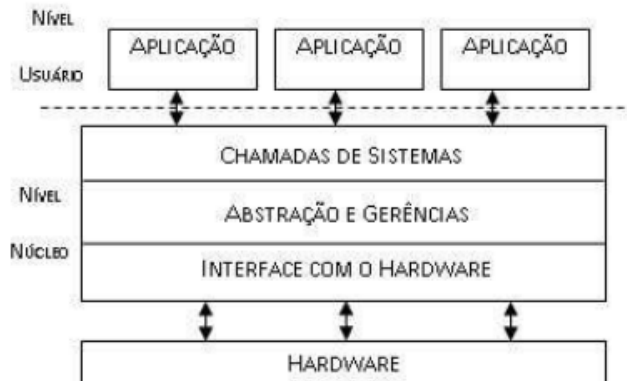
Exemplos comuns de sistemas monolíticos são os Sistemas Operacionais mais antigos como o MS-DOS, o UNIX, Como o Linux originou-se indiretamente do UNIX (como vimos anteriormente), ele também era monolítico, mais com a evolução e desenvolvimento deste sistema, sua arquitetura mudou.

Veja uma representação de um sistema monolítico:



SISTEMAS EM CAMADA

De uma forma mais organizada, o SO com sua arquitetura em camadas tem seu núcleo dividido em camadas de abstração. Cada camada tinha uma importância dentro do núcleo, vamos observar a representação deste sistema:



Notemos que as camadas são bem organizadas e como dito, cada uma com uma função dentro do núcleo. Este não é o único modelo de arquitetura em camadas o sistema em lote THE (*Technische Hogeschool Eindhoven*) foi um dos primeiros neste modelo e era dividido em seis camadas.

Este modelo apresentou problemas substanciais. Quando uma operação precisaria começar na primeira camada e seguir até o hardware ela obrigatoriamente teria que passar por todas as outras camadas aumentando a possibilidade de erro ou inconsistência das informações. O outro problema era que a organização lógica das funcionalidades do sistema em camadas horizontais de abstração não era algo óbvio, o que causava problemas. Por estes motivos, este modelo não foi muito aplicado e hoje em dia é implementado parcialmente nos sistemas.

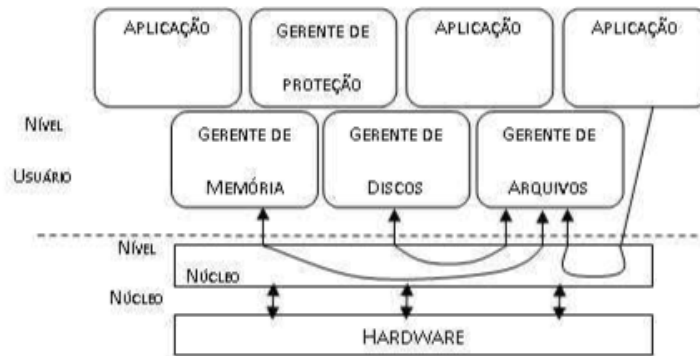
SISTEMAS MICRONÚCLEO

Baseado no sistema em camadas, os projetistas retiraram do núcleo todas as gerências e colocaram em nível de usuário. Se você parar para refletir, qualquer gerente que venha a entrar em colapso seja no monolítico ou no em camadas, o sistema irá entrar em falha geral visto que todos estes gerente e aplicações funcionam a nível núcleo.

Você poderá entender melhor a arquitetura micronúcleo é interessante entender que o núcleo sólido do sistema monolítico e o núcleo organizado do "em camadas" foi dividido em vários pequenos núcleos independentes (daí o nome *micronúcleo*), caso um chegue a causar algum defeito este pode ser restaurado sem causar um colapso geral no sistema. Com isto, esta arquitetura tornou-se mais confiável ao usuário, podendo o usuário agir no sistema sem que tenha uma perda total do sistema.

Na prática, se você está assistindo um filme e ao mesmo tempo um DVD é gravado, caso o processo de gravação gere um erro durante sua execução, você poderá continuar assistindo seu vídeo sem travar a máquina. No sistema monolítico, por exemplo, ele travaria tudo por todas as aplicações serem a nível núcleo.

Veja na imagem abaixo uma pequena representação deste sistema.



SISTEMA DE REDE

Esta arquitetura é uma variação do modelo de micronúcleo. Possui basicamente dois tipos: **Sistema Cliente-Servidor** e **Peer-to-Peer**.

No Sistema Cliente-Servidor, a estrutura é muito parecida, porém os SO diferenciam os processos em dois tipos: **Processos Servidores** que prestam algum serviço e aqueles que utilizam estes serviços que são os **Processos Clientes**. O Núcleo do **Cliente-Servidor** e do sistema **Micronúcleo** é parecido. Este sistema é dividido em dois tipos: *Servidor Dedicado*, onde as máquinas servidoras não executam aplicativos na rede local, limitando-se a atender os processos externos. No tipo *Servidor não Dedicado*, as máquinas servidoras provém serviços tanto para as redes locais como para as redes externas.

No Sistema Peer-to-Peer várias máquinas são ligadas entre si e cada uma delas possui um sistema que tem serviços de servidor e cliente junto com o Sistema Operacional local.

MÁQUINAS VIRTUAIS

A máquina virtual é um recurso muito utilizado nos dias de hoje para que dentro de uma mesma máquina física o sistema operacional possa criar outros "computadores" dentro do mesmo. Como se fosse um clone da máquina original dentro dela mesmo. Desta forma, os usuários podem instalar vários Sistemas Operacionais dentro de um só, então, pode-se instalar um Windows Server 2012, um Suse Linux e um outro sistema qualquer dentro de um Windows 7 instalado no computador.

Os programas de virtualização (Hypervisor, Virtual Box, Microsoft Virtual PC e outros) compartilham os recursos de *hardware* (memória RAM, Disco Rígido, Placa de vídeo, processador e outros) a fim de permitir que a Máquina Virtual (MV) seja praticamente uma máquina dentro de outra permitindo executar qualquer aplicação diferente da máquina real. Evidente que como a máquina virtual compartilha os recursos do hardware, o desempenho da máquina real irá diminuir a medida que as MV são ligadas.

O Recurso de MV será utilizado nesta disciplina para testarmos a instalação de vários tipos de Sistemas Operacionais livres.