

## **Arquitetura Monolítica**

Esta é a arquitetura mais antiga e mais comum, composta e simples. Cada componente do sistema operacional é contido no núcleo (kernel) e pode se comunicar com qualquer outro componente diretamente.

Essa intercomunicação direta permite rapidez na resposta de sistemas operacionais monolíticos, porém, como núcleos monolíticos agrupam os componentes todos juntos, acaba se tornando difícil a identificação da origem de um determinado problema ou erro.

Além disso, todo o código do sistema operacional é executado com acesso irrestrito ao sistema, o que pode facilitar a ocorrência de danos provocados intencionalmente.

Temos como exemplos:

- MSDOS - IBM/Microsoft
- UNIX - Digital
- FreeBSD - FreeBSD

## **Arquitetura em Camadas**

Ao decorrer do tempo os SO's ficaram mais complexos e maiores, os projetos que faziam uso da arquitetura monolítica se tornaram impraticáveis por esse motivo a arquitetura por camadas era uma boa opção para os projetos.

A "Arquitetura em Camadas" se consiste em agrupar em camadas de componentes, ou seja, conjuntos que procedimentos que tinham o propósito de realizar tarefas similares.

Cada camada se comunica apenas com suas camadas inferiores e superiores. Uma camada inferior, sempre presta serviço a sua superiora, e a camada superior não sabe como aquela tarefa (serviço) foi feita, ela apenas solicita.

A implementação de uma camada pode ser feita sem atingir uma outra camada, pois elas possuem componentes autocontidos.

## **Arquitetura de Micronúcleo**

A arquitetura de micronúcleo ou microkernel, é uma forma de arquitetura modular ou em camadas.

Na tentativa de reduzir os procedimentos fundamentais, em pequenos números de serviços como:

- Gerenciamento de memória;
- Sincronização entre processos;
- Comunicação entre processos, que terá acesso direto ao hardware.

Por sua vez, o serviço de comunicação entre processo, que está dentro do micronúcleo, é o responsável por habilitar os serviços, por exemplo:

- Redes;
- Sistemas de arquivos;
- Gerenciamento de dispositivos.

Que normalmente, podem ser implementados no núcleo do SO, não no micronúcleo, ou até como procedimentos externos ao núcleo.

## **Chamadas ao Sistema**

As chamadas ao sistema ou *system calls* são interrupções *traps* originadas por software, que fornecem a interface entre processos e o sistema operacional.

Uma atividade protegida por uma interrupção é específica do sistema operacional e, portanto, tem privilégio sobre a anterior, que, posteriormente, será restaurada. As chamadas ao sistema são semelhantes às chamadas de subrotinas, entretanto, enquanto que as chamadas de subrotinas são transferências para procedimentos do mesmo programa que fez a chamada, as chamadas ao sistema transfere a execução para o sistema operacional, e, por meio de parâmetros, o programa solicitante, informa exatamente o que necessita.

Por exemplo, se um programa lista o conteúdo de um arquivo texto na tela, com certeza fará uma chamada ao sistema para verificar se o arquivo existe é um dos parâmetros dessa chamada será nome do arquivo a ser listado.

## **Interpretador de Comandos**

Apesar do interpretador de comandos, em geral, não fazer parte do núcleo do sistema operacional, possui a importante tarefa de prover a interface primária entre o usuário e o sistema operacional, sem a qual, muitas vezes, não seria possível para o usuário acessar o sistema de computação.

Pode-se visualizar o interpretador de comandos como uma cápsula circundando o sistema operacional, ou seja, criando uma camada, uma interface, entre o usuário e o sistema operacional, permitindo que o usuário execute comandos para acessar todos os recursos que estiverem disponíveis no sistema de computação.

Um interpretador de comandos simples é basicamente um programa composto de um laço que aguarda uma entrada (um comando), cria um novo processo, com uma chamada ao sistema (no Unix ou no Linux seria a chamada *fork*), atribui a esse novo processo o programa especificado na entrada e executa o processo (no Unix ou no Linux, tanto a atribuição do programa quanto a execução do processo são feitos pela chamada *execução*).

---

## **Fontes:**

Material de Apoio:

[Aula 03 - 13-10 - Conceitos básicos de sistema operacionais \(.pdf\)](#)