

Aula 4: Estruturas do Sistema Operacional

Um sistema tão grande e complexo como um moderno sistema operacional deve ser cuidadosamente desenvolvido, para funcionar apropriadamente e ser facilmente modificado. A estrutura e o funcionamento de um sistema operacional são tópicos de difícil compreensão. Um sistema operacional não é executado como uma aplicação sequencial, com início, meio e fim. As rotinas do sistema operacional são executadas sem uma ordem predefinida. Existem diferentes estruturas de sistemas operacionais, veja a seguir as principais delas.

Sistemas Monolíticos

Também conhecida como estrutura simples, esta é a estrutura dos primeiros sistemas operacionais. Consistida, basicamente, por um programa dividido em sub-rotinas, na estrutura monolítica é permitido a qualquer uma dessas sub-rotinas em qualquer parte do programa chamar outra(s) sub-rotina(s). A construção do programa final é dada com base nos módulos compilados separadamente, unidos através de um linker. A boa definição de parâmetros de ligação entre as diferentes rotinas existentes aumenta e muito o desempenho, porém o sistema pode parar devido a algum erro ocasionado por uma dessas rotinas. A exemplo temos o próprio UNIX, o MS DOS, o FreeBSD, dentre outros.

Sistemas monolíticos

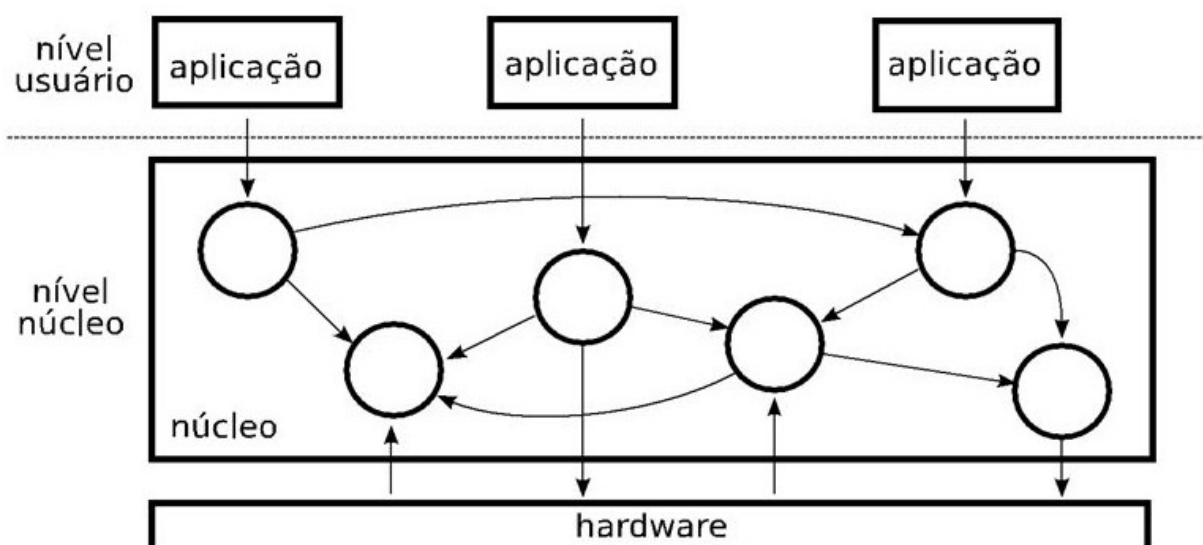


Figura 1: Estrutura e funcionamento de sistemas monolíticos.

Sistemas em camadas

Devido ao crescimento das necessidades dos utilizadores e o aperfeiçoamento dos sistemas, foi-se necessário a modularização da organização do software do sistema operacional. Na abordagem em camadas, o sistema operacional é particionado em níveis, onde o nível mais “baixo” é o hardware, e o nível mais “alto” é a interface com o usuário. A principal vantagem dessa estrutura é justamente a modularização, facilitando sua alteração e depuração de cada camada, além de criar uma hierarquia de níveis de acesso que permite proteger as camadas mais internas.

As camadas são selecionadas de tal modo que cada uma delas opere diretamente com a camada seguinte de nível mais baixo. Foi originado na Holanda por Edsger Dijkstra, que utilizou o algoritmo de busca de menor caminho, também de sua própria autoria, para percorrer dentre as várias camadas, as que atenderão as solicitações de “cima”, de maneira mais eficiente.

Uma desvantagem desta estrutura é o tempo de resposta ao usuário, pois numa requisição sua, uma camada irá se comunicar com a outra diretamente seguinte, e assim por diante, possibilitando a modificação de parâmetros a cada camada, necessidade de dados e ainda o acréscimo de overhead à chamada do sistema, levando, contudo, ao consumo maior de tempo do que nos sistemas operacionais não estruturados em camadas.

Por sua vez, tal estrutura tem uma vantagem maior do ponto de vista tanto do projeto quanto da implementação, pois a impossibilidade de comunicação direta das aplicações de cima com as de baixo, leva a um controle maior do sistema operacional sobre o hardware. A exemplo temos o Windows NT, o THE e o MULTICS.

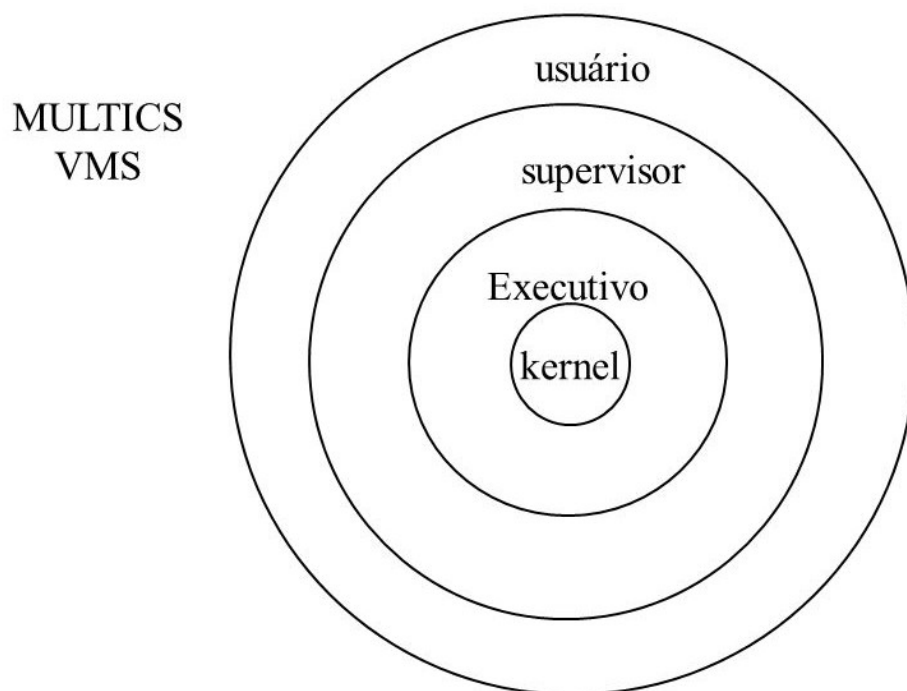


Figura 2: Estrutura dos sistemas em camadas.

Sistemas micro-núcleo

Uma outra possibilidade de estruturação consiste em retirar do núcleo todo o código de alto nível (normalmente associado às políticas de gerência de recursos), deixando no núcleo somente o código de baixo nível necessário para interagir com o hardware e criar as abstrações fundamentais (como a noção de atividade).

Sistemas micro-núcleo

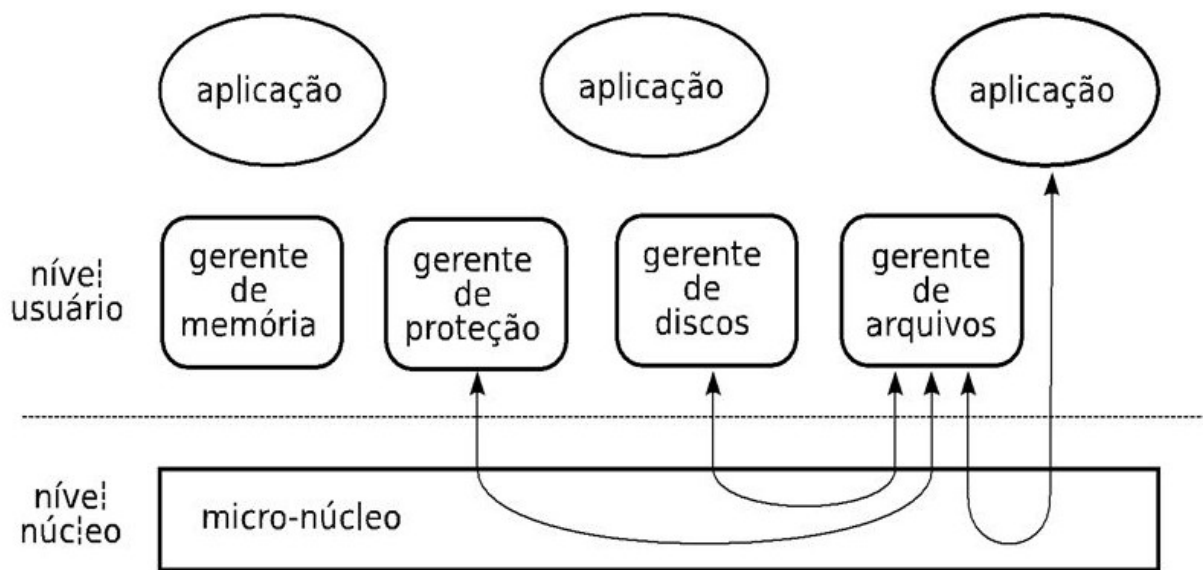


Figura 3: Estrutura e funcionamento de sistemas *microkernel*.

Neste tipo de sistema, as funcionalidades são quase todas executadas fora do núcleo, em oposição a um núcleo monolítico. Os processos se comunicam com um núcleo mínimo, usando o mínimo possível o "espaço do sistema" (kernel space). Neste local os aplicativos tem acesso a todas as instruções e a todo o hardware e deixando o máximo de recursos rodando no "espaço do usuário" (user-space) em que o software tem algumas restrições, não podendo acessar algumas hardwares, nem tem acesso a todas as instruções).

Sistemas Cliente-Servidor

Sistemas Cliente-Servidor são modelos de computação que distinguem dois tipos básicos de equipamentos computacionais: servidores e clientes, sendo interligados entre si geralmente utilizando-se uma rede de computadores. Neste modelo, geralmente os servidores agregam as funções mais importantes do sistema, deixando aos clientes apenas o processamento de aplicações mais básicas.

As principais características deste tipo de sistema são:

- Elevar a camada onde são implementadas as funções normalmente efetuadas pelo sistema operacional;
- Reduzir as funções do sistema operacional;
- Tornar menor e de mais fácil a manutenção cada parte do sistema operacional.

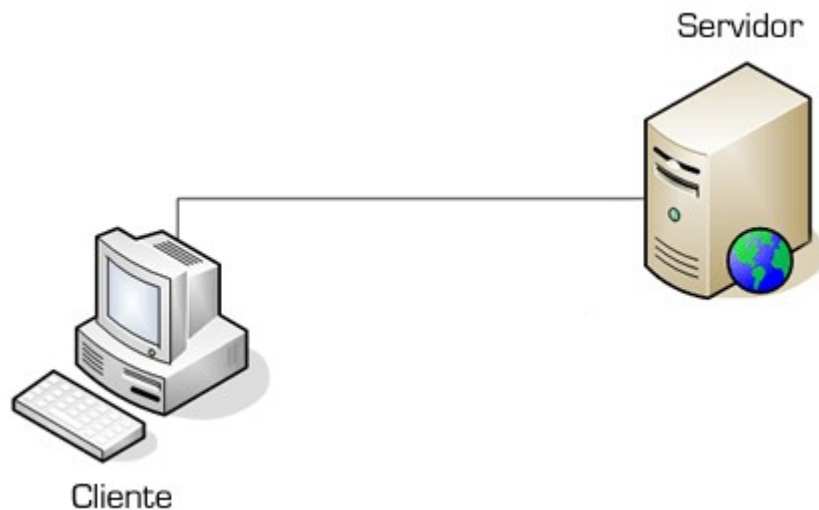


Figura 4: Funcionamento de sistemas Cliente-Servidor.

Exercícios

1. Qual a principal desvantagem de sistemas monolíticos?
2. Qual a diferença entre sistemas Cliente-Servidor e Sistemas micronúcleo?
3. Qual a principal vantagem da abordagem em camadas para o projeto de um sistema?
4. Qual a principal vantagem da abordagem do *microkernel* para o projeto de um sistema?
5. Em uma farmácia, um computador cuida de gerenciar todas as vendas, e as demais máquinas são apenas terminais de acesso as vendas. Qual o melhor tipo de sistema que deve ser adota na farmácia?