

ESTRUTURA DE UM SISTEMA OPERACIONAL

**PROF. ME.
HÉLIO
ESPERIDIÃO**

VISÃO DO USUÁRIO DE UM SO

- Imagem que o usuário tem do sistema
- Interface para acesso aos recursos.

EXECUTAR PROGRAMAS

- Todo sistema operacional oferece meios para programas serem carregados na memória principal e serem executados.
 - O SO recebe o nome do arquivo.
 - Aloca espaço na memória.
 - Copia o conteúdo do arquivo para memória.
 - E finalmente inicia sua execução.

KERNEL (NÚCLEO)

- Os principais componentes do kernel de qualquer sistema operacional moderno são.
 - Gerencia de processador.
 - Gerencia de memória.
 - Sistemas de arquivo
 - Gerencia de entrada e saída.

TIPOS DE KERNEL

KERNEL MONOLÍTICO

- Kernel monolítico contém todos os sub-sistemas em um único executável binário.
- Andrew S.Tanenbaum os define como “A grande bagunça”.
- Drivers, serviços e tudo mais estão contidos neste mesmo binário.
- Imagine um único arquivo em C que implementasse um sistema operacional inteiro.

KERNEL MONOLÍTICO

- **Vantagens**

- Normalmente um kernel monolítico tem um desempenho melhor que outras formas de desenho de kernel.

- **Desvantagens**

- A manutenção do kernel em nível de programação é muito prejudicada pois o kernel deve ser recompilado e substituído por completo para ter acesso a um novo recurso do kernel

- **Exemplos de Kernels Monolíticos**

- FreeDOS, MS-DOS, Unixes antigos, Linux (quando compilados como kernel monolítico), ModulOS, entre outros.

MICROKERNELS

- Os microkernels são kernels que tem um “framework” básico que através de troca de mensagens, “conversa” como os sub-sistemas que estão em nível de usuário, funcionando como programas normais.
- Por exemplo, o sub-sistema que implementa um sistema de arquivos é executado como se fosse um programa qualquer.

MICROKERNEL

- **Vantagens**

- O **Microkernel** é modular, cada sub-sistema do microkernel (gerenciador de arquivos, memória, etc) funciona como um módulo e por isso o microkernel é extremamente flexível.
 - Ou seja cada modulo não faz parte direta do kernel.
- Um microkernel pode ter um executável binário menor que um kernel monolítico por que o microkernel permite a utilização Módulos que podem ser usados de acordo com as necessidades do projeto.

- **Desvantagens**

- O microkernel tem um desempenho menor comparado com um **kernel monolítico** justamente pelo fato de fazer um uso muito intenso de troca de mensagens entre os módulos.

- **Exemplos de Microkernels**

- Windows NT (incluindo Windows 2000,
- Windows XP e Windows 2003), QNX, Fiasco, L4, Minix

CHAMADAS DE SISTEMA

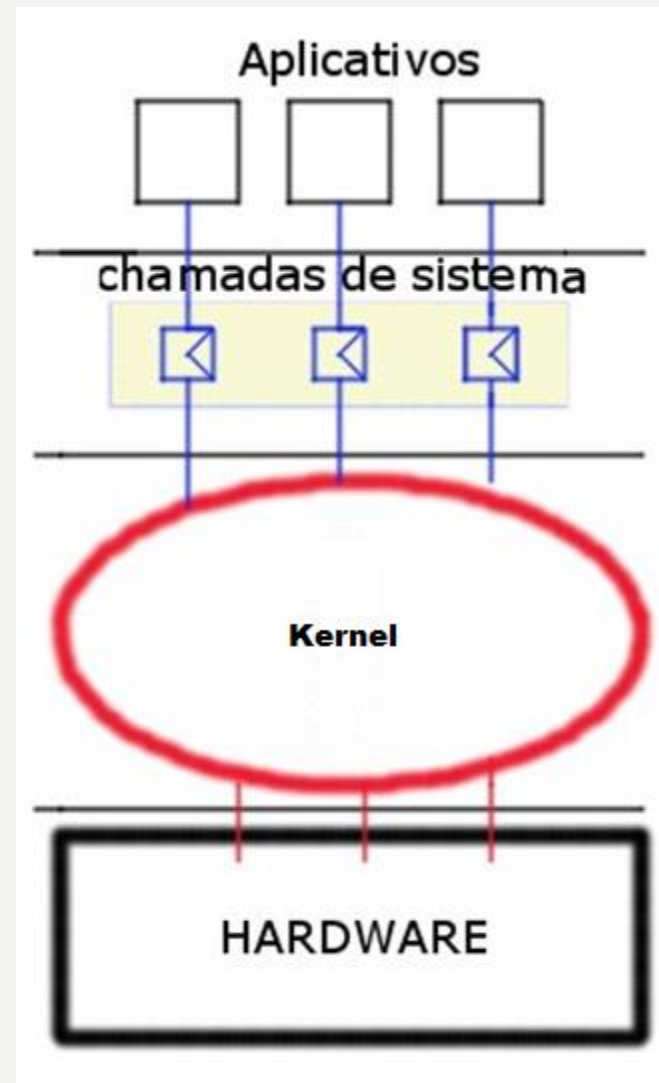
- Podemos definir o conjunto das Chamadas de Sistema como a interface entre o Sistema Operacional e seus programas aplicativos.

CHAMADAS DE SISTEMA

- Os programas solicitam serviços para o SO por meio de chamadas de sistema.
 - As chamadas de sistema transferem a execução do SO.
 - O SO trabalha de duas formas:
 - Modo usuário e modo supervisor(Kernel).
 - O modo usuário não acessa de forma direta os dispositivos de hardware.
 - O modo Kernel é como se fosse um super usuário, capaz de gerenciar as tarefas mais complexas associadas a um SO.

CHAMADAS DE SISTEMA

- Um programa desenvolvido em C para ler um arquivo.
- O seu programa pode ser considerado o aplicativo.
- Quando o programa deseja abrir o arquivo, o programa faz uma solicitação para o SO. O SO por meio do kernel abre o arquivo e apresenta para o aplicativo que segue sua execução.
- Nenhum programa acessa de forma direta o hardware.
- O hardware só é acessado por meio de chamadas de sistema.



PROGRAMAS DE SISTEMA

- São programas normais, também chamado de utilitários, executados fora do *kernel do SO* que utilizam as chamadas de sistema para realizarem suas funções.
- Exemplos: *Windows Explorer, Interpretador de Comandos (DOS ou UNIX) ou Interface Gráfica de Usuário (Windows)*.

SISTEMA OPERACIONAL NA VISÃO DE PROJETO

- Na visão de projeto a preocupação é com a organização interna do SO, ou seja, como os serviços são implementados.
- Na solicitação de algum serviço corresponde a uma chamada de sistema que deverá ser tratada pelo SO quais serão as atividades envolvidas no processo e como serão implementadas.
- Exemplo: na leitura de um registro de arquivo em disco o SO envia comandos à controladora do disco e o programa espera a resposta.
- Concluída a leitura o SO é informado pela controladora e o programa pode retornar sua execução normal