

Sistemas Operacionais I

Estrutura Interna do MINIX 3

Minix 3

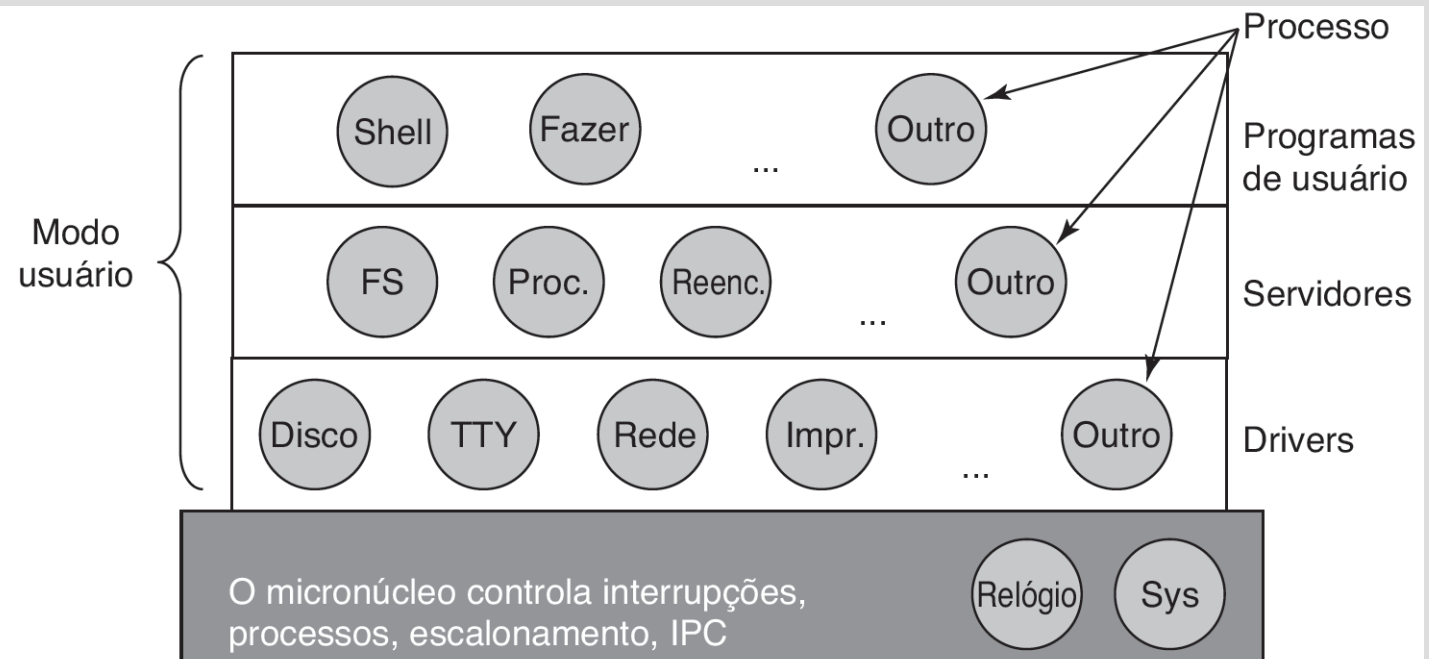
Diferente do UNIX (Linux), em que o kernel é monolítico, o MINIX 3 é baseado em uma estrutura do tipo micro-kernel.

A comunicação entre processos se dá através de troca de mensagens.

Esse design permite uma estrutura mais flexível e modular.

Camadas do MINIX 3

MINIX 3 é estruturado em quatro camadas;
Cada camada tem uma função bem definida;
As camadas podem ser vistas abaixo:



■ **Figura 1.23** Estrutura do sistema MINIX 3.

Camada 1 - kernel

Escalona processos;

Gerência as transições entre os estados pronto, bloqueado e em execução;

Manipula todas as trocas de mensagens entre processos:

- validação do destino

- localização dos buffers de send e receive na memória física

- cópia de bytes do sender para o receiver

Oferece suporte para acesso a portas de I/O e interrupções (modo kernel);

Camada 1 - clock task

Além do kernel, inclui dois módulos cujas funções são similares a device drivers

Clock task é um dispositivo de I/O, que interage com o hardware que gera o sinal de temporização. Interface com o kernel apenas.

Camada 1 - system task

Uma das principais funções da camada 1 é prover chamadas de kernel para camadas superiores

System Task implementa essas chamadas, entre as quais estão:

- Leitura e escrita em portas I/O

- Cópia de dados entre espaços de endereço

Camada 1

System Task e Clock Task são compilados em espaço de end. Kernel, mas possuem call stacks próprias:

Apesar de ser quase todo escrito em C, uma pequena parte do kernel está em Assembly, entre elas:

- Manipulação de interrupção

- Trocas de contextos de processos

- Manipulação da MMU

O porte requer reescrita do código Assembly

Demais Camadas

O kernel trata as três camadas restantes da mesma forma e poderiam se classificadas como uma só;

Cada processo nas camadas acima é limitado a instruções em modo usuário e é escalonado pelo kernel;

Nenhuma delas acessa um segmento de memória fora do atribuído a si;

Processos na camada 2 tem mais privilégios do que os da camada 3 e assim por diante;

Camada 2

A camada 2 inclui os drivers do MINIX 3

Um driver é necessário para cada tipo de dispositivo

Device Drivers podem fazer chamadas ao kernel:

Ex: Transferência de dados entre diferentes processos

Device Drivers, tipicamente iniciam com o sistema, mas podem iniciar posteriormente;

Camada 3

Contém os servers(servidores), que provém serviços úteis aos processos de usuário;

Dois servers são essenciais FS e PM

FS: responsável pelas chamadas de sistema do sistema de arquivos, como read, mount, and chdir.

PM: responsável pelos chamadas de sistema que envolvam processos, como fork e exec, assim como também pelos sinais, alarm e kill. Além de gerenciar a memória;

Camada 3

Além dos dois citados tem-se outros servers específicos para o MINIX 3:

- IS (Information Server): prove informações sobre outros drivers e servers;

- RS (Reincarnation Server): inicia e se preciso reinicia drivers que não foram carregados pra memória

- Inet: servidor de rede

Servers não acessam I/O diretamente, somente através dos drivers;

Servers podem se comunicar com o kernel através do System Task

Camada 3

Chamadas kernel são diferentes de chamadas de sistema POSIX;

Chamadas kernel são de baixo nível e são providas pelo System Task para as camadas superiores;

Programas de usuário usam chamadas POSIX e não chamadas kernel;

Chamadas Kernel podem ser consideradas um caso especial de chamadas de sistema

Camada 3

O sistema não precisa ser recompilado para adição de funcionalidades e de novos servers;

Drivers e Servers nas camadas 2 e 3 podem ser chamados de processos do sistema;

Processos do sistema fazem parte do SO;

Processos do sistema tem maior prioridade do que processos de usuário;

Camada 4

Composto de processos do usuário

Shells, compiladores, são exemplos de processos de usuário

Geralmente existem enquanto o usuário estiver logado, entretanto alguns seguem rodando mesmo depois que o usuário desloga, por exemplo o Init;