

Sistemas operacionais: Conceitos e Mecanismos

Capítulo 3: Arquitetura de SOs

Integrantes do grupo: Jefferson Botitano e Leonardo Ludvig
Professor: Arliones Stevert Hoeller Junior
Disciplina: Sistemas operacionais

Arquitetura de SOs

A arquitetura para cada sistema operacional é distinta para cada um, mesmo tendo uma arquitetura básica que se tem de organizar um sistema operacional como realizar o procedimento de isolamento do núcleo, modularização, desempenho e segurança

Sistemas monolíticos

Em um sistema operacional que é do tipo monolítico seu código funciona em modo núcleo que concede acesso a todos os recursos que o hardware disponibiliza e não impõe restrições de acesso à memória, sendo assim seus componentes internos do sistema operacional podem se complementar entre eles de acordo com seu uso.

A vantagem de um sistema do tipo monolítico é que seu desempenho é muito bom pelo fato de todo componente do núcleo ter acesso aos outros componentes sendo áreas de memória ou componentes periféricos, o resultado da interação entre esses componentes do sistema leva ele a ser mais rápido e compacto pois como não existe necessidade realizar comunicações específicas (chamadas de função) entre os componentes do núcleo.

A desvantagem do sistema monolítico é que como todos os componentes estão com acesso a qualquer componente que o hardware do sistema operacional oferece, caso exista uma maneira incorreta de uso de um componente levando a exceções que o sistema possa disparar isso impactará em uma propagação de erro pelo sistema como um todo e resultando em colapso do sistema como travamentos, reinicializações ou funcionamentos erráticos.

Sistemas micronúcleo

O sistema operacional micronúcleo é baseado em separar do núcleo todo o código de alto nível, que geralmente são abstrações de recursos que estão disponíveis, sendo assim no núcleo opera apenas códigos de baixo nível para interação com o hardware do sistema e quando necessário algumas abstrações de tipo básico. O monólito do sistema operacional anterior ele ainda existe só que o seu tamanho é reduzido e dentro desse bloco os componentes se comunicam por chamadas de função.

A vantagem do micronúcleo é que dispões de uma modularidade elevada, sendo que cada serviço pode ser desenvolvido sem depender dos demais gerando assim uma flexibilidade, onde um serviço pode ser carregado e desativado se necessário, caso um destes serviços resulte em falha somente ele falhará e não propagar seu erro pelo sistema pelo fato que cada memória é separada para cada serviço.

Como um sistema de micronúcleo cria apenas a noção de tarefa, a separação de memória para cada serviço e sua comunicação entre eles juntamente com operações que liberam acesso às portas in/out que são de acesso para dispositivos, impacta que tarefas de alto nível, como regras de como usar o processador, memória, arquivos, manipulação de acesso aos recursos disponível do sistema são criados fora do núcleo em tarefas que se comunicam usando um intermediário que é feito pelo núcleo do sistema. Exemplo o driver do disco está no micronúcleo mas a implementação do sistema do arquivo está fora deste micronúcleo e na hora que se quer escrever ou ler no disco ele faz uma chamada para o micronúcleo para realizar a ação requisitada.

Sistemas em camadas

A organização do sistema operacional do tipo em camadas é feita em camadas, dividida em três camadas com diferentes tipos de privilégios:

Camada baixa: responsável por realizar a interface com o hardware, sendo assim seu nível de privilégio é de acesso total ao hardware.

Camada intermediária: responsável pelo gerenciamento sofisticado de abstração.

Camada superior: responsável por definir a interface do núcleo para as chamadas de sistemas quando necessário por uma aplicação.

A desvantagem deste tipo de sistema é que como o empilhamento de várias camadas de software do sistema faz com que cada requisição de um serviço resulte em um tempo muito longo até chegar o componente periférico ou quando existe necessidade de acessar algum recurso de hardware. Com a divisão de utilidades do sistema resulta em algumas delas serem independentes, sendo assim não seria possível realizar a organização em camadas para estas utilidades, como por exemplo o gerenciamento de alocação, entrada/saída de memória para escrita e leitura de dados com o acréscimo de uma gestão de uma memória que necessita implementar uma paginação de disco resultando em um conflito de nível de camada.

Sistemas híbridos

A combinação da base de dois sistemas operacionais como é o caso do sistema operacional do tipo monolítico mais o do sistema micronúcleo resulta no sistema do tipo híbrido. A implementação da divisão dentro dos sistemas operacionais do tipo híbrido é comum que ajuda os a realizar a divisão tarefas com seus níveis de privilégios somados com um bom desempenho que vêm do núcleo do tipo monolítico que juntamente com as características do núcleo tipo micronúcleo traz de volta componentes segurança para o sistema.

Arquiteturas avançadas

Recentemente, além das arquiteturas clássicas apresentadas, surgiram diversas propostas para organizar os componentes e recursos de um sistema operacional, geralmente desenvolvidos para algum uso específico como nuvens computacionais ou uso de linguagens de programação. Algumas dessas arquiteturas são.

- Máquinas virtuais: Utilizadas para simular um sistema operacional convidado em cima de um sistema hospedeiro como o VirtualBox, ou uma aplicação para uma linguagem específica de programação como o JVM.
- Contêineres: São formas de isolar uma aplicação ou subsistemas reservando um espaço dedicado do sistema chamado de domínio alocando recursos como memória, tempo de processador e espaço em disco.
- Sistemas exonúcleo: Como o nome sugere essa arquitetura tenta remover todas as abstrações do núcleo deixando ele apenas para liberar os acessos.
- Sistemas uninúcleo: É um núcleo monolítico compacto que executa de forma individual suas aplicações, como em sistemas de nuvem rodando várias máquinas virtuais para executar aplicação diversas de diferentes usuários.

O linux possui um núcleo similar com o da figura 3.1 do livro-texto, mas também possui “tarefas de núcleo” que executam como os gerentes da figura 3.2 do livro-texto. Seu núcleo é monolítico ou micronúcleo? Por quê?

O núcleo do linux possui a arquitetura monolítica, ele foi construído originalmente como um sistema puramente monolítico, mas ao longo de suas versões foi sendo gradativamente modularizado aderindo características de outras arquiteturas devido a limitações de um núcleo monolítico.

Sobre as afirmações a seguir, relativas às diversas arquiteturas de sistemas operacionais, indique quais são incorretas, justificando sua resposta:

- **Uma máquina virtual de sistema é construída para suportar uma aplicação escrita em uma linguagem de programação específica, como Java.**

Falso. Uma máquina virtual de sistema é destinada a simular Sistema Operacional convidado dentro de um espaço virtual e simular suas aplicações. O descrito seria a máquina virtual de aplicação.

- **Um hipervisor convidado executa sobre um sistema operacional hospedeiro.**

Verdadeiro.

- **Em um sistema operacional micronúcleo, os diversos componentes do sistema são construídos como módulos interconectados executando dentro do núcleo.**

Falso. O propósito do micronúcleo é retirar todos os programas de alto do núcleo deixando a menor quantidade possível de operações dentro do mesmo, por isso o nome de micronúcleo.

- **Núcleos monolíticos são muito utilizados devido à sua robustez e facilidade de manutenção.**

Falso, um sistema monolítico sofre justamente com essas características pois seus recursos são todos interdependentes tornando qualquer alteração ou manutenção no núcleo mais complexa e com resultados inesperados em outras áreas do mesmo.

- **Em um sistema operacional micronúcleo, as chamadas de sistema são implementadas através de trocas de mensagens.**

Verdadeiro.

O sistema operacional Minix 3 usa uma arquitetura micronúcleo. Ele pode ser obtido gratuitamente no site <http://www.minix3.org>. Instale-o em uma máquina virtual e explore seus processos, usando os comandos `top` e `ps`. Identifique os principais processos em execução, usando a documentação do site.

```
load averages: 0.02, 0.00, 0.00
45 processes: 1 running, 44 sleeping
main memory: 1048124K total, 1006324K free, 996796K contig free, 17992K cached
CPU states: 0.39% user, 0.35% system, 0.43% kernel, 98.84% idle
CPU time displayed ('t' to cycle): user ; sort order ('o' to cycle): cpu
```

PID	USERNAME	PRI	NICE	SIZE	STATE	TIME	CPU	COMMAND
-1	root	0		2802K		0:00	0.43%	kernel
9	root	1	0	180K		0:00	0.35%	tty
7	root	5	0	1200K		0:00	0.17%	vfs
11	root	2	0	4744K		0:00	0.09%	vm
49	service	5	0	7740K		0:00	0.03%	mfs
40	root	7	0	1200K	RUN	0:00	0.03%	procfs
107	root	7	0	188K		0:00	0.02%	devmand
228	root	7	0	568K		0:00	0.02%	top
79	root	7	0	200K		0:00	0.02%	devman
5	root	4	0	596K		0:00	0.01%	pm
164	root	7	0	104K		0:00	0.01%	vbox
4	root	4	0	1200K		0:00	0.00%	rs
118	service	7	0	132K		0:00	0.00%	random
8	root	3	0	116K		0:00	0.00%	memory
6	root	4	0	48K		0:00	0.00%	sched
3	root	4	0	156K		0:00	0.00%	ds

- **Kernel**
- **tty**: porta serial, pty, etc.
- **vfs**: virtual file system
- **vm**: virtual memory
- **top**
- **vbox**: conjunto de drivers que o está rodando pelo virtualbox
- **pm**: controle de energia
- **rs**: servidor de reencarnação
- **sched**: escalonador

```
# ps
PID TTY TIME CMD
184 co 0:00 -sh
185 c1 0:00 /usr/libexec/getty default tttyc1
186 c2 0:00 /usr/libexec/getty default tttyc2
187 c3 0:00 /usr/libexec/getty default tttyc3
194 co 0:00 ps
```