

Universidade Federal de Roraima

Descrição do funcionamento de um kernel

João Lucas Sidney Rodrigues e Luis Henrique Matos Salles

Professor: Hebert

O que é o kernel?

Um kernel é o nível mais baixo de software facilmente substituível que faz interface com o hardware do seu computador. Ele é responsável por fazer a interface de todos os seus aplicativos que estão sendo executados no “modo de usuário” até o hardware físico e permitir que os processos, conhecidos como servidores, obtenham informações uns dos outros usando comunicação entre processos (IPC).

Como o kernel funciona

Existem diferentes maneiras e considerações arquitetônicas ao construir um kernel do zero. A maioria dos kernels se enquadra em um dos três tipos: monolítico, microkernel e híbrido. O Linux é um kernel monolítico, enquanto o OS X (XNU) e o Windows 7 usam kernels híbridos. Vamos fazer um tour rápido pelas três categorias para que possamos entrar em mais detalhes mais tarde.

Microkernel

Um microkernel age de forma à gerenciar apenas o que é necessário: CPU, memória e IPC. Quase todo o resto no computador pode ser visto como um acessório a ser manipulado no modo de usuário. Sua grande vantagem é portabilidade pois não é necessário se preocupar após trocar uma placa de vídeo ou mesmo o sistema operacional, desde que este ainda tente acessar o hardware da mesma forma. Os microkernels também ocupam muito pouco espaço, tanto de memória quanto de de instalação, e tendem a ser mais seguros porque apenas processos específicos são executados no modo de usuário que não tem permissões altas como no modo supervisor.

Kernel Monolítico

Os kernels monolíticos são o oposto dos microkernels porque abrangem não apenas a CPU, a memória e o IPC, mas também incluem coisas como drivers de dispositivo, gerenciamento do sistema de arquivos e chamadas do servidor do sistema. Kernels monolíticos tendem a ser melhores no acesso a hardware e multitarefa porque se um programa precisa obter informações da memória ou outro processo em execução, ele tem uma linha mais direta para acessá-lo e não precisa esperar em uma fila para fazer as coisas. No entanto, isso pode causar problemas porque quanto mais coisas rodam no modo supervisor, mais coisas podem derrubar o seu sistema se ele não se comportar corretamente.

Kernel Híbrido

Kernels híbridos têm a capacidade de escolher se desejam executar algo no modo de usuário ou no modo supervisor. Algo como drivers de dispositivos e E / S do sistema de arquivos serão executados no modo de usuário, enquanto o IPC e as chamadas do servidor ficam no modo supervisor. Isso oferece o melhor dos dois mundos, mas geralmente exigirá mais trabalho do fabricante do hardware porque toda a responsabilidade do driver é dela. Ele também pode ter alguns dos problemas de latência inerentes aos microkernels.

Arquitetura do Kernel Linux

Por ser monolítico, o kernel do linux tem maior complexidade sobre os outros tipos de kernels. Este foi um recurso de design que estava sob muito debate nos primeiros dias do Linux e ainda carrega algumas das velhas falhas de design que os kernels monolíticos costumam ter.

Uma ação que os desenvolvedores do kernel Linux tomaram para contornar essas falhas foi criar módulos do kernel que pudessem ser carregados e descarregados em tempo de execução, o que significa que o usuário pode adicionar ou remover recursos do kernel na hora. Mais do que apenas adicionar funcionalidade de hardware ao kernel, é possível incluir módulos que executam processos de servidor, como virtualização de baixo nível e também permitir que todo o kernel seja substituído sem a necessidade de reiniciar o computador em alguns casos.

Os arquivos do kernel do Linux?

O arquivo do kernel, no Ubuntu, é armazenado na sua pasta / boot e é chamado de vmlinuz-version. O nome vmlinuz vem do mundo unix, onde costumavam chamar seus kernels simplesmente de "unix" nos anos 60, então o Linux começou a chamar seu kernel de "linux" quando foi desenvolvido pela primeira vez nos anos 90.

Quando a memória virtual foi desenvolvida para habilidades multitarefa mais fáceis, "vm" foi colocado na frente do arquivo para mostrar que o kernel oferece suporte à memória virtual. Por um tempo, o kernel do Linux foi chamado de vmlinux, mas ficou muito grande para caber na memória de inicialização disponível, então a imagem do kernel foi compactada e o x final foi alterado para z para mostrar que estava compactado com formato zlib. Essa compressão nem sempre é usada, muitas vezes substituída por LZMA ou BZIP2.

A numeração da versão estará no formato A.B.C.D em que A.B provavelmente será 2.6, C será sua versão e D indica seus patches ou correções.

Na pasta / boot também haverá outros arquivos muito importantes chamados initrd.img-version, system.map-version e config-version. O arquivo initrd é usado como um pequeno disco RAM que extrai e executa o arquivo kernel real. O arquivo system.map é usado para gerenciamento de memória antes que o kernel carregue totalmente, e o arquivo de configuração diz ao kernel quais opções e módulos carregar na imagem do kernel quando ele estiver sendo compilado.

Módulos do Kernel

Módulos de kernel, também conhecidos como módulo de kernel carregável (LKM), são essenciais para mantê-lo funcionando com todo o seu hardware sem consumir toda a memória disponível.

Um módulo normalmente adiciona funcionalidade ao kernel base para coisas como dispositivos, sistemas de arquivos e chamadas de sistema. Os LKMs têm a extensão de arquivo .ko e são normalmente armazenados no diretório / lib / modules. Por causa de sua natureza modular, você pode facilmente personalizar seu kernel configurando módulos para carregar, ou não carregar,

durante a inicialização com o comando `menuconfig` ou editando seu arquivo `/boot / config`, ou você pode carregar e descarregar módulos rapidamente com o `modprobe` comando.

Módulos de terceiros e de código fechado estão disponíveis em algumas distribuições, como o Ubuntu, e podem não ser instalados por padrão porque o código-fonte dos módulos não está disponível. O desenvolvedor do software (ou seja, nVidia, ATI, entre outros) não fornece o código-fonte, mas sim constrói seus próprios módulos e compila os arquivos `.ko` necessários para distribuição.

Referencias

<https://www.certificacaolinux.com.br/como-funciona-o-kernel-do-linux/>

<https://www.linux.org/threads/understanding-the-linux-kernel.12508/>

<https://www.kernel.org/doc/html/latest/admin-guide/mm/index.html>