

MÓDULOS DO KERNEL

O Kernel do Linux tem um design modular. Isso permite que tenhamos um Kernel enxuto, somente com os recursos necessários habilitados. Caso precisemos de outros recursos, normalmente drivers, podemos carregar um módulo correspondente dinamicamente. Esse carregamento pode ser feito durante o boot do kernel, caso o módulo seja adicionado no ram disk inicial (initrd), ou após o boot do kernel, por meio dos scripts de inicialização.

Para adicionar um módulo do Kernel com o sistema em execução usamos o seguinte comando:

```
modprobe <módulo>
```

Para verificar os módulos carregados:

```
lsmod
```

Caso precisemos que esse módulo seja carregado a cada boot, podemos criar um script de SysV init, colocar o comando no “rc.local” ou ainda usar algum mecanismo específico da distribuição. Nos Red hat like, por exemplo, existe o diretório */etc/sysconfig/modules/*, onde criamos scripts executáveis com o nome *<meumodulo>.modules*, onde podemos colocar o comando *modprobe* e/ou alguns comandos auxiliares, como uma verificação se determinado recurso de hardware existe e/ou está disponível, por exemplo.

Personalizar o ramdisk inicial pra que ele carregue o módulo durante o boot do Kernel é um recurso que podemos usar para carregar módulos necessários ao processo de boot do próprio kernel, como drivers SCSI, por exemplo. Para isso, podemos usar o utilitário *mkinitrd*:

```
mkinitrd --with=<modulo> </boot/meuinit.img> $(uname -r)
```

Onde *</boot/meuinit.img>* indica o path e o nome do novo initrd gerado e o *\$(uname -r)* serve para informar o kernel que servirá de base.

O problema dessa solução é que a cada update de kernel precisaremos gerar um novo ramdisk com os nossos módulos. Isso pode ser resolvido usando o arquivo */etc/mkinitrd/modules*.

O SISTEMA DE ARQUIVOS VIRTUAL /proc

O */proc* é um sistema de arquivos virtual que oferece informações detalhadas sobre o kernel, hardware e processos em execução. A maioria dos arquivos do */proc* podem ser visualizados com o comando *cat*. Em alguns casos, o comando *strings* se torna a melhor opção devido ao uso de caracteres nulos. Em outros casos, o conteúdo dos arquivos é quase indecifrável. Isso ocorre pela falta de padronização na forma de exibição das informações do */proc*.

Algumas entradas interessantes do */proc*:

```
/proc/<PID> → Informações sobre um processo em execução (ro).  
/proc/cmdline → Linha com opções usadas para o boot do kernel (ro).  
/proc/cpuinfo → Detalhes sobre os processadores (ro).  
/proc/mdstat → Informações sobre as partições RAID gerenciadas pelo mdadm (ro).  
/proc/meminfo → Detalhes sobre uso da memória RAM e swap (ro).  
/proc/swaps → Partições e/ou arquivos usados para swap (ro).  
/proc/modules → Informações sobre módulos em uso pelo kernel (ro).  
/proc/mounts → Detalhes sobre sistemas de arquivos montados (ro).  
/proc/net/ → Diretório com informações relacionadas a rede (ro).  
/proc/partitions → Dados sobre partições ativas (ro).  
/proc/version → Detalhes sobre versão do kernel e sistema operacional (ro).  
/proc/sys/kernel/hostname → Nome do computador (rw).  
/proc/sys/net/ipv4/ip_forward → Estado do roteamento (rw).  
/proc/sys/vm/drop_caches → Efetua o descarte do cache da RAM (rw).  
/proc/sys/vm/swappiness → Nível (de 0 a 100) de propensão do SO em usar a swap (rw).
```

É comum o uso de ferramentas para visualizar e manipular os arquivos do */proc*. Por exemplo, o *lspci* e o *lsusb* sumariza e torna legível as informações sobre dispositivos conectados.

O COMANDO sysctl

O comando *sysctl* é usado para ver, configurar e automatizar configurações do kernel no diretório virtual */proc/sys*.

Para ver todas as configurações possíveis do *sysctl*, execute:

sysctl -a

Essas informações são as mesmas obtidas visualizando cada arquivo individualmente. A mesma hierarquia é mantida, com apenas algumas diferenças. Em vez de barras, usa-se pontos para separar os níveis. No *sysctl*, a estrutura considera o */proc/sys* como sendo a raiz, assim, tudo é representado a partir desse ponto. Por exemplo, o caminho *kernel.hostname* representa o path real */proc/sys/kernel/hostname*.

O comando *sysctl* pode ser usado para substituir o *echo* na atribuição de valores para os arquivos.

Observe o exemplo de atribuição usando o comando *echo*:

echo 1 > /proc/sys/net/ipv4/ip_forward

Equivalente com *sysctl*:

sysctl -w net.ipv4.ip_forward = 1

Ou ainda, você pode adicionar a seguinte linha no arquivo */etc/sysctl.conf*:

net.ipv4.ip_forward = 1

E carregar todas as configurações que estão no arquivo com o comando:

sysctl -p

O arquivo */etc/sysctl.conf* será recarregado a cada boot. Isso faz com que todas as configurações presentes nele sejam persistentes.