

Capítulo 13 - Módulos

13.1. Objetivos :

Entender como trabalhar com módulos no Linux.

A primeira coisa que é preciso pensar quando se fala em dispositivos no Linux, é listá-los:



```
#lspci
```

```
01:00.0 VGA compatible controller: nVidia Corporation GeForce 8500 GT  
(rev a1)
```

```
03:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd.  
RTL8101E/RTL8102E PCI Express Fast Ethernet controller (rev 01)
```

```
04:01.0 Ethernet controller: Atheros Communications Inc.  
AR5212/AR5213 Multiprotocol MAC/baseband processor (rev 01)
```

O comando lspci irá imprimir na tela as informações sobre meus dispositivos:



```
#lspci -v | more
```

O -v é verbose para mostrar as informações em detalhes. Aqui e mostrado a relação dos dispositivos que é necessário configurar, tais como placa de rede, placa de som, vídeo etc.



Dica: Para você saber mais informações sobre todo o seu hardware, você pode instalar um pacote chamado lshw.



```
# aptitude install lshw
```

Para usá-lo:



```
# lshw
```

Quando é instalado uma distribuição como, por exemplo, Debian, RedHat, Suse, Slackware, dentre outras, esta sendo utilizando um kernel que foi compilado pelos desenvolvedores da distribuição.

Como esse kernel deve ser capaz de rodar em praticamente qualquer PC e dar suporte a qualquer tipo de recurso que o usuário pretenda utilizar, o desenvolvedor compila um kernel que fornece todas as funcionalidades básicas e, em separado, compila pedaços de código que dão suporte a funcionalidades mais específicas.

Esses pedaços de código são os chamados módulos. Dessa forma, quando o sistema é carregado um kernel básico se coloca na memória e passa a controlar a máquina.

Neste ponto são verificadas outras funcionalidades que se espera que o kernel dê suporte, como por exemplo utilizar uma partição XFS; neste momento, se o kernel não possuir suporte nativo a esse file system ele irá verificar se o módulo que dá suporte a ele foi compilado.

Se esse módulo for encontrado, ele será carregado expandindo as funcionalidades do kernel. Para descobrir qual a versão do kernel está sendo usada:



```
# uname -r
```

Antes de começar a configurar qualquer dispositivo, veja as explicações de alguns comandos para a manipulação de módulos.

Um módulo é built-in quando o mesmo já está incorporado no kernel do sistema, ou seja, assim que o kernel for iniciado, o dispositivo também será, caso a máquina o tenha.

Agora os módulos que não são built-in , ou seja, que não são carregados quando o sistema inicia, significa que o kernel pode suportar, mas o mesmo precisa ser levantado manualmente.

Então, módulo é como lego, pode estar ali, mas para funcionar, você precisa montar, ou melhor carregar. E é justamente aí que entra a sua função. Já os built-in são transparentes para os usuários, não é necessário se preocupar com eles pois já estão no kernel.

Por exemplo: no Debian as placas de rede Realtek são built-in, então depois que o sistema liga, já é possível digitar:



```
# ifconfig -a
```

Com esse comando é possível ver que a mesma já está funcionando, com isso é necessário apenas colocar o IP.



Lembrando que só é possível aplicar um ip na placa de rede se a mesma já estiver com seu módulo levantado (ou fazendo uma analogia com o Windows, o driver configurado).

Já no Red Hat essa mesma placa não vem como módulo built-in, então é preciso levantar o módulo. Por isso que compilar um kernel é muito importante, mas também exige conhecimento de hardware e de kernel.



Lembrando que as distribuições trazem um kernel genérico, e você deverá customizá-lo especificamente para a sua máquina, mas para isso haja Kernel Panic.

Então se o módulo é built-in não é necessário ter trabalho nenhum. Agora se não for built-in, terá que levantar o módulo com os comandos a seguir.

O comando lspci irá listar todos os dispositivos, mesmo que este já tenha o módulo built-in no kernel. O demais comandos que vai ser mostrado mas a frente não se aplica para módulos built-in (afinal de contas eles já foram carregados).

Os comandos mostrados agora são para manipulação de módulos. Depois que foi listado nossos dispositivos com o comando lspci .

Para listar todos os módulos ativos no meu sistema:



```
# lsmod  
ou  
# cat /proc/modules
```

Esse comando tem a função de listar todos os módulos que estão ativos no sistema, mesmo que o módulo não esteja sendo usado no momento. O próximo comando essencial nessa parte de módulos é o modprobe:



```
#modprobe nome_módulo
```

Ele é o responsável por ativar/levantar um módulo. É através dele que se habilita um determinado dispositivo, como por exemplo a placa de rede. Para ver quais os módulos estão compilados (disponíveis para uso):



```
# modprobe -l
```

O parâmetro é “l” de limão, e não é o número um.

Os módulos ficam no seguinte diretório:



```
# cd /lib/modules/$(uname -r)
```



É possível colocar um comando dentro de \$() quando se deseja que essa variável guarde como valor o resultado do comando. Se existir mais de um kernel compilado na máquina, logicamente haverá mais de um diretório em /lib/modules.

Você pode descobrir quais módulos tem disponíveis da seguinte forma:



```
# find /lib/modules/$(uname -r) -name '*.ko'
```

Uma coisa legal no Linux é que temos várias maneiras de fazer uma coisa só, aí você escolhe a que mais lhe agrada.



Lembre-se que é possível ter vários kernels disponíveis, mas só um funcionando ao mesmo tempo, pois isso é para se ter certeza que esta entrando no diretório que guarda os módulos do kernel que esta sendo utilizado no momento, aí é necessário digitar esse comando acima.

O modprobe levanta um módulo com a sua dependência, ou seja, se determinado módulo para ser iniciado precisa que um módulo em específico já esteja no ar, o modprobe faz isso sem lhe perguntar nada. Veja que prático!



Um exemplo prático é a placa de som, que para funcionar, além do módulo da placa em si, precisa do soundcore (módulo de som). Então, nesse caso o comando modprobe já resolveria essa dependência levantando o módulo da placa e também o soundcore.

Para determinar o número de módulos carregados:



```
# lsmod | grep -v ^"Module" | wc -l
```

Para determinar o número de módulos disponíveis:



```
# modprobe -l | wc -l
```

No caso tenho 1756 módulos disponíveis e 88 módulos carregados na memória (isso pode variar de sistema para sistema). Para levantar um módulo sem resolver sua dependência, lembre-se que nem todos os módulos tem dependências, pode ser usado o seguinte comando:



```
# insmod módulo
```

Se o módulo tiver dependência e for utilizado o comando insmod, ele acusará um erro. Agora, mostrando o inverso, se deseja derrubar um módulo, ou seja, desativá-lo:



```
#modprobe -r nome_do_módulo
```

E para remover um módulo sem Dependência:



```
#rmmod módulo
```



OBS : Lembre-se de que um módulo só pode ser desativado se o mesmo não estiver em uso!

Exemplo: só é possível derrubar o módulo da placa de rede, caso a mesma também esteja derrubada. Ou seja, sem funcionamento.



É válido saber que quando é dito remover, refere-se a tirar o módulo da memória e não deletá-lo do sistema!

Exemplo: Determine quais módulos são utilizados pelo dispositivo de cdrom:



```
# lsmod | grep cdrom  
cdrom 32544 1 sr_mod
```

Essa mensagem significa que o módulo cdrom é usado pelo módulo sr_mod.

32544 é o tamanho do módulo.

1 é quantidade de módulos que usam o módulo listado.

Tente remover o módulo cdrom utilizando o rmmod:



```
# rmmod cdrom  
ERROR: Module cdrom is in use by sr_mod
```



Não funcionou porque o módulo cdrom está sendo usado pelo sr_mod, e lembre-se que só é possível tirar um módulo da memória se ele não estiver em uso.

Remova o módulo sr_mod com rmmod já que ele está utilizando o módulo cdrom:



```
# rmmod sr_mod
```

O comando não vai retornar nenhuma mensagem, significa que ele tirou o módulo da memória. Verifique que o módulo ide_cd foi removido mas o cdrom ainda está carregado:



```
# lsmod | grep sr_mod
```

Não retornou nada, significa que não está mais na memória.



```
# lsmod | grep cdrom  
cdrom 32544 0
```

Retornou a linha do módulo cdrom, ele ainda continua na memória. Já que o módulo cdrom não está mais sendo utilizado podemos removê-lo:



```
# rmmod cdrom  
# lsmod | grep cdrom
```


Abra a bandeja do cdrom e verifique que o suporte ao CDROM foi carregado novamente:



```
# eject  
# lsmod |grep cdrom
```

Não foi!

Então, levante os módulos do cdrom novamente, só que com o comando insmod (para o comando insmod deve ser informado o caminho completo de onde está o módulo):



```
# modprobe -l cdrom  
/lib/modules/2.6.18-5-686/kernel/drivers/cdrom/cdrom.ko  
Ou:  
# insmod $(modprobe -l cdrom)
```

E depois:



```
# insmod $(modprobe -l sr_mod)
```

Verificando se os módulos levantaram:



```
# lsmod |grep cdrom  
cdrom 32544 1 sr_mod
```

Verifique se consegue usar o cdrom:



```
# eject
```

Puxar o cdrom:



```
# eject -t
```

Tire os módulos da memória novamente para ver que o modprobe é bem mais prático para usar:



```
# modprobe -r sr_mod
```

Tente ejetar o cd agora, você não conseguir. Levante os módulos agora:



```
# modprobe sr_mod
```

Veja que ele levantou o cdrom também:



```
# lsmod | grep cdrom  
cdrom 32544 1 sr_mod
```

Como o modprobe sabe quais módulos dependem de quais módulos:



```
# cd /lib/modules/$(uname -r)  
# ls -l  
# vi modules.dep
```

Esse arquivo é construído com o comando depmod (cobrado na LPI):



```
# depmod -a
```

Esses seriam os principais comandos que podem ser usado nessa parte de módulos. Sabendo os comandos, o diretório e os dispositivos, faça a configuração dos mesmos.

13.2. Exemplo de configuração da placa de som

Primeiro listaremos para ver o modelo:



```
# lspci | grep -i audio  
00:05.0 Multimedia audio controller: C-Media Electronics Inc CM8738 (rev 10)
```

No caso é uma C-Media. Então, vá até o diretório dos drivers e procure o módulo para a placa:



```
# cd /lib/modules/2.6.18-3-686/kernel/sound/pci
```



Agora vem a questão. Como descobrir qual o módulo exato para a placa de som?

Um comando que ajuda um pouco, mas ainda sim exige prática é o modinfo, veja um exemplo:



```
# modinfo cmpci  
filename: cmpci.o  
description: "CM8x38 Audio Driver"  
author: "ChenLi Tien, cltien@cmedia.com.tw"  
license: "GPL"
```

É possível perceber que o módulo diz: CM8x38 . Onde x será qualquer número nessa série. Então, CM8738 é o modelo que se encaixa nesse módulo. Nesse caso foi sendo testado arquivo por arquivo dentro do diretório com o modinfo.

No caso, sabe-se que o módulo para a Placa C-Media é cmpci , pois foi confirmado com o modinfo, então levante o módulo com o comando:



```
#modprobe cmpci
```



E vale lembrar que não foi colocado a extensão ".ko " quando for levantar o módulo com o comando!

Para ver o se módulo carregou corretamente:



```
#lsmod  
Module Size Used by  
Not tainted  
cmpci 26040 0 (unused)  
soundcore 3236 2 [cmpci]
```

Perceba que ele levantou o módulo cmpci e sua dependência soundcore que está sendo usado pelo cmpci soundcore 3236 2 [cmpci]



Lembre-se que nessa parte de módulos é mais ou menos como tentativa e erro.



No Red Hat, podemos usar um aplicativo chamado sndconfig.



```
#sndconfig
```



Que vem na instalação padrão e é pergunta da LPI!

Veja que a placa de som está funcionando (pelo menos foi carregada, veja com o lsmod) com o comando lsmod . Pelo menos seu módulo foi iniciado. Então, agora tem que testar. Existe muitas opções através de programas de áudio. Mas tem o jeito mais rápido que não precisa dos programas de áudio (xmms, mplayer).

Para você mexer com o volume em modo texto:



```
$ alsamixer
```

13.3. Exemplo de configuração da placa de rede.

Novamente executo o lspci para ver o modelo:



```
#lspci | grep -i eth  
00:0e.0 Ethernet controller: 3Com Corporation 3c905B 100BaseTX
```

Agora sei que a placa é uma 3com. Vá ao diretório:



```
# cd /lib/modules/2.6.18-4-686/kernel/drivers/net
```

Nesse diretório veja os módulos que a 3com tem e execute o modinfo para tentar detectar o modelo:



```
#modinfo 3c59x  
filename: 3c59x.ko  
author: Donald Becker <becker@scyld.com>  
description: 3Com 3c59x/3c9xx ethernet driver  
license: GPL
```

Este modelo é 3c905 e a resposta do modinfo diz que esse módulo é para todos os modelos 3c9XX .

Neste caso é possível saber que é o módulo: 3c59x , levante o módulo:



```
#modprobe 3c59x
```

Para finalizar, veja se o mesmo está no ar:



```
#lsmod  
  
Module Size Used by  
cmpci 26040 0 (unused)  
soundcore 3236 2  
[cmpci]  
3c59x 24648 1
```

Agora, basta testar usando o comando:



```
#ifconfig -a
```

Caso aparecer a eth0 está OK!



Para que esses módulos sejam carregados sempre na inicialização da máquina faça (Debian):

O de som:



```
# echo "cmpci" >> /etc/modules
```

O de rede:



```
# echo "3c59x" >> /etc/modules
```



Mais uma dica para a placa de rede: Se sua placa de rede suportar, defina que ela deve operar em full duplex.



```
# vi /etc/modules.conf  
alias eth0 8139too  
options 8139too full_duplex=1
```



Atenção: no arquivo `/etc/modules.conf` é possível relacionar que um módulo estará associado a uma determinada interface. Veja que `eth0` está associada ao módulo `8139too`. Isso é muito útil quando é preciso mais que uma placa de rede na máquina, assim o sistema sabe que tal módulo é para tal interface.

O modo full-duplex permite que a placa de rede envie e receba dados simultaneamente. Só vai existir vantagem a placa de rede operar nesse modo, se as máquinas usadas precisarem transmitir grande quantidade de dados ao mesmo tempo.

Para ver se sua placa de rede estará operando em modo full-duplex faça:



```
# mii-tool  
eth0: negotiated 100baseTx-FD flow-control, link ok
```

Se isso “100baseTx-FD” aparecer, sua placa de rede está operando no modo full-duplex.



*FD é uma abreviação para Full-Duplex.
HD é uma abreviação para Half-Duplex.*

Pode ser usado o mii-tool para fazer com que placa de rede opere no modo FD ou HD:

No modo FD:



```
# mii-tool -F 100baseTx-FD
```

No modo HD:



```
# mii-tool -F 100baseTx-HD
```

13.4. Configuração do teclado

O objetivo agora é configurar teclado, a linguagem do sistema e o mouse em modo texto que não precisam de módulos para funcionar, exceto ser for um dispositivo USB.

13.4.1. Teclado em modo texto.

Para configurar o teclado em modo texto, temos o comando: loadkeys

Na prática:



```
#loadkeys -d br-abnt2  
Loading /usr/share/keymaps/i386/qwerty/defkeymap.kmap.gz  
Loading /usr/share/keymaps/i386/qwerty/br-abnt2.kmap.gz
```

Onde a opção -d é para setar o que será nosso modelo por default.

Está sendo mostrado um exemplo para Teclado ABNT2. Para saber os nomes corretos entramos no diretório:



Debian



```
#cd /usr/share/keymaps/i386/qwerty
```



Red Hat



```
#cd /usr/lib/kbd/keymaps/i386/qwerty
```


Lá vai ser possível ver todos os modelos de teclado que o Linux suporta. No caso de teclados sem o " ç ":



```
# loadkeys -d us
```

Depois de ter nosso teclado corretamente instalado, passamos para a variável de ambiente. É muito importante ter a variável de ambiente corretamente setada no sistema, para que o teclado possa funcionar corretamente obedecendo a acentuação de sua localidade. Nossa localidade é o Brasil, então nossa variável é pt_BR

Para verificar a localidade que está no sistema, usa-se o comando locale. No Debian Sarge vinha como POSIX, agora no Etch já vem como pt_BR.



Na Prática (Debian Sarge):



```
#locale  
LANG=POSIX  
LC_CTYPE="POSIX"  
LC_NUMERIC="POSIX"  
LC_TIME="POSIX"  
LC_COLLATE="POSIX"  
LC_MONETARY="POSIX"  
LC_MESSAGES="POSIX"  
LC_PAPER="POSIX"  
LC_NAME="POSIX"  
LC_ADDRESS="POSIX"  
LC_TELEPHONE="POSIX"  
LC_MEASUREMENT="POSIX"  
LC_IDENTIFICATION="POSIX"  
LC_ALL=
```

Por padrão (Debian Sarge) essa variável vem definida com o POSIX em algumas distros (Debian é um exemplo). E isso faz com que o sistema fique todo em Inglês.

Exemplo:



```
#ifconfig
Link encap:Ethernet HWaddr 00:09:6B:DF:94:FF
inet addr:192.168.0.87 Bcast:192.168.0.255 Mask:255.255.255.0
inet6 addr: fe80::209:6bff:fedf:94ff/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:16007 errors:0 dropped:0 overruns:0 frame:0
TX packets:15433 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:12143652 (11.5 MiB) TX bytes:2511041 (2.3 MiB)
```

Como a saída desse comando está toda em Inglês, é preciso mudar para a localidade, que por padrão está como POSIX:

Para isso, primeiro é preciso que verifique se o pacote está instalado na máquina:



```
# dpkg -l | grep locales

ii locales 2.3.6.ds1-13 GNU C Library: National Language (locale)
```

Caso o mesmo não esteja instalado, é preciso instalar. Depois que ele foi instalado, use o dpkg reconfigure , no Debian, para definirmos para pt_BR:



```
#dpkg-reconfigure locales
```

Esse comando irá chamar um menu com diversas opções do locales. Escolha pt_BR, coloque o mesmo como default. Assim que acabar de configurar o locales, será criado um arquivo no /etc o qual está setado a configuração padrão do locales, no caso do Debian.

E esse é o arquivo importante:



```
#cat /etc/environment  
LANG=pt_BR
```



O Red Hat trata a localidade em variáveis a serem exportadas. Assim sendo, é preciso colocar essa variável no /etc/profile para a mesma ser exportada a cada boot do sistema:



```
#cat /etc/profile
```



```
export LANG=pt_BR  
export LC_ALL=pt_BR
```



No Debian, defini-se a variável naquele arquivo que foi mostrado. Depois basta deslogar e logar que as alterações já serão feitas. Exemplo:



```
#locale
LANG=pt_BR
LC_CTYPE="pt_BR"
LC_NUMERIC="pt_BR"
LC_TIME="pt_BR"
LC_COLLATE="pt_BR"
LC_MONETARY="pt_BR"
LC_MESSAGES="pt_BR"
LC_PAPER="pt_BR"
LC_NAME="pt_BR"
LC_ADDRESS="pt_BR"
LC_TELEPHONE="pt_BR"
LC_MEASUREMENT="pt_BR"
LC_IDENTIFICATION="pt_BR"
LC_ALL=pt_BR
```

Agora vejam a diferença no comando ifconfig:



```
# ifconfig

Encapsulamento do Link: Ethernet Endereço de HW 00:09:6B:DF:94:FF
inet end.: 192.168.0.87 Bcast:192.168.0.255 Masc:255.255.255.0
endereço inet6: fe80::209:6bff:fedf:94ff/64 Escopo:Link
UP BROADCASTRUNNING MULTICAST MTU:1500 Métrica:1
RX packets:16275 errors:0 dropped:0 overruns:0 frame:0
TX packets:15698 errors:0 dropped:0 overruns:0 carrier:0
colisões:0 txqueuelen:1000
RX bytes:12207543 (11.6 MiB) TX bytes:2534017 (2.4 MiB)
```

Comparando com o POSIX:



```
#ifconfig
```

```
Link encap:Ethernet HWaddr 00:09:6B:DF:94:FF  
inet addr:192.168.0.87 Bcast:192.168.0.255 Mask:255.255.255.0  
inet6 addr: fe80::209:6bff:fedf:94ff/64 Scope:Link  
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
RX packets:16007 errors:0 dropped:0 overruns:0 frame:0  
TX packets:15433 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:12143652 (11.5 MiB) TX bytes:2511041 (2.3 MiB)
```

Já foi configurado o teclado na sua localidade.

13.5. Configuração do mouse



Agora configure o mouse no modo texto. Para configurar o mouse no modo texto no Debian, precisamos ter instalado o gpm, pois a configuração será através dele.

Para instalar o mouse usamos:



```
#aptitude install gpm
```

Para configurar:



```
# dpkg-reconfigure gpm
```



É necessário ter o gpm, mas a configuração é feita pelo comando:



mouseconfig

Lá vai ter as opções de dispositivo e modelo para escolher. Lembre- se sempre:



Mouse Serial:

Device - /dev/ttyS0

Modelo - MouseMan ou Microsoft

Mouse PS/2:

Device - /dev/psaux

Modelo - PS/2

Mouse USB:

Device - /dev/input/mice

Modelo - PS/2