



# **Módulos do Kernel**

# Sumário

## Capítulo 1

Módulos do Kernel.....	3
1.1. Mãos a obra.....	4

## Capítulo 2

Gerenciando .....	7
2.1. Objetivos.....	7
2.1. Troubleshooting.....	7

## Índice de tabelas

## Índice de Figuras

# **Capítulo 1**

## **Módulos do Kernel**

- Gerenciar módulos do Kernel.

# 1.1. Mãos a obra

O suporte a recursos ao Kernel pode ser adicionado de algumas formas, aplicando patches para correções, compilando um recurso de forma interna no próprio Kernel ou através de módulos, onde o Administrador do sistema possa gerenciá-los quando o necessário.



*O que são módulos?*

Módulos são trechos de código encontrados em arquivos com a extensão .ko, que adicionam algum tipo de recurso ao Kernel, como por exemplo um suporte ao novo sistema de arquivos e/ou ao novo hardware da máquina.

O diretório dos módulos no sistema pode ser acessado em /lib/modules/versão\_do\_kernel. Exemplo:



```
# ls -l -color /lib/modules/$(uname -r)
```

```
drwxr-xr-x 9 root root 4096 Feb 24 2010 kernel
drwxr-xr-x 2 root root 4096 Feb 3 22:16 misc
-rw-r--r-- 1 root root 410560 Feb 3 22:16 modules.alias
-rw-r--r-- 1 root root 69 Feb 3 22:16 modules.ccwmap
-rw-r--r-- 1 root root 392448 Feb 3 22:16 modules.dep
-rw-r--r-- 1 root root 813 Feb 3 22:16 modules.ieee1394map
-rw-r--r-- 1 root root 451 Feb 3 22:16 modules.inputmap
-rw-r--r-- 1 root root 22763 Feb 3 22:16 modules.isapnpmap
-rw-r--r-- 1 root root 74 Feb 3 22:16 modules.ofmap
-rw-r--r-- 1 root root 297942 Feb 3 22:16 modules.pcimap
-rw-r--r-- 1 root root 1429 Feb 3 22:16 modules.serionmap
-rw-r--r-- 1 root root 190056 Feb 3 22:16 modules.symbols
-rw-r--r-- 1 root root 554230 Feb 3 22:16 modules.usbmap
```

## Comando uname

Para completar o comando ls, foi usado um sub-shell com o comando uname. Veja abaixo o uso do comando para a exibição das seguintes opções:

- i → Tipo de processador;
- m → Arquitetura da máquina;
- n → Nome da máquina na rede;
- p → Processador;
- o → Sistema operacional;
- r → Versão do código fonte do kernel;
- s → Nome do kernel;
- v → Versão de compilação do kernel.

## Gerenciando módulos

O pacote module-init-tools instalado no sistema traz uma série de comandos para gerenciar os módulos. Acompanhe a descrição e o exemplo de utilização.

**lsmod** → Lista módulos carregados. Exemplo de uso:



```
# lsmod
```

**modinfo** → Exibe informações do módulo, como descrição, nome do arquivo .ko, tipo de licença e dependências. Exemplo de uso:



```
# modinfo ext3
```

**modprobe** → Carrega o módulo e suas dependências. Exemplo de uso:



```
# modprobe cdrom
```

**insmod** → Insere o módulo usando o arquivo .ko, mas não carrega dependências. Exemplo de uso:



```
# insmod cdrom.ko
```

**modprobe -r** → Descarrega o módulo e suas dependências. Exemplo de uso:



```
# modprobe -r cdrom
```

**rmmod** → Descarrega o módulo sem controle de dependências. Exemplo de uso:



```
# rmmod cdrom
```

**depmod** → Recria o arquivo `modules.dep`, este usado para controle de dependências pelo comando `modprobe`. Exemplo de uso:



```
# depmod -a
```

### Carregar módulos na inicialização

Para carregar um módulo durante a inicialização, adicione o nome do módulo no final do arquivo `/etc/modules`. Exemplo:



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

No exemplo o módulo `loop` será carregado na inicialização. Para confirmar o carregamento do módulo, use o comando `lsmod`



```
# lsmod | grep loop
```

```
loop                12748  0
```

# Capítulo 2

## Gerenciando

### 2.1. Objetivos

- Troubleshooting: Identificar módulos do hardware.

### 2.1. Troubleshooting



*Como posso identificar os módulos de meu hardware?*

Isso é possível através do comando `lspci` com flag `-k`, que exibe os módulos de cada dispositivo presente em sua maquina. Vamos a prática:

Identificar o módulo da interface de rede



```
# lspci -k | grep -i eth -C 2 | grep "Kernel modules"
```

```
Kernel modules: piix  
Kernel modules: e1000
```



Exibir informações sobre o módulo



```
# modinfo e1000 | head
```

```
filename:      /lib/modules/2.6.26-2-686/kernel/drivers/net/e1000/e1000.ko
version:       7.3.20-k2-NAPI
license:       GPL
description:    Intel(R) PRO/1000 Network Driver
author:        Intel Corporation, <linux.nics@intel.com>
srcversion:    FEA549E5D933BA8322EF9A6
alias:         pci:v00008086d000010B5sv*sd*bc*sc*i*
alias:         pci:v00008086d00001099sv*sd*bc*sc*i*
alias:         pci:v00008086d0000108Asv*sd*bc*sc*i*
alias:         pci:v00008086d0000107Csv*sd*bc*sc*i*
```

Identificar o módulo do dispositivo de audio



```
# lspci -k | grep -i AUDIO -C 2 | grep "Kernel modules"
```

```
Kernel modules: vboxguest
Kernel modules: snd-intel8x0
```

Exibir informações sobre o módulo



```
# modinfo snd-intel8x0 | head
```

```
filename:      /lib/modules/2.6.26-2-686/kernel/sound/pci/snd-intel8x0.ko
license:       GPL
description:    Intel 82801AA,82901AB,i810,i820,i830,i840,i845,MX440; SiS 7012;
Ali 5455
author:        Jaroslav Kysela <perex@perex.cz>
alias:         pci:v000010B9d000005455sv*sd*bc*sc*i*
alias:         pci:v00001022d000007445sv*sd*bc*sc*i*
alias:         pci:v00001022d00000746Dsv*sd*bc*sc*i*
alias:         pci:v000010DEd00000026Bsv*sd*bc*sc*i*
alias:         pci:v000010DEd0000000EAsv*sd*bc*sc*i*
alias:         pci:v000010DEd0000000DAsv*sd*bc*sc*i*
```