Grupo de Resposta a Incidentes de Segurança – UFRJ

Última modificação: 18/06/05

Recompilando o kernel Linux

GRIS-2005-T-003

Universidade Federal do Rio de Janeiro
Instituto de Matemática
Departamento de Ciência da Computação
Grupo de Resposta a Incidentes de Segurança

Rio de Janeiro, RJ – Brasil

Recompilando o kernel Linux

GRIS-2005-T-003

Breno Guimarães de Oliveira
Frederico Henrique Bohm Argolo

A versão mais recente deste documento pode ser obtida na página oficial do GRIS

GRIS – Grupo de Resposta a Incidentes de Segurança
CCMN Bloco E 2° andar
Salas: E2000 e E2003
Av. Brigadeiro Trompowski, s/nº
Cidade Universitária - Rio de Janeiro/RJ
CEP: 21949-900
Telefone: +55 (21) 2598-3309

Sumário:

1 - Introdução

Sempre que uma distribuição Linux é instalada, um kernel – ou núcleo - do sistema é escolhido e instalado no mesmo. Acontece que o kernel Linux está sempre em desenvolvimento e novas versões vão surgindo muitas vezes sem aviso prévio. Essas versões implementam novos recursos e funcionalidades, deixam o sistema mais rápido e confiável, e corrigem falhas de segurança.

Infelizmente, hoje em dia vemos poucas pessoas preocupadas em manter seu kernel atualizado, e por isso muitos sistemas permanecem vulneráveis a ameaças remotas, ofuscadas pela utopia de que usando o Linux o usuário estará sempre seguro.

Assim, este tutorial procura detalhar ao máximo o processo de atualização do kernel Linux, para que todos possam usufruir sempre da última versão do mesmo sem receios, mantendo-se seguros.

Mãos à obra!

2 - Pré-requisitos

Antes de mais nada, você precisa obviamente ter um sistema Linux já instalado no computador. Além disso, outros programas são necessários para o procedimento. São eles:

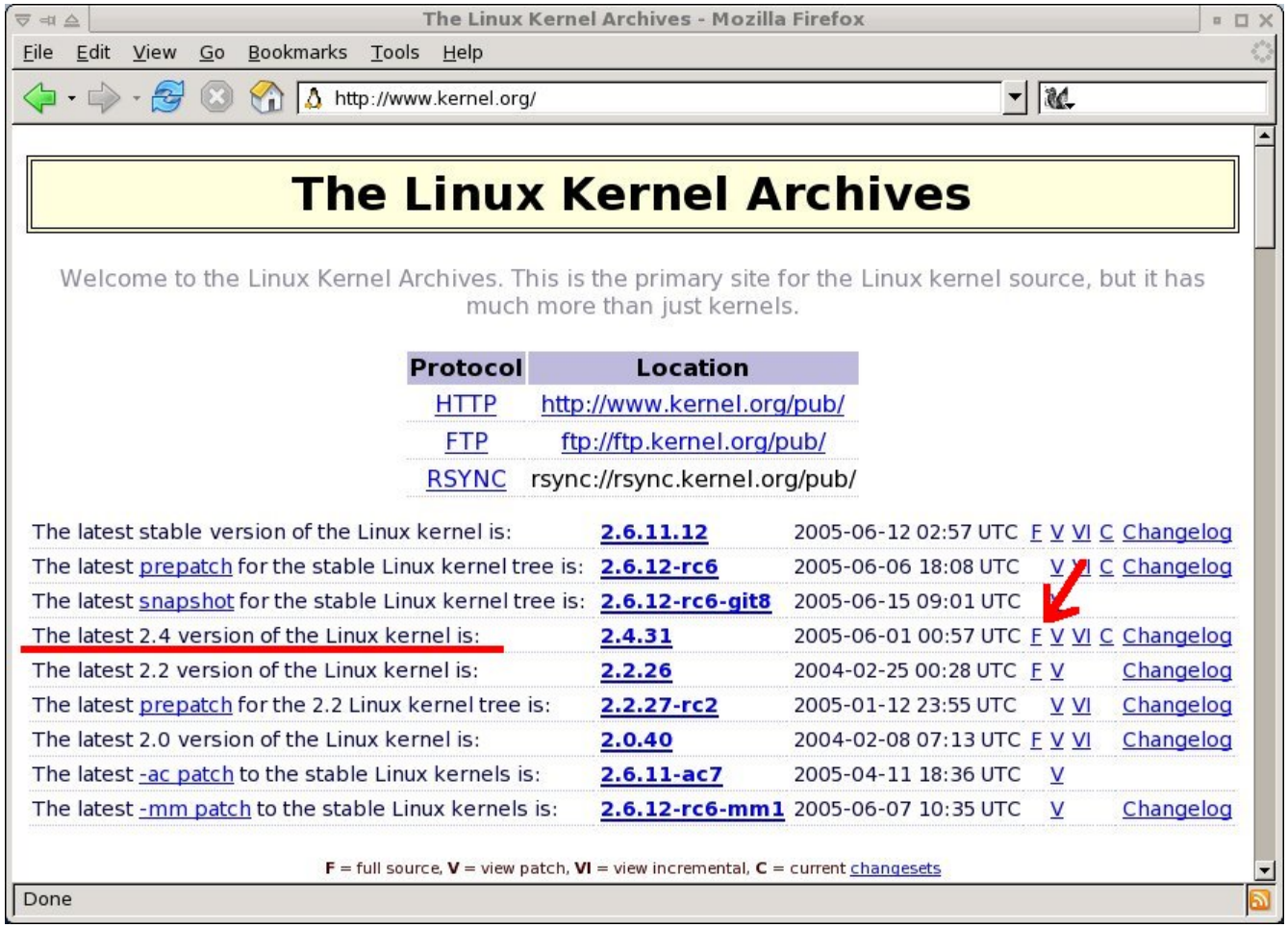
- `glibc`
- `gcc`
- `make`
- `binutils`
- `ncurses`

O mais provável é que o sistema já venha com todas as dependências acima, então preocupe-se com essa lista apenas se algo der errado.

3 - Obtendo e Descompactando

Entre em *http://www.kernel.org* para obter a última versão do kernel. É possível escolher entre diferentes versões, como a *2.4* e a *2.6*, que possuem diferentes características. O kernel utilizado para os exemplos é o *2.4.31*, mas o procedimento deve ser o mesmo para qualquer versão *2.4.X*. Para versões *2.6.X* do kernel, alguns dos passos não serão necessários (fique atento aos avisos ao longo do tutorial).

A página do pode parecer um pouco confusa à princípio, mas para baixar todo o código-fonte do kernel basta ir na linha contendo a versão escolhida e clicar no link "F" (de "Full source"). A figura abaixo mostra em vermelho a linha com a última versão do kernel *2.4*, com uma seta apontando para o link “F”.



Grave o arquivo no diretório `/usr/src/`. Caso não tenha permissão para escrever no diretório `/usr/src`, grave o arquivo em um diretório qualquer e depois copie para este diretório abrindo um terminal e digitando (com privilégios de "*root*"):

```
# mv linux-2.4.31.tar.bz2 /usr/src/
```

Lembre-se de substituir o número da versão na linha acima pelo número da versão que você escolheu.

Agora precisamos descompactar o arquivo obtido. Vá para o diretório `/usr/src/` ...

```
# cd /usr/src
```

... e digite o comando:

```
# tar jxvf linux-2.4.31.tar.bz2
```

Se tudo correu bem, um diretório chamado "`linux-2.4.31`" foi criado. Criamos então um link simbólico do mesmo chamado "`/usr/src/linux`":

```
# ln -sf /usr/src/linux-2.4.31 /usr/src/linux
```

Lembre-se de modificar a linha acima de acordo com o número da versão do kernel Linux obtida. Não se preocupe se o arquivo "/usr/src/linux" já existir. Ele é sempre um link simbólico apontando para a versão que se deseja compilar, e não é utilizado em mais nada pelo sistema.

Para confirmar se o link foi criado com sucesso, digite:

```
# ls -l
```

e procure pelo arquivo "`linux`". A última coluna da linha desse arquivo deve conter algo como

`linux -> /usr/src/linux-2.4.31/`

Verifique se o diretório apontado é o que você acabou de criar ao descompactar o kernel (em nosso caso, o link está correto e aponta para "`linux-2.4.31`"). Caso o link esteja apontando para outro local, remova-o com o comando "`rm /usr/src/linux`" e tente criá-lo novamente.

4 - Configurando

Chegou a hora de configurar o que desejamos incluir em nosso kernel. Por sua versatilidade e liberdade, o Linux permite a inclusão e exclusão de diversos elementos de seu núcleo. É importante identificar exatamente o que incluir e o que deixar de fora da sua compilação do kernel, e para isso vamos usar o menu de configurações do kernel.

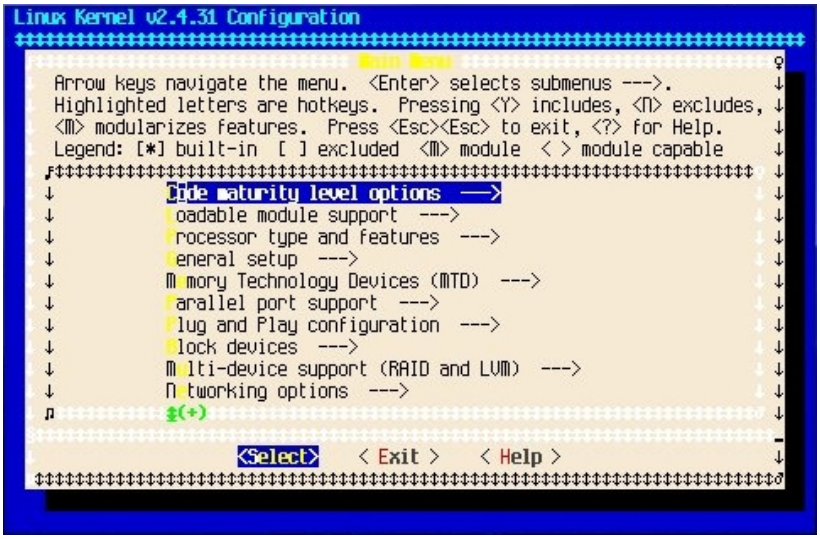
Primeiro, entre no diretorio com o código-fonte do kernel:

```
# cd /usr/src/linux
```

Depois digite o comando:

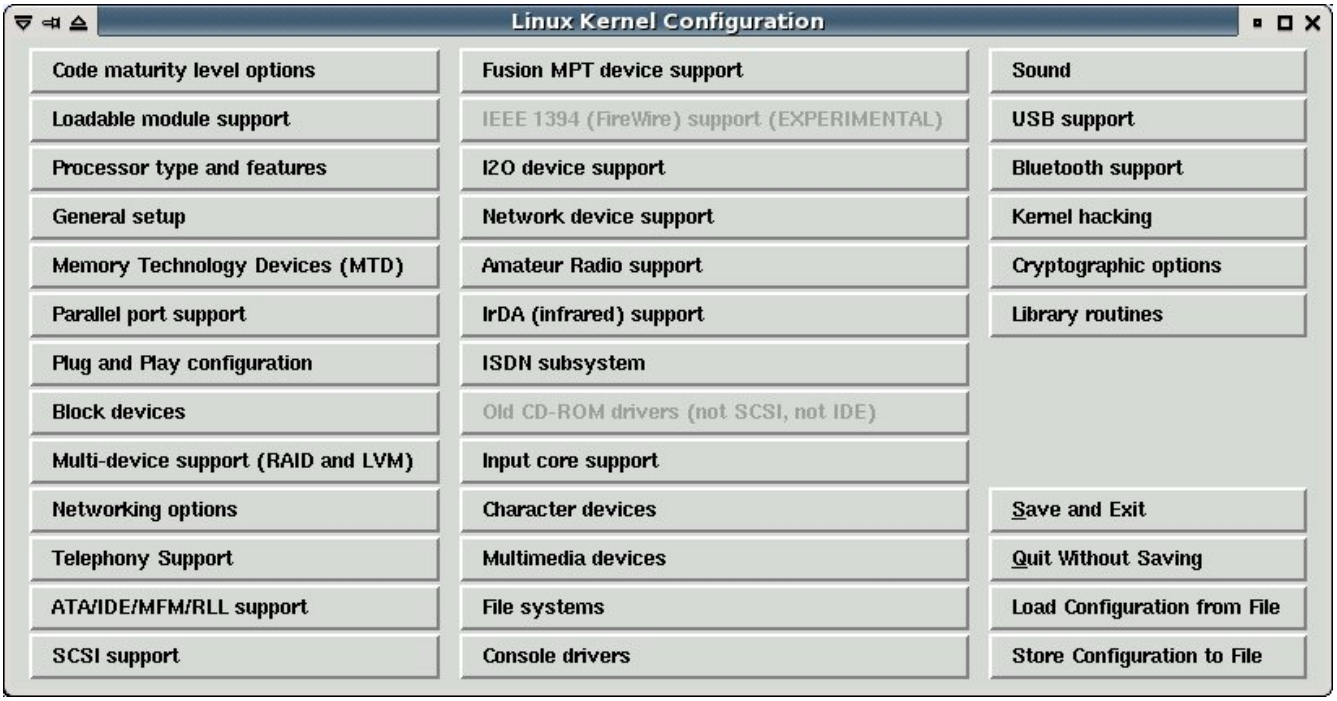
```
# make menuconfig
```

em seu terminal. Uma tela com um grande menu de configurações deve aparecer, como mostra a figura abaixo:



Utilize as setas do teclado para navegar pelo menu. Submenus acompanham uma seta "`--->`" e podem ser escolhidos apertando a tecla `[Enter]`. Para incluir uma opção, selecione-a com as setas e aperte "*y*". Para excluí-la, selecione e aperte "*n*". Algumas opções podem ser compiladas como módulos separados do kernel, que podem ser ativados/desativados sob demanda. Para compilar a opção como módulo, selecione-a e aperte "*m*".

Caso seu sistema possua ambiente gráfico, é possível digitar "`make xconfig`" dentro de um terminal em seu ambiente gráfico para iniciar a configuração via janelas. O resultado será um menu de configurações parecido com o da imagem abaixo:



A quantidade de opções é enorme e por isso não vamos percorrer aqui cada uma delas. Recomendamos que você faça isso com calma, para não deixar nada importante de fora. O menu oferece uma ajuda explicativa para quase todas as opções, que você pode ver usando as setas direita e esquerda do teclado e escolhendo a opção "`help`" com a opção que deseja devidamente selecionada. Lembre-se, no entanto, que o tamanho do kernel é limitado e muitas coisas podem (e devem) ser incluídas como módulos (através da opção "*m*"), e não diretamente (através da opção "*y*").

Alguns submenus que valem a pena serem explorados:

> **"*Processor type and features*"** - Escolha aqui o modelo que mais se adequa ao seu processador. Se estiver na duvida, escolha "386".
>
> ***"General setup"*** - Certifique-se de que a opção "*Networking support*" esteja marcada. Marque também "*PCI support*" e "*support for hot-pluggable devices*", caso possua placas PCI (você provavelmente têm) e pretenda utilizar dispositivos USB (como uma câmera digital), respectivamente.
>
> ***"Plug and Play configuration"*** - Marque tudo, caso não estejam.
>
> ***"Networking options"*** - Se pretende ligar seu sistema à Internet ou em outra rede, marque a opção "*Network packet filtering*" deste submenu para que a firewall seja compilada, e dentro de "*IP: Netfilter Configuration*", marque pelo menos a opção "*IP tables support*".

Se você tem ou pretende ter gravadores de CD/DVD, convém adicionar suporte a emulação SCSI. Para isso, certifique-se de que a opção "*ATA/IDE/MFM/RLL support*"`--->`"*IDE, ATA and ATAPI Block devices*"`--->`"*SCSI emulation support*" esteja marcada.

Caso utilize placas de vídeo ou outros dispositivos AGP, marque a opção "*Character devices*"`--->`"*/dev/agpgart (AGP Support)*" e procure ativar também quaisquer opções relativas a sua placa de vídeo.

No submenu "*File systems*", marque as opções "*Reiserfs support*", "*Ext3 journalling file system support*", "*DOS FAT fs support*", "*Compressed ROM file system support*", "*ISO 9660 CDROM file system support*", "*NTFS file system support*", "*/proc file system support*" e "*Second extended fs support*".

Se pretende acessar os diretórios compartilhados de um sistema Windows, marque a opção "*File systems*"`--->`"*Network File Systems"*`--->`"*SMB file system support*".

Quando terminar de configurar o novo kernel, volte para o menu principal e aperte a tecla `[ESC]` duas vezes. Uma caixa de confirmação aparecerá perguntando se deseja salvar suas configurações. Escolha a opção "*Yes*" e aperte `[Enter]`.

Se tudo correu bem, seu kernel já está devidamente configurado. Caso tenha cometido algum erro ou não se lembre se ativou ou não determinada opção, digite novamente "`make menuconfig`": ele estará exatamente como você o deixou.

5 - Compilando

Agora é a hora da verdade. Antes precisamos verificar e compilar todas as dependências das opções escolhidas. Para isso, se estiver compilando uma versão *2.4.X* do kernel (como o nosso exemplo), digite:

```
# make dep
```

e vá tomar uma água, pois demora um pouco. Caso algum aviso de erro tenha surgido logo antes da execução terminar, verifique se os pré-requisitos estão lá (seção 2 deste documento) ou reveja suas opções no "`make menuconfig`".

Se estiver compilando uma versão *2.6.X* do kernel (como a 2.6.10, por exemplo), não é necessário digitar "`make dep`".

Agora vamos limpar o resultado das compilações passadas, arquivos temporários e tudo mais que foi criado nos passos anteriores mas que não é mais necessário. Para isso, digite:

```
# make clean
```

De fato, é importante digitar esse comando sempre que modificar qualquer elemento de configuração do seu kernel, uma vez que o sistema precisa renovar todos os arquivos modificados.

Agora vamos compilar o kernel Linux efetivamente. Para tal, digite:

```
# make bzImage
```

e vá comer alguma coisa, ler um livro, porque esse demora bastante (em torno de 10 minutos em um Pentium IV, dependendo das opções).

Caso algum aviso de erro tenha surgido logo antes da execução terminar, verifique se os pré-requisitos estão lá ou reveja suas opções no "`make menuconfig`", já que muitas opções não convivem bem junto de outras, ou não funcionam para determinados sistemas. Observe o conteúdo da mensagem de erro para obter informações sobre o que causou o mesmo. Durante um dos testes para este tutorial, tentamos incluir o módulo de uma placa de vídeo *SiS* juntamente com outras placas, e obtivemos erro de compilação referente ao mesmo. Foi só remover a opção no "`make menuconfig`" que a compilação correu sem problemas. Caso sua compilação também apresente problemas, observe a mensagem de erro, digite "`make clean`" e volte ao passo do "`make menuconfig`", procurando identificar qual módulo cuja presença (ou ausência) causou o problema.

Passada essa etapa, chegou a hora dos módulos que devem ser compilados separadamente. Digite:

```
# make modules
```

O procedimento não deve demorar muito, mas depende da quantidade de módulos definidos durante a etapa do "*menuconfig*". O procedimento caso essa etapa tenha apresentado erros é o mesmo da etapa anterior: anote o erro, digite "`make clean`" e volte ao passo do "`make menuconfig`". Uma dica: Na etapa do "`make modules`" só entram os módulos que foram marcados para compilação separada, que podem ser identificados no menu de opções através da marca "`M`", ao invés do "*".

Finalmente, concluindo a etapa de compilação, vamos instalar os módulos recém compilados. Para tal, digite:

```
# make modules_install
```

6 - Usando o novo kernel

Agora que o novo kernel está devidamente compilado no diretório */usr/src/linux* (que é um link simbólico para o diretório que você criou), precisamos copiá-lo para o diretório */boot* para que possamos utilizá-lo. O diretório */boot* contém o seu kernel atual, mas não se preocupe, nada será substituído. Para efetuar a cópia, digite:

```
# cp /usr/src/linux/arch/i386/boot/bzImage  /boot/vmlinuz-2.4.31
```

Lembre-se de substituir o número da versão do comando acima pela versão que você obteve (em nosso caso, *2.4.31*). Note também que, dependendo da arquitetura de seu sistema e de suas escolhas no "`make menuconfig`", o binário *bzImage* pode ter aparecido em outro diretório. O nome vmlinuz é uma convenção, e a versão é justamente para distingüir essa compilação de outras versões (inclusive da que atualmente funciona em seu sistema).

É preciso também copiar o arquivo "*System.map*" localizado no diretório */usr/src/linux* para o seu diretório */boot* (mudando novamente o nome para ficar claro a qual versão do kernel esse arquivo pertence).

```
# cp /usr/src/linux/System.map  /boot/System.map-2.4.31
```

Mudamos agora o link do arquivo /boot/System.map para apontar para o nosso novo arquivo:

```
# ln -sf  /boot/System.map-2.4.31  /boot/System.map
```

Para confirmar se o link foi criado com sucesso, entre no diretório */boot* e digite:

```
# ls -l
```

Procure pelo arquivo "*System.map*". A última coluna da linha desse arquivo deve conter algo como:

```
System.map -> System.map-2.4.31
```

Verifique se o arquivo apontado é o que você acabou de copiar (em nosso caso, o link está correto e aponta para "*System.map-2.4.31*"). Caso o link esteja apontando para outro local, remova-o com o comando "`rm /boot/System.map`" e tente criá-lo novamente.

Nota: o arquivo *System.map* funciona como uma tabela de símbolos para o kernel. Substituir essa referência por uma nova é apenas para facilitar, já que o kernel em atividade deve procurar no diretório */boot* pela tabela correspondente à sua versão. No pior caso, mesmo que não encontre, o máximo que vai acontecer é a exibição periódica de uma mensagem como "*system.map does not match actual kernel*" e a execução incorreta de alguns programas não críticos. Caso perceba essa mensagem, apenas refaça o link do arquivo "*/boot/System.map*" para o arquivo original.

Finalmente, precisamos editar o LILO para que possamos selecionar o novo kernel sempre que o sistema for ativado. Se o seu sistema não utiliza o LILO, procure documentação da ferramenta alternativa.

Para editar o arquivo de configurações do LILO, use seu editor de textos favorito e abra o arquivo */etc/lilo.conf*. Para fazer isso utilizando o editor "*vim*", por exemplo, digite:

```
# vim /etc/lilo.conf
```

Dentro do arquivo "*lilo.conf*", tudo que você deve fazer é acrescentar as seguintes linhas ao final do arquivo:

```
image = /boot/vmlinuz-2.4.31
  root = /dev/hdXn
  label = Kernel-2.4.31
  read-only
```

ATENÇÃO: O seu arquivo "lilo.conf" já deve conter linhas similares para a configuração de seu kernel atual. Não modifique essas linhas! Apenas adicione as linhas acima como uma nova entrada, ao final do arquivo. Isso permitirá que você escolha ao ativar o sistema qual *kernel* deseja utilizar: o seu antigo ou o recém compilado. Dessa forma, caso haja algum problema com seu kernel novo, é sempre possível iniciar o sistema pela versão anterior.

Note que as linhas acima **não** devem ser copiadas literalmente. Veja os detalhes de cada uma:

"`image = /boot/vmlinuz-2.4.31`"

Modifique a linha acima de acordo com o nome do arquivo de kernel que você copiou para o diretório */boot* (início do passo 6).

"`root = /dev/hdXn`"

Substitua "`hdXn`" pelo nome da unidade de disco que contém a partição "`/`" (*root*) do sistema, como por exemplo *hda1*, *hdc2*, etc. Se estiver na dúvida, olhe um pouco mais acima no arquivo

"*lilo.conf*" e você deve encontrar a entrada antiga do seu sistema Linux atual. Ela deve ter um campo "*root* =" indicando o nome da unidade correta.

"`label = Kernel-2.4.31`"

Aqui você deve dar um nome único para o kernel. Esse é o nome que vai aparecer no menu de opções do LILO durante o início do sistema. **O NOME DEVE SER APENAS UMA PALAVRA** (ou seja, nada de espaços) mas nao precisa ser "*Kernel-2.4.31*". Escolhemos esse nome apenas porque ele é bastante sugestivo e não trará dúvidas na hora da escolha.

"`read-only`"

Esta linha deve conter exatamente isso mesmo. ;-)

Finalmente, grave as modificações e volte ao terminal. Digitamos agora o comando "lilo" para que o arquivo de configurações que acabamos de modificar seja carregado:

```
# lilo
```

Caso tenha recebido uma mensagem dizendo

"`Fatal: Kernel /boot/vmlinuz-2.4.31 is too big`"

volte ao "*menuconfig*" (seção 6) e repita os passos de configuração, transformando alguns dos itens incluídos diretamente no kernel (marcados com "*") em módulos (marcados com "M") e repita o procedimento.

Se tudo correu bem, você deve ver uma mensagem dizendo algo como:

"`Added kernel-2.4.31`"

Isso significa que o novo kernel já pode ser acessado no menu do LILO da próxima vez que o seu sistema for ligado. Parabéns!

Caso algum problema ocorra ao tentar iniciar seu sistema com o novo kernel, basta reiniciar o sistema e ativar o kernel antigo no menu do LILO para que o seu sistema retome o estado original.

Lembre-se de verificar periodicamente a página oficial do kernel linux para obter novas versões e manter seu sistema sempre atualizado, estável e seguro.