Compilação de Kernel

Introdução

O Kernel é o coração do sistema, o Linux em sí. Todos os demais programas, incluindo o shell, são softwares que rodam sobre o Kernel. A função principal dele é ser o ponto de ligação entre os programas e o hardware.

Para manter a compatibilidade com o maior número possível de dispositivos de hardware, as distribuições devem incluir também todos ou quase todos os drivers de dispositivos disponíveis para o Linux. Para evitar que isto torne o Kernel muito grande, é criado um kernel básico, com os drivers mais importantes, incluindo assim os demais drivers como módulos. Durante a instalação, os módulos necessários são carregados no kernel. E como vimos anteriormente, podemos carregar ou descarregar os módulos da memória, de acordo com o seu uso ou não.

Compilando o Kernel

Mas, voltando ao tema principal, recompilar o Kernel do Linux lhe dá a chance de criar um kernel adaptado às suas necessidades, ao contrário do tamanho único incluído nas distribuições. Além disso, você vai precisar recompilar o Kernel, caso precise adicionar o suporte a algum dispositivo, cujo driver só está disponível na versão mais recente. Por exemplo, se o kernel que você instalou **não suportar** o seu **disco SATA**, ou sua **placa de rede**, ou seu **dispositivo bluetooth**, então você precisará compilar um novo kernel para adicionar esse novo suporte.

Adquirindo um novo Kernel

O primeiro passo é naturalmente obter o código-fonte do Kernel, que iremos compilar. Se você quer apenas criar um kernel personalizado, pode usar como base o próprio kernel incluído na sua distribuição. Ou então, você pode baixar a versão mais recente no site **http://www.kernel.org**, onde poderá encontrar tanto a versão de **desenvolvimento** (sendo **ímpar o 2º número da versão**, como em **2.5.x**), quanto a versão **estável** (sendo **par o 2º número da versão**, como em **2.6.x**).

Preparando a Compilação

Para realizar a compilação, você precisará instalar no seu sistema os seguintes pacotes:

 # apt-get install gcc make ncurses-bin ncurses-base libncurses5 libncurses5-dev

OBS: Para verificar se os pacotes estão instalados, podemos usar o seguinte comando:

 # dpkg -l | grep <nome do pacote>

Descompactando o Kernel

Feito o download do código-fonte do Kernel, salve o arquivo no diretório **/usr/src**, onde por padrão ficam armazenados os fontes do Kernel (o arquivo com o fonte do Kernel é muito grande, com mais de **30 MB** nas versões recentes! Porém, depois de compilado, ele ficará bem menor).

Depois de baixar o pacote, você deverá descompactá-lo no diretório onde você salvou o código-fonte (o /usr/src):

```
# cd /usr/src
# tar -xvjf linux-2.x.x.tar.bz2
```

OBS: será gerado um sub-diretório com o mesmo nome da versão, no padrão **/usr/src/linux-2.x.x**

Iniciando a Compilação

Acesse agora o diretório **/usr/src/**, onde os fontes da versão do Kernel que será recompilada está armazenada:

 # cd /usr/src/linux-2.x.x

Com o código em mãos, o próximo passo é definir os componentes que serão incluídos no novo Kernel. Para isto, temos três opções:

 # make config – configuração via texto
 # make menuconfig – configuração via interface ncurses
 # make xconfig – configuração via interface gráfica

O comando make menuconfig

Usaremos no curso o estilo do **menuconfig,** pois o mesmo é em **modo console** (texto), porém possui uma janela interativa em uma interface da biblioteca **ncurses**, para que possamos recompilar o Kernel.

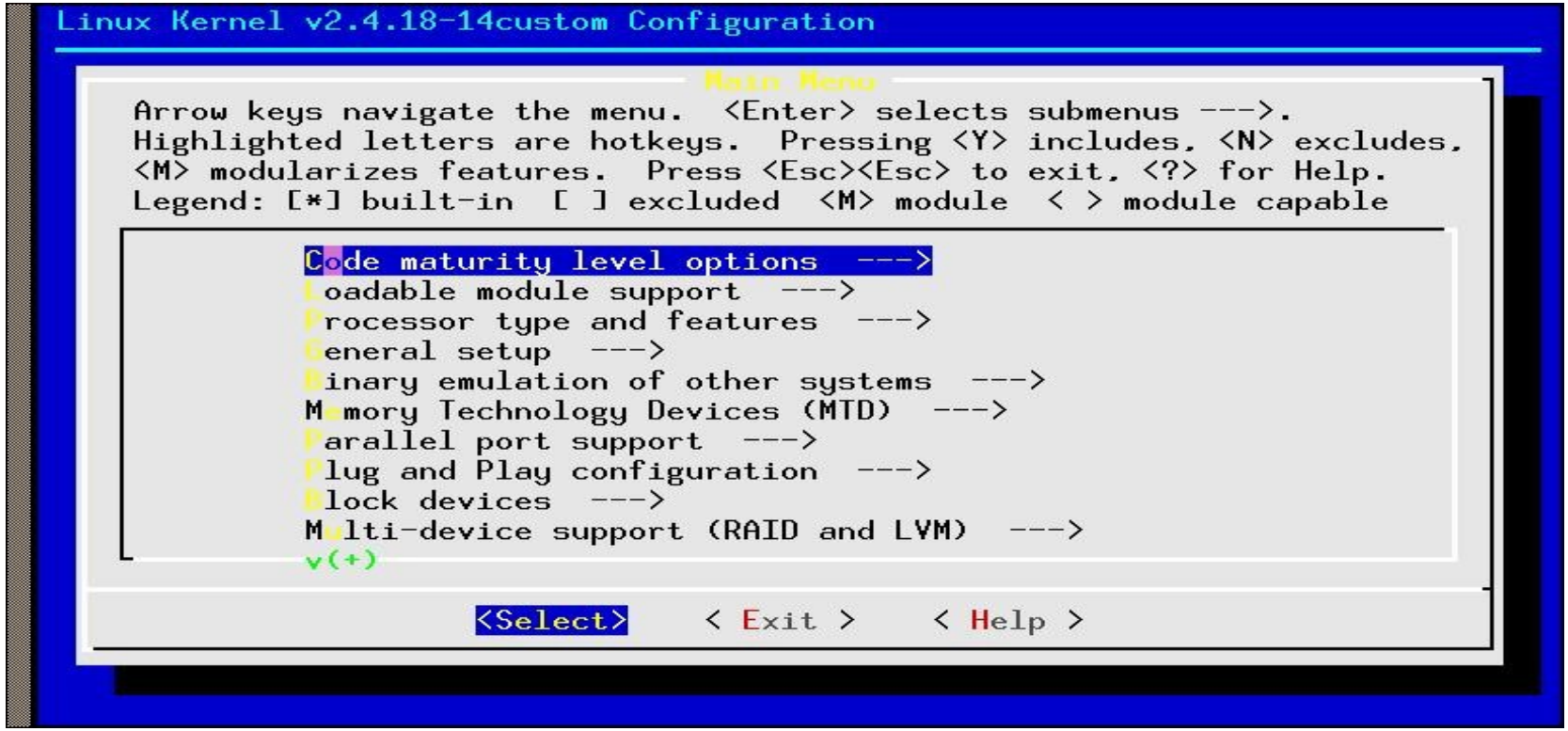
OBS: Essa opção necessita obrigatoriamente que o ncurses esteja instalado na máquina, como verificamos anteriormente.

Para executar o menuconfig, basta digitá-lo no shell:

 # make menuconfig

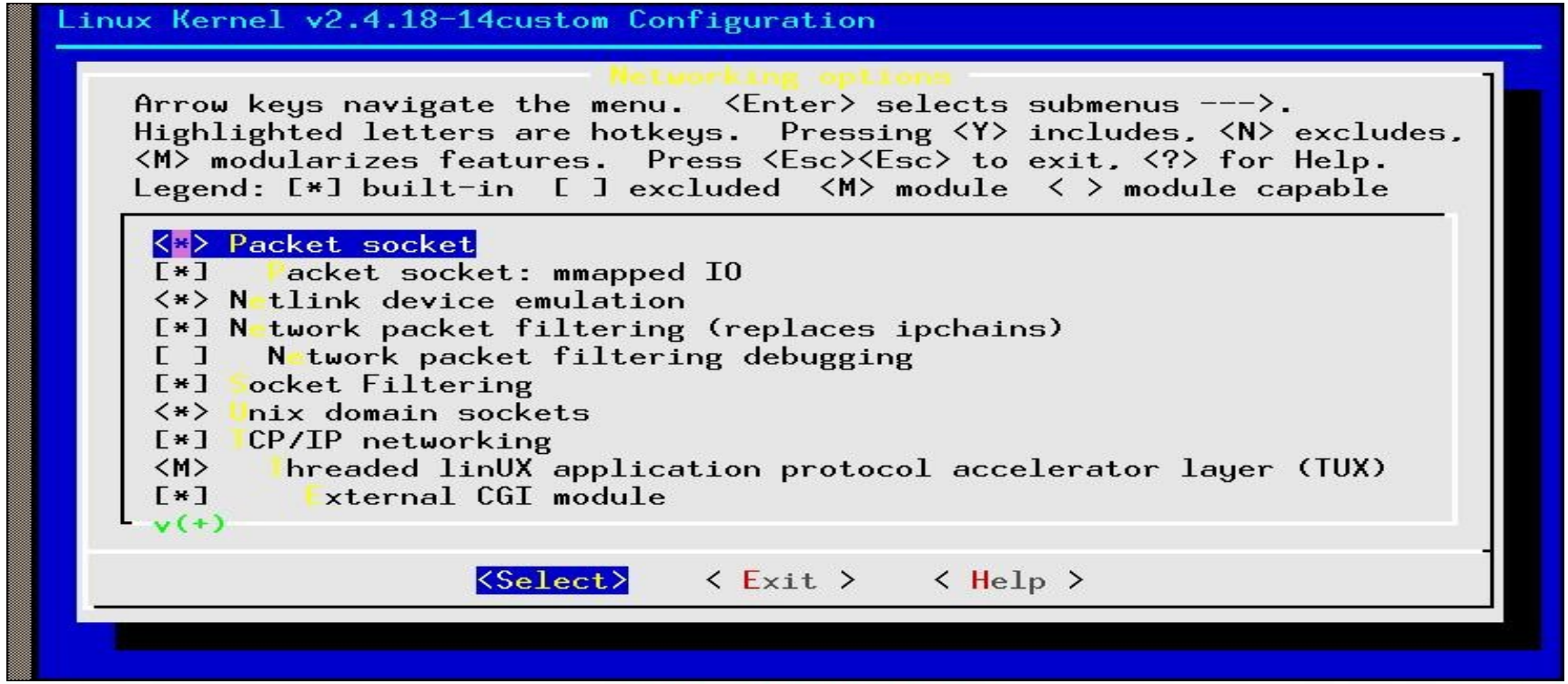
O comando make menuconfig (cont.)

Os componentes disponíveis estão organizados em categorias. A maior parte se relaciona justamente ao suporte a dispositivos:



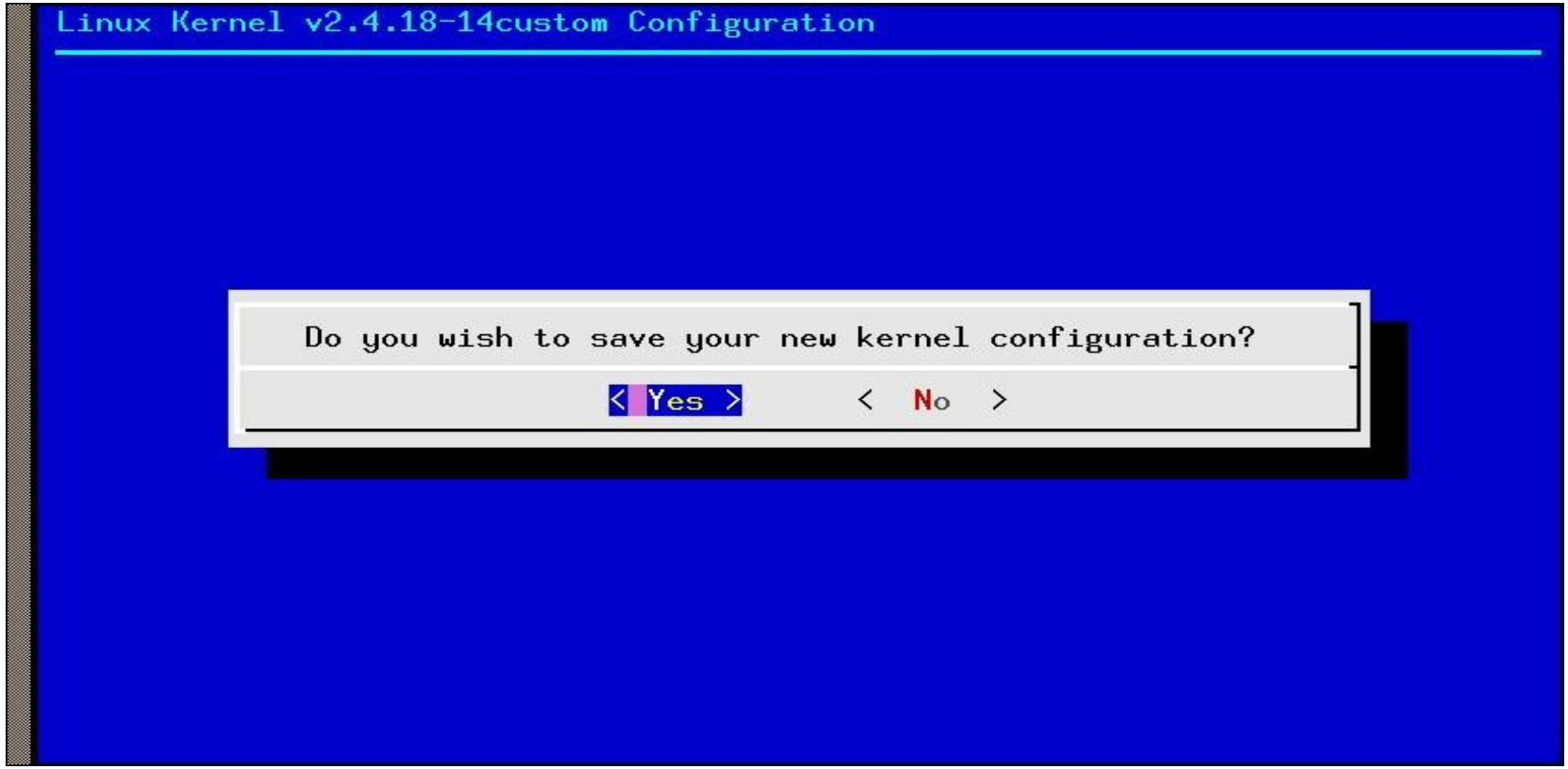
O comando make menuconfig (cont.)

Para cada módulo, existem três opções padrões: **Built-in (*)**, **No ( )** ou **Module (M)**, que permite carregar o componente na forma de um módulo, sem inchar o Kernel. Esta é a opção ideal para todos aqueles componentes que quiser manter, porém não tem certeza se serão usados freqüentemente:



O comando make menuconfig (cont.)

Depois de terminar, clique na opção **“Exit”** no **Menu Principal** e escolha a opção para salvar todas as alterações.



Compilando o Kernel

Depois de configurar o novo kernel, basta compilá-lo com os comando abaixo:

 # make

Ele irá verificar a cadeia de interdependências do kernel, assegurando que todos os componentes necessários farão parte da compilação. Este é o comando que realmente compila o Kernel, ou seja, o mais importante de todos, que irá gerar o novo kernel.

OBS: o procedimento anterior é utilizado para **kernels 2.6.x**, em **kernels 2.4.x** seriam dois comandos: **make dep** e **make bzImage.**

Compilando o Kernel (cont.)

O próximo comando, após termos o kernel compilado, seria:

 # make modules ; make modules_install

Este último comando conclui o trabalho, gerando os componentes que serão adicionados como módulos e criando a estrutura dos módulos do novo kernel.

Além disso, precisamos gerar o arquivo **initrd** do novo kernel, que será carregado na inicialização:

 # mkinitrd -o /boot/initrd-2.x.x.img /lib/modules/<versão do novo kernel

Instalando o Kernel

Após a compilação, o novo kernel será gravado no arquivo **/usr/src/linux-2.x.x/arch/i386/boot/bzImage.**

Então deveremos copiá-lo para o diretório **/boot** e em seguida configurar o **gerenciador de boot** (LILO ou GRUB) para inicializar o novo kernel, a fim de termos a possibilidade de trabalhar com o novo kernel e o antigo, caso ocorra algum erro com o novo kernel:

 # cp /urs/src/linux-2.x.x/arch/i386/boot/bzImage /boot/novo_kernel

Onde em “novo_kernel” pode ser o nome que você queira colocar.

Configurando o boot-loader (LILO)

Para carregar o novo kernel, edite o arquivo /etc/lilo.conf:

 # vi /etc/lilo.conf

Nesse arquivo ficam as opções de inicialização que são dadas durante o boot. Precisamos apenas adicionar nossa nova imagem do kernel, acrescentando essas linhas no final do arquivo:

image=/boot/novo_kernel
initrd=/boot/initrd-2.x.x.img
label=Novo Kernel
read-only

Para validar as novas configurações do arquivo, basta digitar:

 # lilo

Configurando o boot-loader (GRUB)

Para carregar o novo kernel, edite o arquivo **/boot/grub/menu.lst**, com as seguintes linhas:

```
title          Novo Kernel
root           (hd0,1)
kernel         /novo_kernel root=/dev/hda3 ro
initrd         /initrd.img-2.x.x
savedefault
boot
```

Para validar as novas configurações do arquivo, basta digitar:

```
# grub-install /dev/hda
```

Referências Bibliográficas

Linux - Guia do Administrador do Sistema
Autor: Rubem E. Pereira
Editora: Novatec

Manual Completo do Linux (Guia do Administrador)
Autor: Evi Nemeth, Garth Snyder, Trent R. Hein
Editora: Pearson Books

Guia Foca GNU/Linux
http://focalinux.cipsga.org.br/