Kernel/Configuration

This article describes the manual configuration and setup of the Linux kernel. For an automatic method see the genkernel article.

Contents

Set symlink

The symlink `/usr/src/linux` should always point to the kernel sources that is currently being used. This can be done in one of three ways:

1. Installing the kernel sources with active *symlink* USE flag
    This will make the `/usr/src/linux` point to your newly installed kernel sources. If necessary, it can still be modified later with the following two methods:

2. Setting the symlink with **eselect**

```
root #  eselect kernel list

Available kernel symlink targets:
  [1] linux-3.3.8-gentoo *
  [2] linux-3.4.9-gentoo
```

This outputs the available kernel sources. The asterisk indicates the chosen sources. To change the kernel sources, e.g. to the second entry, do:

```
root # eselect kernel set 2
```

3. Setting the symlink manually

```
root # ln -sf /usr/src/linux-3.4.9-gentoo /usr/src/linux
root # ls -l /usr/src/linux

lrwxrwxrwx 1 root root 11 Aug 29 22:10 /usr/src/linux -> /usr/src/linux-
3.4.9-gentoo
```

Configuration tools

The kernel offers several tools to configure itself:

make config
> Text based configuration. The options are prompted one after another. All options need to be answered, and out-of-order access to former options is not possible.

make menuconfig
> Graphical menu (only text input). You can navigate through the menu and change the options you want.

make nconfig
> Graphical menu based on ncurses. Requires sys-libs/ncurses (http://packages.gentoo.org/package/sys-libs/ncurses) to be installed.

make xconfig
> Graphical menu using Qt4. Requires dev-qt/qtgui (http://packages.gentoo.org/package/dev-qt/qtgui) to be installed.

make gconfig
> Graphical menu using GTK+. Requires x11-libs/gtk+ (http://packages.gentoo.org/package/x11-libs/gtk+), dev-libs/glib (http://packages.gentoo.org/package/dev-libs/glib) and gnome-base/libglade (http://packages.gentoo.org/package/gnome-base/libglade) to be installed.

make oldconfig
> Review changes between kernel versions and update to create new `.config` for kernel.

There are also several scripts to create miscellaneous default configurations. See:

```
root # make help
```

Configuration

This article describes the configuration using **make menuconfig**, but the procedure is similar for the other menus.

```
root # cd /usr/src/linux
root # make menuconfig
```

Usage

In the shown menu the blue bar indicates your position. With the arrow keys `up` and `down` you can change the position. The arrow keys `left` and `right` traverse the menu bar in the bottom and define what happens when you hit the `Enter` key. **Select** switches to a sub menu - the menu entries ending with ---> - , while **Exit** exits a sub menu. As an alternative you can also hit the `Esc` key twice.

Menu entries, which start with brackets, are drivers or features which can be activated:

[ ], [*]
> Options in square brackets can be activated or deactivated. The asterisk marks the menu entry as activated. You change the value with the `space` key or press `Y` (**Y**es) to activate or `N` (**N**o) to deactivate the entry.
> If the option is activated, the chosen driver will be built into the kernel and will always be available at boot time.

< >, <M>, <*>
> Options in angle brackets can be activated or deactivated, but also activated as module (indicated by a *M*). You change the value as before and press key `M` for activate as module.
> See the Kernel Modules article for differentiation.

{M}, {*}
> Options in curly brackets can be activated or activated as module but not be deactivated, because another driver needs this activated in either way.

-M-, -*-
> Options between hyphens are activated in the shown way by another driver. There is no choice.

Furthermore some menu entries have a tag at the end:

(NEW)
> This driver is new in the kernel and is maybe not stable enough.

(EXPERIMENTAL)
> This driver is experimental and most likely not stable enough.

(DEPRECATED)
> This driver is deprecated and not needed for most systems.

(OBSOLETE)
> This driver is obsolete and should not be activated.

Most options have a description, which get by pressing the `H` key or choose in the menu bar **Help.**

Driver selection

See the hardware detection article and the articles in the Hardware category.

Search modules

Within *menuconfig*, you can use `/` to search modules by name.

Enabling Gentoo Linux common settings

There is a kernel configuration option called `CONFIG_GENTOO_LINUX` only present in sys-kernel/gentoo-sources (http://packages.gentoo.org/package/sys-kernel/gentoo-sources) and other Kernel Project maintained kernels. It does nothing on its own, but sets various required configuration options for typical installations.

This setting automatically selects tmpfs and devtmpfs support, which are needed for handling `/dev` on Gentoo Linux, but might be expanded in the future to enable other mandatory settings for a Gentoo Linux system. For more information, read the help information available through the kernel configuration system as described earlier in this guide.

Build

After configuring the kernel you have to compile it:

```
root # make
```

If you have a processor with multiple cores, you can let all the cores do the work. For this add the parameter **-j(NUMBER_CORES +1)**. For a dual core processor:

```
root # make -j3
```

Setup

If activated drivers as modules, you have to install them:

```
root # make modules_install
```

The modules will be copied to a sub directory of `/lib/modules`.

To install the actual kernel:

```
root # make install
```

This command executes **/sbin/installkernel**, which is part of the sys-apps/debianutils (http://packages.gentoo.org/package/sys-apps/debianutils) package. *The new kernel is installed into* `/boot/vmlinuz-{version}`. *If a symbolic link* `/boot/vmlinuz` *already exists, it is refreshed by making a link from* `/boot/vmlinuz` *to the new kernel, and the previously installed kernel is available as* `/boot/vmlinuz.old`. (*installkernel* man page). The same for `config` and `System.map` files. These symlinks are handy, because they point always to the newest kernel without changing the file path (e.g. you can use them in the bootloader configuration).

Bootloader

Change your bootloader configuration to pick up at boot the new kernel.

Finally restart your system with the new kernel.