

How to Compile the Linux Kernel

James Andrews

Saturday, May 29, 1999 01:36:47 PM

Introduction

RedHat is not Linux, it is a distribution based on the Linux kernel. The Linux kernel is a complex program which provides the underlying services to the rest of a Linux distribution. But it is easy to add new features or improvements to it as, unlike commercial operating systems like Windows 95 or MacOS, the source code is freely available. It is common practice with a Linux based operating system to recompile the kernel from source and much effort has been put in to make this a relatively user-friendly experience.

Why Compile a New Kernel?

You may be thinking "But why recompile it? It works fine as it is." There are three reasons for a recompile. Firstly, you may have some hardware that is so new that there's no kernel module for it in on your distribution CD. Secondly, you may have come across some kind of bug which is fixed in a revision of the operating system. Lastly, you may have some new software which requires a newer version of the operating system.

Getting the Kernel Sources

The most recent releases of the kernel sources are available on <ftp.kernel.org>. This site is mirrored worldwide so there is probably a fast copy of it somewhere near you. Locate the file for the latest version of the operating system and download it to `/usr/src`. Then issue the `tar xzf` command to unpack it.

If you do not need the very latest version, then installing a kernel source package from your distribution CD may be an easier way to proceed.

Configuring For a Build

Change directory to `/usr/src/Linux` and issue the command:

```
make menuconfig
```

This will build a few programs and then quickly pop up a window. The window menu lets you alter many aspects of kernel configuration.

After you have made any necessary changes, save the configuration and follow these instructions--do a

```
make dep; make clean
```

The first of these commands builds the tree of interdependencies in the kernel sources. These dependencies may have been affected by the options you have chosen in the configure step. The `make clean` purges any now-unwanted files left from previous builds of the kernel.

Then you are ready to go! Issue this command:

```
make zImage
```

and then, if you are on a machine slower than a Pentium 200, go and make a cup of tea. This takes about 20 minutes on a Pentium 90...the kernel has a lot of source code as you may have noticed when downloading it. When this is complete do a:

```
make modules
```

This will not take as long.

Installing a New Kernel

Phew, finally! The last step is installing the new kernel. On an Intel-based system the kernel is installed to the right place in `/boot` with the command

```
cp /usr/Linux/src/arch/i386/boot/zImage /boot/newkernel
```

then

```
make modules_install
```

This will install the modules in `/lib/modules`. Next, edit `/etc/lilo.conf` to add a section like this

```
image = /boot/newkernel
```

```
label = new
```

```
read-only
```

At the next reboot, select the kernel 'new' in lilo, and it will load the new kernel. If it works fine, move it to the first position in the `lilo.conf` so it will boot every time by default.

Alternatively, if you already have the kernel sources and only need to upgrade one version number (from 2.2.1 to 2.2.2 for instance) then download the patch file to `/usr/src` and then use the command:

```
gzip -dc patch-2.2.2.gz |patch -p0
```

to upgrade your old kernel sources. Patch files are much smaller to download than the whole thing.

Summary

Compiling the kernel is a relatively simple operation- if you have done it before! At first it can seem daunting. There are many Web sites dedicated to the kernel; at the left are just a few.