

Linux Kernel Compile

Introduction

This guide is meant to help those who wish to create a custom kernel at their own home computer, thus allowing working on the course assignments from their homes. The following instructions were tested on my computer, which was running Red Hat Linux 8.0 (GRUB loader) via VMWare, on a windows XP computer.

A similar guide was written by Artiom Myaskovskey, and is supposed to be more general. However, following his instructions, I couldn't make it work, so I tried my own.

I stress again: My knowledge of the processes involved here is very very limited! This is mostly a result of trial and error, and some internet data bases. I do not know if this will work on other configurations (like using LILO), so just try for yourself.

Good Luck!

Getting ready

Note: commands are printed in **green**. Things you need to fill up are in *Italic*.

First, start the graphic environment (called x-windows) and log in as ROOT. Logging as root is not a "must do" for most of the stages, and actually is not recommended by most internet sites, but we'll ignore this for now. If you know what you can or cannot do as a non-root user, be my guest.

After you log in, start a shell window (RedHat icon -> System tools -> Terminal).

Now we need to make a copy of the original kernel. Do this by:

```
cp -r /usr/src/linux-2.4.18-14 /usr/src/linux-2.4.18-14custom
```

Now change into the directory of the custom kernel:

```
cd /usr/src/linux-2.4.18-14custom
```

Compile

The next commands clean out unwanted leftovers, and delete the configuration file from the old kernel:

```
make clean
```

```
make mrproper (config delete)
```

```
make clean
```

Now we'll make a new config based on the old one:

```
cp config file .config (config file = /boot/config-kernel version)
```

```
make menuconfig (choose Load alternate configuration, select the file .config)
```

Now build dependencies tree:

```
make dep
```

```
make clean
```

This is the actual compile of the kernel source. Pay attention to the messages on screen and make sure there are no errors in compilation!

```
make bzImage (compile kernel source)
```

```
make modules (all the extras ñ this will take quite a whileÖ )
```

```
make modules_install (I have no idea what this is, but it sure is shorter)
```

```
make clean
```

This is the last stage. It will copy the kernel image into the boot directory, and will update GRUB so next reboot you'll have a "custom kernel" option.

```
make install
```

reboot

Closing remarks:

- If you get errors during `make bzImage` the first time, you should try to delete the entire directory and start all over again.
- In order to recompile the kernel next time, simply type `make bzImage` in the custom kernel directory, and then copy the new image yourself by :
`cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.18-14custom`

Niv Tokman,

23/1/04