

Debian-Kernel-Compile-Howto (Kernel 2.6)

By Falko Timme

Published: 2004-12-27 18:34

Debian-Kernel-Compile-Howto (Kernel 2.6)

Version 1.1

Author: Falko Timme <ft [at] falkotimme [dot] com>

Last edited 07/15/2005

In some cases you might want to compile your own kernel that suits your needs better than the standard kernel that comes with your distribution. I will describe how to do this on a **Debian Woody** machine (a description for **Debian Sarge** can be found [here](#)!). Please note that this tutorial is for **kernel 2.6 only**! A tutorial for kernel 2.4 can be found [here](#).

Be aware that there is some risk in doing so! For example, it could happen that your machine does not boot properly after you have installed the new kernel so you might be forced to boot from a rescue CD to repair your system. You have been warned! This document comes without warranty of any kind!

Kernel-Compile-Howto

First login to your Debian machine on the command line as root. Install the prerequisites that we need to compile the new kernel:

```
apt-get install kernel-package ncurses-dev fakeroot wget bzip2
```

Then go to /usr/src:

```
cd /usr/src
```

Then get the latest Linux kernel source (or the kernel source you need) from <http://www.kernel.org/pub/linux/kernel/v2.6/>:

```
wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.8.1.tar.bz2
```

Unpack the kernel sources:

```
tar xjf linux-2.6.8.1.tar.bz2
```

```
cd linux-2.6.8.1/
```

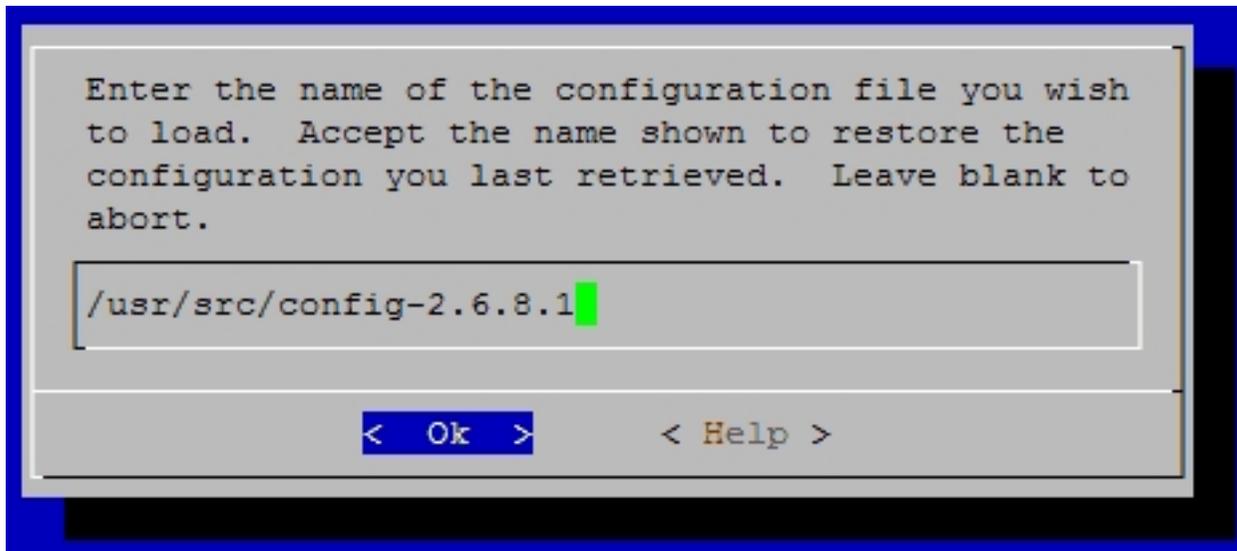
It is normally a good idea to take the configuration of your existing (working!) kernel 2.6 as a starting point for the configuration of your new kernel. Usually the current kernel configuration is saved in a file under /boot, e.g. /boot/config-2.6.3. We will load this configuration and then do the changes we desire. If you do not have the config file of a working kernel 2.6 [this one](#) might help you.

```
make menuconfig
```

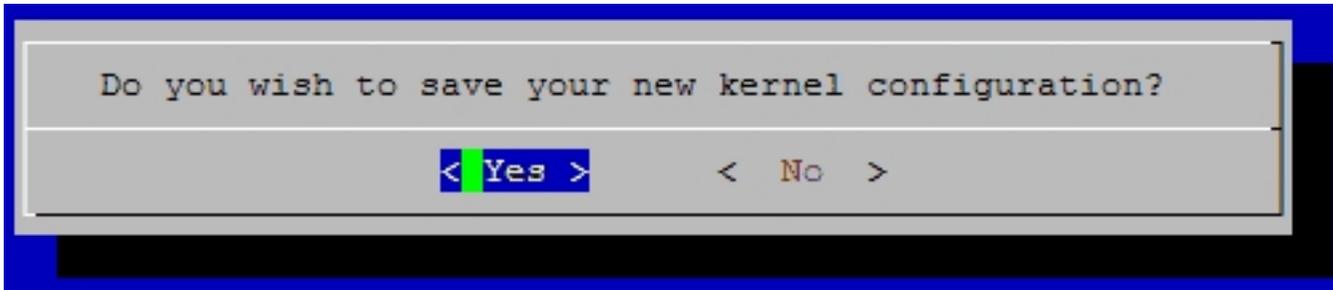
Select Load an Alternate Configuration File and enter the location of the configuration file of your current kernel (or upload [my kernel configuration](#) to your machine and enter its path):

```
----- Linux Kernel Configuration -----
Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> incl
c><Esc> to exit, <?> for Help.  Legend: [*] built-in  [ ] excluded  <M> module  <

Code maturity level options  --->
General setup  --->
Loadable module support  --->
Processor type and features  --->
Power management options (ACPI, APM)  --->
Bus options (PCI, PCMCIA, EISA, MCA, ISA)  --->
Executable file formats  --->
Device Drivers  --->
File systems  --->
Profiling support  --->
Kernel hacking  --->
Security options  --->
Cryptographic options  --->
Library routines  --->
---
Load an Alternate Configuration File
Save Configuration to an Alternate File
```



The configuration of your current kernel will be loaded, and you can now browse through the menu and change the configuration to suit your needs. When you are finished, save your new kernel configuration:



Then run the following commands (please note that `make dep` is not needed any more for kernel 2.6):

```
make-kpkg clean
fakeroot make-kpkg --revision=custom.1.0 kernel_image
```

If the compilation stops with an error, run

```
make clean
```

and then re-run the previous commands starting with

```
make menuconfig
```

Change the kernel configuration where the error occurs. If no error occurs you will find the new kernel as a Debian package called `kernel-image-2.6.8.1_custom.1.0_i386.deb` under `/usr/src`.

```
cd ../
```

Now you have to install some packages that are needed by kernel 2.6. Add the following line to `/etc/apt/sources.list`:

```
deb http://www.backports.org/debian/ woody module-init-tools initrd-tools procps
```

Then run:

```
apt-get update
apt-get install module-init-tools initrd-tools procps
```

If you are asked the following question:

```
"If you really want to remove modutils type 'yes':"
```

type yes.

It might also be necessary to update packages like bind9, quota, etc. - depending on your configuration. If you have problems with your existing packages try to get the appropriate package from www.backports.org.

Install your new kernel:

```
dpkg -i kernel-image-2.6.8.1_custom.1.0_i386.deb
```

Create a ramdisk of your new kernel (otherwise your system will most likely not boot):

```
cd /boot/
mkinitrd -o /boot/initrd.img-2.6.8.1 2.6.8.1
```

We are almost finished now. Edit the `image=/vmlinuz` stanza of your `/etc/lilo.conf` and add the line **`initrd=/boot/initrd.img-2.6.8.1`**:

```
# Boot up Linux by default.
#
default=Linux

image=/vmlinuz
    label=Linux
    read-only
```

```
initrd=/boot/initrd.img-2.6.8.1
# restricted
# alias=1
```

Run

```
lilo
```

to update your boot loader and reboot your machine:

```
shutdown -r now
```

and if everything is ok your machine should come up with the new kernel. You can run

```
uname -a
```

to verify that. Good luck!

Update:

This is an excerpt from an email that [R \(Chandra\) Chandrasekhar](#) (thanks a lot for this one!) sent to me regarding the kernel configuration in case you use flash drives and nVidia cards:

"1. If you intend to use flash drives, de-select the following during <make xconfig>:

```
Block Devices -> Low Performance USB Block driver (BLK_DEV_UB)
```

There is a clear explanation at <http://linux.usc.edu/pipermail/usclug-chat/2004-December/000111.html> and at

<http://dev.gentoo.org/~dsd/gentoo-dev-sources/issues-current.htm#2.6.9-ub> on why this is advisable until further development of the ub module.

I had to

```
apt-get install udev hotplug
```

to use the flash drive, but it worked like a charm after that!

2. If you have an nVidia card, de-select the following during <make xconfig>:

```
Graphics Support -> nVidia Riva support (FB_RIVA)
```

```
Kernel Hacking -> Use 4Kb for kernel stacks instead of 8Kb (4KSTACKS)
```

```
Processor Type and Features -> Local APIC support on uniprocessors (X86_UP_APIC)
```

To install the nVidia drivers, I followed the HOWTO at <http://home.comcast.net/~andrex/Debian-nVidia/index.html> and compiled the drivers from the Debian packages after applying the `vmalloc_reserve.patch` during kernel build."

Original location of this document: http://www.falkotimme.com/howtos/debian_kernel2.6_compile/