

Installing a Kernel in Ubuntu

Are you running Ubuntu 10.10 and wish to install a newer kernel not found in the repository? Are you using the new Ubuntu 12.04 and want to experiment with a developmental kernel? Maybe you simply want to use the latest kernel before it appears in the repository?

Whatever the reason, installing prepackaged kernels in Ubuntu is easy without needing to compile one from source code. Here is how to do it.

A WORD OF WARNING: Backup all critical data before updating the kernel. While problems are unlikely, there is always the risk that the system might become unbootable following a kernel update. The kernel is the heart of Linux, and any form of heart surgery risks death to the patient. Having a backup will ease your mind in case something goes wrong and you find that you are unable to recover your system.

Preparation

Have the video drivers ready in a directory. Installing a new kernel often requires that the video drivers be reinstalled also because the two are closely connected. This is especially true for proprietary video drivers from ATI and Nvidia. Rename the driver's filename into something easily typed from a command line, such as amd.run or video.run. You'll thank yourself later.

Getting a Kernel

The site <http://kernel.ubuntu.com/~kernel-ppa/mainline/> contains many prepackaged kernels free for download for Ubuntu. Find the kernel you want, download the files, and then install them by double-clicking on the .deb files. Ubuntu Software Center will handle the rest.

Kernels exist for 32-bit and 64-bit systems, so make sure that you download the kernel matching your system. If you have 64-bit Ubuntu, then download the amd64 files. If you have a 32-bit system, then use i386 files.

Three separate files are required:

linux-headers_____all.deb (for both 32-bit and 64-bit)

linux-headers_____amd64.deb (64-bit)

linux-image_____amd64.deb (64-bit)

The _____ part of the filename will vary depending upon the kernel version.

Suppose you wish to install kernel 3.2.16 in Ubuntu 10.10 64-bit. Go to the [directory of the 3.2.16-precise kernel](#) and download these files:

```
linux-headers-3.2.16-030216_3.2.16-030216.201204221935_all.deb  
linux-headers-3.2.16-030216-generic_3.2.16-030216.201204221935_amd64.deb  
linux-image-3.2.16-030216-generic_3.2.16-030216.201204221935_amd64.deb
```

If you have a 32-bit system, download the 32-bit files instead. (32-bit filenames end with **i386.deb** and 64-bit filenames end with **amd64.deb**.)

Take note that the kernel directories have Ubuntu release names, such as maverick, oneiric, or precise. This is fine. The kernels will work even if the filenames do not match the name of the installed Ubuntu. For example, the 3.2.16 precise kernel will work in Ubuntu 10.10 maverick.

Installing a Kernel

A new kernel is added to the system by installing the three files and updating GRUB2 (the bootloader).

For ubuntu 10.10 64-bit, install the three files in this order to avoid any dependency issues:

- 1: linux-headers-3.2.16-030216_3.2.16-030216.201204221935_all.deb
- 2: linux-headers-3.2.16-030216-generic_3.2.16-030216.201204221935_amd64.deb
- 3: linux-image-3.2.16-030216-generic_3.2.16-030216.201204221935_amd64.deb

Remember, the order is,

1. **linux-headers_____all.deb**
2. **linux-headers_____amd64.deb**
3. **linux-image_____amd64.deb**

Install all files without rebooting.

Update GRUB2

After all three files are installed, open a terminal and enter **sudo update-grub** to update GRUB2 and make the kernel available for loading from the GRUB menu. Updating GRUB2 is most important, but easy to forget.

Reboot

After updating GRUB2, reboot the system. Normally, software installation never requires a reboot, but since we are dealing with the kernel, the heart of a Linux system, a reboot is necessary.

The system should reboot into a GUI with the new kernel active. To check, open a terminal and enter **uname -a**. You should see Linux distribution information like this,

```
Linux CompName 2.6.35-32-generic #67-Ubuntu SMP Mon Mar 5 19:39:49 UTC 2012
x86_64 GNU/Linux
```

If the kernel version displayed matches the kernel you just installed, then congratulations! You are now running the newly installed kernel. If you installed an older kernel, you may need to select it from the GRUB boot menu to load it since GRUB automatically loads the most recent kernel by default.

Oh, no! All I see is a terminal!

If the system boots to a terminal and all attempts to switch to a GUI fail (**Ctrl + Alt + F7**), you might need to reinstall the video drivers. This sometimes happens with Nvidia and ATI Catalyst drivers.

First, log in and check that the new kernel is running by entering **uname -a** at the command prompt. Assuming that the new kernel is properly installed and running, locate the video driver from the command line and install it using something like **sudo sh ./driver.run** (or whatever the manufacturer's installation dictates). A text-based installer will appear and guide you through the video driver installation.

Reinstalling the video driver usually restores the GUI. Following the video driver installation, reboot the computer. The GUI login screen should appear. You may need to enable the desktop effects again if they are not working.

Removing a Kernel

Did you install the wrong kernel by mistake? Is an existing kernel old and outdated? Kernels can be removed like any other software. Here is an easy way using the Synaptic Package Manager:

1. Open the Synaptic Package Manager (**System > Administration > Synaptic Package Manager**)
2. Enter the kernel version in the search box, such as 2.6.35
3. Remove the three 2.6.35 kernel files by checking **Mark for Complete Removal** for each. Click the **Apply** button to make the changes take effect. Many more 2.6.35 files might exist with extra numbering, so take your time selecting the files having the same numbering.
4. Update GRUB2 from a terminal using the command **sudo update-grub**
5. Reboot

With the kernel removed from the system, it will no longer appear in the GRUB boot menu.

Why Would I Want to Update the Kernel?

Newer kernels contain support for newer hardware and include improvements and bug fixes. If there are no mysterious problems occurring on your system and you are happy with it, then leave the kernel alone. For those who find that some hardware is not working in Linux or who crave running the latest software to score bragging rights, then go ahead.

But the main reason is hardware compatibility. Kernel 3.2.16 supports hardware not recognized by kernel 2.6.35. For example, if hibernation mode is not working on a new computer running Ubuntu with kernel 2.6.35, then upgrading to kernel 3.2.16 should enable hibernation. This is not guaranteed, but you increase your chances. Sometimes, some cutting edge hardware is so new that even the most recent kernels do not support it yet.

Which Kernel Should I Use?

Always use the latest production release for best results unless you have a need to use a specific kernel. A production release is one that is considered stable for the Linux world, and it often contains the latest bug fixes and best hardware support.

About Kernel Version Numbers

Browsing through the [kernel list](#), you will see that many versions are available.

“How do I know which one to use?”

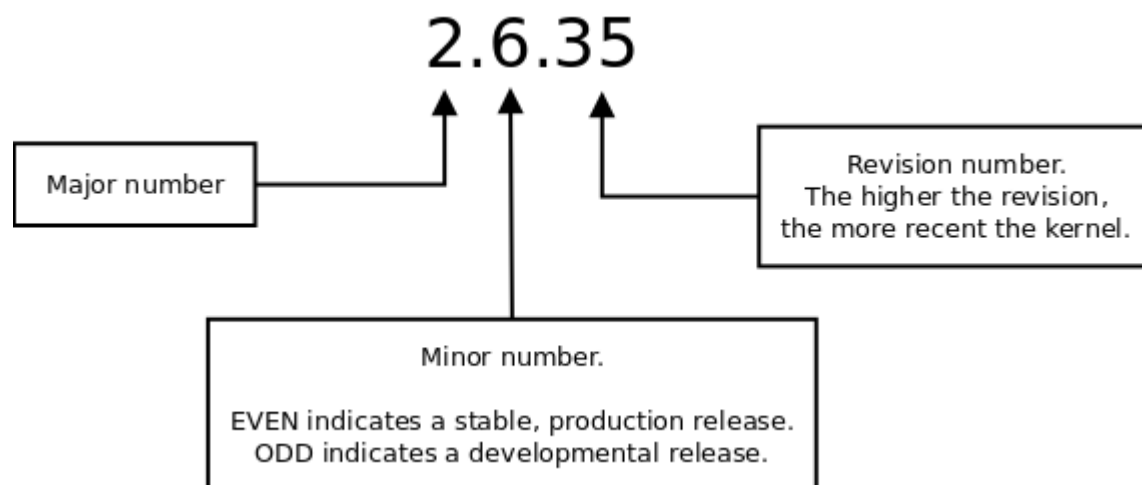
Look at the kernel version. This is a series of three numbers separated by dots, such as 3.3.3, 2.6.35, or 3.2.0. Other numbers and text might exist, but the version is the most important.

The kernel version consists of three parts:

Major number (First number)

Minor number (Second number, Indicates a production or developmental kernel)

Revision number (Third number, indicates revision)



The major number is first. A kernel with a major number of 3 is more recent than 2. The second number is the minor number, and it indicates either a developmental kernel or a stable production kernel.

- ♦ If the minor number is **even**, then the kernel is a **production kernel** intended for everyday use.
- ♦ If the minor number is **odd**, then it is a **developmental kernel** still under development and testing. Developmental kernels are not ready for release although you are free to install and try them.

In the 2.6.35 example above, the minor number is 6, which is even, so 2.6.35 is a stable production release, not a developmental kernel.

The revision number, the third number, indicates the kernel release. Kernel version 2.6.35 has the revision number 35, which means it is the 35th release of the 2.6 kernel. Kernel 3.2.16 has a revision number of 16 making it the 16th release of the 3.2 kernel. The latest revision is the most recent, so kernel 3.2.16 is more updated than 3.2.15, for example.

Conclusion

While other command-line kernel installation methods exist, this provides an easy way for those shy about kernels to update Ubuntu 10.10 (or any other version) to a newer kernel and enjoy its benefits without upgrading to Ubuntu 12.04 and Unity or compiling the kernel from source.

The only major issue most likely encountered involves booting to a terminal instead of the GUI. If this happens, reinstalling the proprietary video drivers under the new kernel usually restores the GUI.

Upgrading the kernel may resolve hardware incompatibilities and fix bugs, but if your system is already running the way you like, then you might not want to risk a kernel upgrade. Otherwise, back up your data and breathe new life into your old Ubuntu with a kernel upgrade.

Source : <https://delightfullylinux.wordpress.com/2012/05/01/installing-a-kernel-in-ubuntu/>