

Discussion on building Linux kernels

Feel free to comment on this Wiki page. Do you have suggestions to make it better? Critical remarks? Leave them here.

— Eric

Are you sure merging the xauth is necessary? I tried it and had to step out for a minute. Gnome locked the screen automatically and since then X hasn't worked, and I get an error saying "error in locking authority file /home/name/.Xauthority" Not sure if this is a result of merging the .Xauthority file or not. But I have also never seen those commands used in any other FAQ I have used.

~David

Hi David - I guess you did the merge while you were in X, then? The safe way would have been to do this in a console... before starting X.

The fact that it is not mentioned in other manuals and FAQs does not mean it does not work 😊
As for your problem - look at the /home/name/.Xauthority file's ownership and permissions. It should be owned by your account, not by root, and it should have 0600 permissions (readable/writable only by you).

— Eric

Found typo: '... a X terminal...' / Should be: '... an X terminal...'

But did it really need **six** revisions of your feedback line to tell me this 😊 That is what the [Preview] button is for...

— Eric

So sorry about that. When it comes to editing: can't help myself, mate :)

Spotted a formatting typo. Look under **Modifying lilo.conf**

The first code block in the section says:

```
label = linux read-only # Non-UMSDOS filesystems should be mounted read-only for checking
```

It should be two lines:

```
label = linux
read-only      # Non-UMSDOS filesystems should be mounted read-only for checking
```

Perhaps a rogue `Return` key didn't show up? It was probably taking some unwarranted holiday somewhere or other. It might even be slacking somewhere else on this site for all we know. Hey, in the ChangeLog recently Pat lost a '['. "It had fallen off my desk and ended up under a table". That's always a pain.

Nice guide Eric. Thanks for the rest of your good work too. I appreciate you highlighting the point Linus made (the infamous `/usr/src` 'zombie' instructions), and the fact that it's a 'non-issue' with Slackware.

However I still think Linus has a point. If nothing else, juggling chainsaws when you're tired, and you've done it a million times before, **could** be hazardous to your health 😊

- mjc -

Thanks for spotting this mjc, I fixed it. The kernel sources location is a hot topic on [LQ nowadays](#).
— Eric

Hello Eric/Alien;

Thanks to this guide, I 'slacked my box from 2.6.27.7 to 2.6.29.1 this morning. You can't refrain from holding your breath while booting but it went fine. Almost:

The nvidia self-compiled kernel wasn't gathered from the 'check your other compiled modules' command;

It's obviously arch-stupid of me to have forgotten about this one, what with the nvidia logo popping up in my face every morning, but a warning to keep some sources at hand in some particular cases may come handy.

As examples that come to mind, there is the Proprietary graphic drivers like ATI or NVIDIA and the VirtualBox module.

I am suggesting, if my system wasn't wrong and if it is normal for the new kernel to struggle to get to X with an `Xorg.conf` claiming for a proprietary module from an older kernel, that you edit the relevant part more or less like this:

Other packages that contain kernel modules

Most certainly you will have packages installed that contain kernel modules that are not part of the default kernel. Slackware has "alsa-driver" for instance, and if you installed a proprietary graphic driver or any wireless driver, these are basically kernel modules too. Now, with the installation of your new kernel, you will lose these modules, and you have to recompile the sources so that the binary modules match the new kernel.

You can get an overview of some packages that have installed a kernel module for your current kernel

by running this command (i.e. you must run this command while still running your old kernel):

```
> cd /var/log/packages  
> grep -l "lib/modules/$(uname -r)" *
```

Warning: not all of them will appear here; typically, for instance, Nvidia or VirtualBox won't!

The mentioned packages will need a recompile, which isn't a chore unless for your graphic driver since you will end up in a terminal, with X respawning in a loop, upon your first reboot. Be sure to have your driver's source at hand (a good opportunity to download the latest version), with the necessary documentation at the ready, to recompile and be able to start your graphic environment again.

For ALSA you have a choice: either enable the ALSA driver that is part of the kernel you've just downloaded, or leave the kernel configuration like Slackware's: disable all ALSA support in the kernel and instead re-build the alsa-driver package. The 2.6 kernels of Slackware 12.2 have all the ALSA drivers built-in because they will not work with the ALSA driver releases you can install separately.

That's it, thank you & sorry for the bother/clutter if it's irrelevant: I run Enlightenment DR16 from SLiM so it may be slightly a-typical. Cheers. Jean-Philippe 2009/04/07 07:18

Hi Jean-Philippe.

Good idea there, I've added a clear warning sign in that section. Thanks,
— Eric

Hi Eric,

Text under the "Building your kernel" heading says make all builds "vmlinuz (the uncompressed kernel)". This should read "vmlinux (the uncompressed kernel)", no?

Thanks for the article.

— Dan 2011-08-05

Kernel key ID ended

Hi Eric,

the kernel key ID is changed. In the square, you write 0x517D0F0E, but it's ended. Now it's 6092693E.

Regards

Davide

Hi Eric,

it's always a good idea to do a “make mrproper” or at least “make clean” before building the kernel. Never expect a source tree to be clean. This applies actually to any software you build from source. Thanks for this article.

Regards,

Heiko

From:

<https://wiki.alienbase.nl/> - **Alien's Wiki**

Permanent link:

<https://wiki.alienbase.nl/doku.php?id=wiki:talk:linux:kernelbuilding>

Last update: **2016/12/30 13:50**

