

Re: Linux drivers management

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-02/msg02059.html>

- *From:* Jesper Juhl <jesper.juhl@xxxxxxxxxx>
 - *Date:* Mon, 6 Feb 2006 20:46:37 +0100
-

On 2/6/06, David Chow <davidchow@xxxxxxxxxxxxxxxxxxxx> wrote:

Please read Documentation/stable_api_nonsense.txt in your copy of Linux source.

I've read the document, I strongly disagree, because it is not relevant to my question or to my original purpose of this question.

It has a lot of relevance.

If you split up drivers and the core kernel into two separate projects I can easily imagine the two getting out of sync whenever an API change needs to be made.

Currently whenever someone changes a kernel API that person is required to also update all users of that API. Since all users are in-tree that's reasonably easy to do. If the drivers are out-of-tree it's a *lot* harder for several reasons;

a) to fix users of the API I can no longer just 'cd drivers/' I now have to go download & extract a separate source tree.

b) if an API change I make gets accepted by the 'core kernel' team but the 'driver-tree' maintainers refuse my updates we get out-of-sync – just as if the 'core kernel' tree rejects the API change but the 'driver-tree' maintainers already merged the update to the drivers to use the new API.

c) presumably the two projects 'driver-tree' and 'core-kernel' would use separate mailing lists, making it harder to communicate about API changes.

Maintaining the drivers out of tree would make API changes a lot more painful and thus we'd need to lock down the API a lot harder than we do today. This in turn would mean that some types of bugs will be harder to solve (since a proper fix would involve a painful API change) and advancement of kernel capabilities/features etc will be slower.

Re: Linux drivers management

Putting the driver source code in the kernel source tree has nothing to do with talking about a stable kernel API . Even you put the driver sources into the main kernel tree, it will still need a lot of work to port all drivers if the API changes.

You can't dodge the work that needs to be done when API's change, but you /can/ try to make the work as easy to do as possible by keeping the core kernel code and the drivers close together and maintained as a single unit.

Driver sources can still host in a different project (e.g. projects in sf.net) and maintain open-source and om by the community, no difference than before

For different compile time options that affect data structures, this is well known a bad idea . These types of techniques no longer allowed in Java and other OO languages .

The kernel is written in C. What other languages do/allow/recommend etc is irrelevant.

Because I can simply say the code is not portable. If really need a recompile and optimize, the distro vendor should bare this, but according to the document, "As Linux supports a larger number of different devices "out of the box" than any other operating system" , do you think Linux should one day or some day grow to 1TB source tree to include all possible drivers for all hw come from the world?

First of all, drivers are regularly dropped from the tree; either because they become unmaintained and bitrot or because the hardware becomes extremely obsolete and rare.

Secondly, I predict that available storage sizes and bandwidth available to users of the kernel will grow faster than the size of the source tree (and cost of storage & bandwidth will likely continue to drop as well).

Thirdly, the day Linux supports "all hw come from the world" I'll be dancing with joy.

Re: Linux drivers management

I don't see there is reason why a kernel or OS need to include all the drivers for all the hardware.

This is **extremely** convenient for our users.

I don't think there is any OS vendors on the market to capable to distribute all drivers integrity, then the choice is to make a disabled Linux OS because of an OSV who has only limited supporting resources to support and certify limited hardware devices.

The user always has the option of building any driver they need themselves. This is easy since the drivers are all in the main kernel source tree, the user doesn't have to go hunt for them online (assuming the user even has an internet connection).

Please see my other email responded to Jes about the learning curve and documentation issues of a Linux driver developer to pick up Linux skills.

The fact that all documentation (well, at least a lot of it) is kept in a central place (Documentation/) is a very nice thing for someone trying to learn their way around the kernel – I know I've personally bennefitted from that.

Also the fact that core kernel code, drivers and supporting scripts are all kept in a single source tree is very convenient for new developers since there's only one thing to download and then you have a complete tree with the full picture to learn from – again I say this based on personal experience.

But in any case, if you want to maintain one or more drivers out-of-tree, then go ahead, noone's stopping you. But your maintenance work will be a lot lower if you instead submit your drivers for inclusion in mainline since then other people will keep your driver up-to-date and in sync with API changes when they happen. You might even get a few bugs fixed for free and get the code cleaned up for free.

—

Jesper Juhl <jesper.juhl@xxxxxxxx>

Don't top-post <http://www.catb.org/~esr/jargon/html/T/top-post.html>

Plain text mails only, please <http://www.expita.com/nomime.html>

—

Re: Linux drivers management

Re: Linux drivers management

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>