

How not to piss off a kernel subsystem maintainer - part 4

RSS

Third in a long series of complaints... See [part 1](#) and [part 2](#) and [part 3](#) for previous atrocities.

It seems that [some people are now worried](#) that if I critique a patch a bit harshly, they will end up being mentioned here.

Now to be fair [this patch](#) does contain some obviously wrong things (wrong coding style, linewrapped patch, no tabs in sight, odd types created (`sysfs_ref?`), `#define NULL`, did not CC: the proper subsystem maintainer, etc.) that should have never escaped into public to see the light of day (come on, a STSM making such basic mistakes makes me really wonder...), but it doesn't deserve to make me or anyone else "pissed off".

In fact it's good that he posted the code, and the issues that are being raised are good ones, as people obviously try to subvert sysfs to do bad things at times, as they are just not that familiar with it and are used to path based filesystems like `/proc` where you can just say "give me a file named `foo/baz/bar` and it will be created automatically for you.

For the longest time, I felt that it would be good to be able to create subdirectories easier in sysfs, but after the ability to name attribute groups happened (you do know about sysfs attribute groups, right? Yeah, no one does it seems...), that need really went away.

For Xen, they want to move all of their `/proc/xen` files into sysfs easily, as they know it's not acceptable to add new files to `/proc` that do not deal with processes. Unfortunately, putting those files into sysfs isn't a simple api change. It requires you to usually rework your logic and create some kind of tree that can be properly represented. This patch was made with the goal to put a procfs like api on top of sysfs, and do this only for xen, both of which are not acceptable.

Well, the later is not acceptable at all, it remains to be seen if the former is really a good idea or not. But if it is, it should not be restricted to only Xen users, but available to all.

The main point here is, raising these types of questions, and doing it by providing a real example in code, is the absolute best way to do kernel development, and makes me, as a kernel subsystem maintainer, very happy to see.

That all being said, the Xen people really need to work on getting their stuff into mainline as they have been playing in their own repository for way to long and are not used to having anyone review their code. Hopefully some day they will learn and join the mainline kernel.

Until then, I expect to see more posts like this (there was another one recently on the linux-usb-devel mailing list that also ended with that developer having to go off and do some general work that everyone can benefit from), as the xen development group slowly learns the proper way to interact with the kernel developers.

This post is brought to you today by the letters I, B, and M.

posted Wed, 11 Jan 2006 in [[/linux](#)]