

Re: legacy platform drivers and hotplugging

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-08/msg08621.html>

- *From:* Kay Sievers <kay.sievers@xxxxxxxx>
 - *Date:* Mon, 20 Aug 2007 11:51:35 +0200
-

David,
your change to the platform bus disables uevents for almost all platform devices. These events are needed and expected by userspace today, just like they are for all other devices of the kernel too. The device event sequence now has holes in the chain of events, which breaks setups that depend on that.

Udev coldplug is also broken:
udevtrigger --subsystem-match=platform
leads to nothing after your change, while it worked as expected before.

Userspace will not code around your very personal idea of hotplug. You can not disable driver core events globally because *_you_* don't need it. The rest of the world needs them and wants them back. Upstream hotplug works differently than you expect, and your change broke that.

My patch fixes all these issues, and make platform behave like a nice citizen of the kernel, make it look like the rest of this small world, without any special treatment.

Please take care of that or I will file a regression report with a revert request for your change, to get userspace back into a working state as it was before.

Thanks again,
Kay

On Sun, 2007-08-19 at 18:44 -0700, David Brownell wrote:

Notice I fixed \$SUBJECT to not mention rtc-ds1742, which has never been a platform device and thus is offtopic to your specific complaints about platform bus hotplugging.

On Saturday 18 August 2007, Kay Sievers wrote:

On Fri, 2007-08-17 at 21:06 -0700, David Brownell wrote:

Re: legacy platform drivers and hotplugging

On Friday 17 August 2007, Kay Sievers wrote:

I'm not the one who's
advocating a change here. If
you want to
first change/break and then
fix things, all of that is up to
you.

I'm happy to do that. Patch is attached.

NAK. That wasn't even a serious attempt at the "fix" part,
though it does the "break" part well enough to cause severe
regressions.

You disabled uevents which breaks udev and HAL setups, because we can't
track the existence of the devices.

Only certain code paths -- ones primarily used by legacy
drivers -- disable those events.

The basic problem with such legacy drivers is that they
violate core driver model assumptions ... notably, by
creating device nodes for non-existent hardware.

Which means that for these legacy drivers, userspace can't
track device existence in *that* way.

If said driver creates some other device node after it's
verified the hardware exists -- maybe one that's coupled
to a character device -- it'd be fine to track *those*
events. ISTR that most such drivers *do* create such nodes,
since back in the prehistoric ooze from which they crawled,
that was the only way to be seen by userspace.

You can't just disable device events
to disable module loading. Uevents have are by far not only about module
loading.

Only for legacy drivers, because of the modprobe loop thing.

So your proposal for how to handle such broken/legacy drivers

Re: legacy platform drivers and hotplugging

is ... exactly what, then?

Near as I can tell, you propose that the platform bus never support automatic module loading. Which seems to me like a nonstarting proposal: a significant and needless regression, inflicting harm on what for most platforms (by type, and also maybe by number-in-the-field) is the primary system bus.

(As well as leaving all my technical points about your pushback un-addressed. As I noted before, the evident fact that you don't have technical responses to them says to me that your pushback has no real technical basis ...)

It has. Disabling uevents to control module loading is just the totally wrong thing to do. Uevents are there to let userspace know that a device exists.

See above. By your definition, legacy drivers would be issuing bogus events. So one key issue here is how to handle them; the patch you object to won't issue those buggy events.

From you, I've not seen even a proposal for a viable workaround to that looping bug. Not even that udev configs should handle it, for example, although that would appear to be the most natural conclusion given your premises and the fact that ISTR you're still maintaining it ...

Whatever happens, I believe that legacy drivers will necessarily involve special casing to cope with the fact that they don't follow the current set of rules for drivers (though some folk seem to want to make them act as if they do).

Only the existence of MODALIAS and the "modalias" file controls module loading, not the enabling and disabling of uevents, which is a completely broken hack.

Well, ISTR my original patch for the "looping" bug only affected those values, but you objected to that too...

Which is why the code now just uses a pre-existing "no uevents" hook (on code paths used primarily by legacy drivers) instead of

Re: legacy platform drivers and hotplugging

only addressing the "this driver can't be hotplugged" issue.

It looks like you're maybe coming around to my original position, at least that key part of it.

My patch fixes tons of issues, and that is "technical basis" enough. It makes platform play nice with userspace, by behaving like the rest of the kernel.

Yet as I noted: it causes a regression of just shy of 100% for platform device hotplug. Ergo a well-founded "NAK" ... two steps forward versus hundreds of steps backward, rarely a good idea.

(You still haven't actually described any case where the current code is a problem. Handwaving "tons of issues" doesn't count.)

Userspace udev/HAL relies entirely on uevents, for hot- and for coldplug. Coldplug is done by writing "add" to all "uevent" files during early boot, that does not work anymore with platform, and needs to be fixed.

Like I said: "technical basis" enough not to merge your patch; it breaks coldplug too, as well as hotplug.

Out of around 300 platform drivers in the tree (and many more not yet merged upstream), this makes it so that only *THREE* of them can hotplug ... versus in the current tree, essentially everything that's not a legacy driver is hotplugging just fine.

That's one heck of a regression. Just shy of 100% ...

So where are the 100's of drivers that send an uevent? I'll go and fix them.

Device creation sends those uevents, as you know, not drivers. (Modulo the issue of broken/legacy platform drivers, which are wrongly creating devices themselves. When I counted before, I recall thinking that only a few dozen drivers were broken in that way. I mentioned i82365, there are others.)

Re: legacy platform drivers and hotplugging

What you broke is the connection between the device "add" events and the "modprobe" parameter which loads the relevant driver. That currently (i.e. without your patch) works for most platform drivers.

Plus it treats rtc-ds1742 as if it's a platform_driver not an i2c_driver.

Wrong, the whole model is the other way around.

Please strike yourself repeatedly over the head with a clue-by-four. I2C != platform, and ds1742 is an i2c device. So your patch:

```
+MODULE_ALIAS("platform:ds1742");
```

violates the (new/undocumented) rule *YOU* previously proposed (which is violated by most MODULE_ALIAS declarations I found in the source tree) whereby the syntax of those aliases requires that they start with a subsystem prefix.

Wrong subsystem. strcmp("i2c", "platform") != 0.

Please stop doing adding weird hacks to the kernel to fit your personal taste and breaking all assumptions existing tools rely on.

The only "personal taste" I've observed here is yours: that in regards to this issue, you've avoided constructive responses to most issues I've raised ... more than once giving unfounded ad hominem "responses" instead of providing a counter-argument (such as evidence backing some of your assertions).

Fix the stuff you broke and enable uevent for _all_ devices again. And let userspace make the decision what to do with the event, like all other subsystems in the kernel do.

You know, all I was trying to do was fix a bug reported to me with the legacy i82365 driver. (The modprobe loop thing.) That's what started your flamage.

Re: legacy platform drivers and hotplugging

Root cause was that it's a legacy driver, can't hotplug, and shouldn't even try. There are other legacy platform drivers in that same boat. So the fix couldn't apply to just that one driver.

Now, if you had ever come up with a viable alternative fix to that issue (patch or proposal) rather than just arguing against actually fixing that nasty bug ... well, such "what-ifs" can't matter a lot here. Except that constructive discussions don't feel like "talking to a brick wall" time-wastage.

– Dave

–

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