

Re: ANNOUNCE: CE Linux Forum – Specification V1.0 draft

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-05/3720.html>

From: Mark Gross (mgross_at_linux.jf.intel.com)

Date: 05/18/04

To: Christoph Hellwig <hch@infradead.org>, Mark Gross <mgross@linux.jf.intel.com>

Date: Tue, 18 May 2004 12:32:48 -0700

On Monday 17 May 2004 23:48, Christoph Hellwig wrote:

- > *On Mon, May 17, 2004 at 01:42:49PM -0700, Mark Gross wrote:*
- > > *All that these "organizations" are doing is collecting REAL requirements*
- > > *for features that REAL application developers need. As well as putting*
- > > *up resources to enable the features.*
- > >
- > > *These features represent input from real application developers and*
- > > *system integrators on requirements that would be cool for their*
- > > *applications if Linux supported them. Why not look at them from the*
- > > *"what features are missing from Linux today and by whom" point of view?*
- >
- > *Have you ever worked in a real life software shop? For every given feature*
- > *X you can find N customers X1...XN that will declare they want this*
- > *feature. Even more for every given customer Y you'll always find M feature*
- > *he'll absolutely want. Until you tell him it's not in the next version*
- > *unless he provides funding where the feature set usually gets shrunk down*
- > *a lot.*
- >

I have worked in a number of real life software shops. The more effective shops I've worked in have been those where the developers were open minded enough to read the "requirements" to extract the true needs.

I'm sensing you are not so open minded about recognizing requirements and needs.

- > *Even the most clueless project manager gets this, and the engineering*
- > *response cuts down this list a lot.*
- >

so.

- > *With many of the large opensource projects this traditional scheme pretty*
- > *much falls flat because for every given feature F there's always N broken*
- > *implementations and the 'project managers' solution is rather to filter*

> *what features are usefull enough and what implementation is good enough.*
>

It seems to me that these organization are doing just this. Those features or requirements that stick across major revisions of such specifications and become more stable are likely the usefull enough features with good enough implantation's, that are worth looking at.

> *CELinux doesn't help either job, it's just listing the implementation*
> *details of the broken patches of a bunch of companies with vested interest,*
> *barely mentioning the requirements they try to fullfill.*
>

Yeah, the decomposition of the features/implementations to requirements does need work in these things. CELF is a new group. However; I think CGL isn't so bad at this.

> *You'd make our life a lot easier by first writing down the requirement,*
> *the thinking of soultions instead of taking the one \$MEMBERCOMPANY*
> *proposed, where that thinking shoould involve talking to the mailinglists*
> *for that area and finally proposing a patch describing _the requirement_.*
> *not what you've done. CELinux, just like CGL or DCL has very much failed*
> *this procedure.*
>

I agree that patches should get posted with requirement and implementation detail in addition to the patch itself. However; I see a lot of patches that provide neither posted by every one.

> *So far I can't see CELinux as anything but a useless specication tricking*
> *PHBs into buying a products of the member companies because they're*
> *following a specification (of which \$PHB of course doesn't know how useless*
> *it is)*
>
> *The patches do get submitted on a regular basis to the LKML. Many seem*
> *to get ignored.*
>
> *If patches aren't even discussed on lkml it means you've done something*
> *very wrong. I don't really remember any of the patches you submitted on*
> *lkml.*

I didn't post it.

>
> *But let's take one of the patches, the first one I looked at on your wiki*
> *apart:*

Lets not.

This about requirements, missing features and needs aplication developers have. The example below is simply one patch to enhance the kenrel latency

for applications using IDE drives. Its also an old patch for an older kernel.

This topic is not about implementation pulled out of some embedded linux kernel hack implemented by some CE company in the past to work around latency issues with a base kernel that didn't have good kernel preemption behavior.

This topic is about features that are worth considering and enhancements needed by your users.

The follow snippet is clearly an implementation attempting to fullfill some type of latency requierement that the IDE implementation was blocking.

Actually I think this code may be related to boot time reduction. It may be worth looking at the CELF specification to see if there is a boot time reduction section.

```
>
> --- linux-2.4.20.orig/drivers/ide/ide.c Thu Nov 28 23:53:13 2002
> +++ celinux-040213/drivers/ide/ide.c Thu Feb 12 10:25:12 2004
> @@ -2739,12 +2776,17 @@
> */
> void ide_delay_50ms (void)
> {
> +#ifdef CONFIG_IDE_PREEMPT
> + __set_current_state(TASK_UNINTERRUPTIBLE);
> + schedule_timeout(1+HZ/20); /* from 2.5 */
> +#else /* CONFIG_IDE_PREEMPT */
> #ifndef CONFIG_BLK_DEV_IDECS
> mdelay(50);
> #else
> __set_current_state(TASK_UNINTERRUPTIBLE);
> schedule_timeout(HZ/20);
> #endif /* CONFIG_BLK_DEV_IDECS */
> +#endif /* CONFIG_IDE_PREEMPT */
> }
>
> This great piece 'called IDE-preempt' to be buzzword-compliant is (and
> that's noticeable just from looking at the diff!) so braindead that it's
> not explainable by incompetence alone. You'd get your same result by just
> _disabling_ CONFIG_BLK_DEV_IDECS instead of adding another broken config
> option (modulo 2.6 adjustments to the sleep time).
>
> Every engineer with the slightest clue would first disable that option, or
> if ide-cs support is actually needed think _why_ it's different instead of
> just adding a config option to disable it. Either it's safe to always use
> the sleeping variant in which case the original ifdef should go away, or
> it's not in which case your patch is completely broken.
>
```

OK I'll bite, but just because in your blind hostility and haste you've made a mistake ;)

Just taking this code out of context (typically a bad thing to do) I would say you would like to have CONFIG_BLK_DEV_IDECS enabled. Note the "#ifndef" to avoid the mdelay call. However; this wont help much if you don't have your IDE drive connected to a PCMCIA adapter and you are using the card services driver.

Most REAL software shops that ship product end up having to make such hacks to enable critical paths for applications and avoid the implementation of more general solutions based on engineering costs. This patch is clearly a hack to enable partial or enhance kernel preemption on the ide path.

These are some of the types of problems engineers at REAL software shops have to solve to be able to ship REAL product for REAL money. If you haven't HAD to produce code like this yourself at some point in your career then you've lived a sheltered life.

Its disingenuous for you to get on your ivory tower to point and laugh.

> *So if I can give you guys from the various industry consortia some hints:*

>

> *o think before you code*

> *o don't drink and code*

> *o get a clue*

It would be cool if you could be a bit more open minded about what these organizations are doing and consider the requirements and the needs they are trying to address.

–

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>