

## Re: what's next for the linux kernel?

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-10/1617.html>

---

**From:** Michael Concannon (*mike\_at\_concannon.net*)

**Date:** 10/06/05

Date: Thu, 06 Oct 2005 17:53:40 -0400  
To: Luke Kenneth Casson Leighton <lkcl@lkcl.net>

Luke Kenneth Casson Leighton wrote:

>On Thu, Oct 06, 2005 at 04:13:15PM -0400, Michael Concannon wrote:  
>  
>  
>  
>>1. It\_is\_a file: registry.dat  
>>2. It is a binary file at that...  
>>3. That file has become a dumping ground for everything that every app  
>>thinks is "important" and of course every app writer thinks everything  
>>they write is the most important thing ever – I am sure a have never  
>>done such a thing :-)  
>>  
>>  
>  
> s/"that file"/"openldap" and substitute "every app writer"  
> for "every major free software developer we respect greatly which  
> can store its data and/or configuration details in an LDAP database"  
> and your evident distaste for "that file" looks a little like religious  
> zealotry.  
>  
>

I don't believe in religion :-) That is not to say I am atheist, just don't see the distinction between one mythology and the next... but that really is another thread...

As for your prior comment regarding mounting the registry as a filesystem, I did see that and made a book mark for the next time I have to recover an NT box... thanks :-)

However, I am now a little confused though (ok, I am always a little confused – see comment above on religion – now I am really confused).

If you concede that it need not be a binary file and it need not be centralized, whether by mounting a "filesystem" or not, then what are we talking about? That is what we have with /etc /proc /sys?

## Linux–Kernel: Re: what's next for the linux kernel?

Assuming you fix the issues of easy of editing it in "dead" filesystems, binary corruption, permissions and programming style of dumping crap in there, then I guess I could care less how that is implemented, proc is already a "virtual filesystem"....

I think someone already mentioned that the issue is that the delta between an idealized NT registry (which has a few notable hurdles – see above) and what we have to day is simply a matter of "KISS". What do you gain from complicating the system that cannot be gained with visualization tools on top of what is there?

Someone wants XML in /proc? Well, that's just fabulous they can write a virtual filesystem that accomplishes that on top what is there and leave the rest of us out of it :-). If what they do becomes indispensable for a critical mass, then even better, we mount "xmlprocfs" in our future systems and are fat dumb and happy.

Back to your original point which seemed to be, at least to me, to try to re–evaluate the portioning problem between Hardware/Software/Drive/OS/User/Threads. I agree, that the system appears to be strained and chaotic with all OSES chasing an ever increasing and impossibly large array of hardware and all–the while the future is even more complex as it seemingly must be a heavily parallelized future to compensate for the "end of Moore's law". Given the hardware in question, though, I am not sure I that I see that Linux should go to micro–kernels to solve the problem...

<rant>

It seems to me that the driver for "correcting" this is actually closer to the hardware side... I am flabbergasted as a hardware engineer that at this point in time with the time elapsed between today and the first PCs that things have evolved so little...

"drivers" should be the exception not the rule...

Few gadgets architecturally do anything different than anything else in that class of gadget to really require a driver that could not be standardized. Gadgets need user–space applications, but with all the well defined standards we have for talking to devices, there is no excuse with the wealth of compute power and storage that we have that we (the entire PC industry) are still shipping hacked, one–off drivers wither every new gadget...

The same applies to the x86 instruction set – waded through that beast (well all N volumes of it) recently? WTF? Its as if 199Million of those 200M gate chips are devoted to obfuscating the user interface.... Most of those bits had a purpose at some point, but we don't seem to be converging to simpler interface...

I would fire me if ever I even proposed such a horrible design... but then I don't design x86 processors...

Re: what's next for the linux kernel?

## Linux-Kernel: Re: what's next for the linux kernel?

If the CPUs and associated hardware started providing a more pleasant interface, then I am certain that Linux would respond by taking advantage of it... We aren't there yet...

</rant>

/mike

> *i say that with the greatest respect.*  
>  
> *especially when "that file" is actually a database, just like*  
> *Berkeley DB (and we all know and \_love\_ Berkeley DB).*  
>  
> *and especially in light it being possible to do a "decent" job, and*  
> *make "that file" available via a POSIX filesystem interface.*  
>  
> *l.*  
>  
>  
>  
>> *I guess you could argue that #3 is the fault of the app writers and not*  
>> *the architecture,*  
>>  
>>  
>  
> *yes. i would say it's more to do with the dumb-ass nature of the app*  
> *writers, yes. typicall dumb-ass windows app writers give a shit about*  
> *security and care greatly about making money hand-over-fist.*  
>  
> *whereas on linux it's far less likely for an app writer to*  
> *be able to get away with a) making money b) friggin up security. the*  
> *distros wouldn't allow an app writer to get away with either.*  
>  
> *l.*  
>  
>  
>  
>

-

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/>