

Re: Nobody should ever need to patch the kernel!!

Re: Nobody should ever need to patch the kernel!!

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.development.system/2006-10/msg00150.html>

- *From:* phil-news-nospam@xxxxxxxxx
 - *Date:* 13 Oct 2006 18:36:49 GMT
-

On 13 Oct 2006 01:30:18 -0700 440gtx@xxxxxxxxx wrote:

| phil-news-nospam@xxxxxxxxx wrote:

> All code that works with that struct must be compiled with that change.

> The code inside the kernel would all be compiled at once and always

> match. The code if a driver may not.

|

| Untrue! Modules are **required** to be built against the exact kernel

| they will run with. They will be just as aware and susceptible to the

| most minute changes as code being recompiled in the kernel.

I've run mismatched modules before. They even ran OK.

> This is one reason proprietary modules are not always a good thing.

> I personally avoid them unless I have source code and can recompile

> them.

|

| Designing a system where only gurus can install system level software

| limits the appeal of the operating system. Computers are increasingly

| used by more general types who don't want to know what a kernel is

| these days.

Well at least I can install system level software in Linux. You can't even do that at all in Windows.

> There are APIs for the commonly expected things. It would be impractical

> to have a hook for every possible thing because then you'd end up with a

> hook every 3 or 4 lines of code, and a kernel many times its size.

|

| Just offer simple methods to access existing things in architected ways

| each with its own versioning. Leave the performance path and internals

| alone. Just about everything is already there anyway like kobjects.

| Insulate modules, don't kill them on every new build.

Actually what we need are fewer modules.

> TSR was a horrible hack

Re: Nobody should ever need to patch the kernel!!

Re: Nobody should ever need to patch the kernel!!

|
| You must admit they could be written to work on every single version of
| dos and they added terrific features that everyone used. Sidekick, disk
| caches, key recalls, mouse, cdrom, >540MB hard drives, you name it, it
| could and was done. Dos would have been very boring without them, hmm.

I never used any of them.

> It doesn't work in modern Microsoft OSes.

|
| I beg to differ. Windows allows installing 3rd party drivers just as
| easily as installing applications and you can easily support all the
| important windows versions with one package. There is some very cool
| stuff out there that can't be ported to linux because of the kernel
| nightmare situation.

Only drivers from companies that pay through the nose to Microsoft for
the licensed development products and key signings.

> Why don't you describe to me a "creative product" that is being
> strangled because of Linux's model for driver interfacing?

|
| This I believe is the fundamental problem. I notice a lot of "if I
| can't think of a reason, there must not be any and therefore I will
| lock the door to the possibliity". This is fine if you are designing
| something for yourself, but there are unlimited numbers of creative
| people out there who have ideas you will never dream of and create new
| classes of products that need non traditional device drivers you never
| thought about.

You didn't want to describe a real example because whatever it is, I
could describe why it doesn't really need a driver in the first place.

So until you do come up with some examples, even hypothetical ones,
I'm just chalking up the above as "blowing more smoke".

> "Creative" should not be "neat tricks" to use the CPU to do the hardware's
> work. A classic example of a very UN-creative product is the winmodem.

|
| I don't know the hardware specifics of winmodem in particular, but the
| key is if you move intelligence out of the hardware, you need to have
| an extra intelligent person writing the driver to make it work right.
| For example, IDE and all its protocols and DMA modes is much more
| primitive to control than fancy SCSI hardware, but it works great if
| you write a good driver for it. But put some random joe on it and it
| will be full of problems--perhaps what happened to winmodem? But sure,
| some hardware can be fundamentally flawed (polling, static delays,
| etc). Fancy hardware is not always good because it tends to cost more.

Re: Nobody should ever need to patch the kernel!!

Re: Nobody should ever need to patch the kernel!!

| That's why IDE dominated SCSI and USB dominated firewire. Cost wins.

It's the wrong way to go about it. There are numerous different operating systems. But the manufacturer clearly didn't want to make a driver for all of them. The reason had nothing to do with kernel versions in Linux. It had everything to do with the fact that they were making a cheap product and didn't want to put forth any effort to make a driver for any other OS.

The correct design would be to NOT move the intelligence out to the CPU. It should be left in the hardware itself. The interface from the software running the CPU and the hardware should be basic and simple and common to all hardware of the same class (e.g. every modem should use the very same interface). Then if you want to make a hardware product that runs faster or processed bits better under noisy conditions or whatever, put all that in the hardware where it's safe. Then any OS can use it.

|-----/-----|
| Phil Howard KA9WGN (ka9wgn.ham.org) / Do not send to the address below |
| first name lower case at ipal.net / spamtrap-2006-10-13-1325@xxxxxxxx |
|-----/-----|

.