Subject: Re: [PATCH 0/59] Cleanup sysctl
Posted by ebiederm on Tue, 16 Jan 2007 18:12:44 GMT
View Forum Message <> Reply to Message

CC list trimmed.

"H. Peter Anvin" <hpa@zytor.com> writes:

> Eric W. Biederman wrote:
>>
>> - Removal of sys_sysctl support where people had used conflicting sysctl
>>   numbers. Trying to break glibc or other applications by changing the
>>   ABI is not cool. 9 instances of this in the kernel seems a little
>>   extreme.
>>
>
> It would be highly advantageous if we could have a file that acts as a central
> registry of architectural sysctl numbers *and have the numbers in the kernel
> derived from there*. As I've said before, I don't really think sys_sysctl is
> any worse than ad hoc system calls (sys_mips and the like), but the real problem
> is that there are architectural and non-archtectural numbers, and they're mixed
> in all over the place.

Conflicting with generic sys_sysctl numbers is a problem. Period.
All of the conflicts were with the binary version of:
/proc/sys/kernel or /proc/sys/dev/cdrom
Which are respectively: 1 and 7/1.

The conflicts were because people were simply not-trying. I didn't
look hard and scrutinize things I just stumbled on them while C99
converting the tables. So I could remove the stupid proc_dir_entry
field from ctl_table.

All you need for an architecture depending sys_sysctl is your own
top level architectural number. After that you get a unique path.
It's easy. If anyone had really cared these problems simply would
not have happened because in most cases the conflicting sysctl
were not exported to user space because they were masked, or
similarly architecture entities like /proc/sys/kernel/ostype and
/proc/sys/kernel/osrelease were masked.

Currently I'm happy with the current maintenance situation if someone
does not care to deal with the binary interface they don't have to.
And we can just mark them CTL_UNNUMBERED in the table. I would say
that is the real problem with the entries I fixed, they didn't want
to use the binary interface in the first place.

> I think it would be fair to say that if they're not in <linux/sysctl.h> they're

> not architectural, but that doesn't resolve the counterpositive (are there
> sysctls in <linux/sysctl.h> which aren't architectural? From the looks of it, I
> would say yes.) Non-architectural sysctl numbers should not be exported to
> userspace, and should eventually be rejected by sys_sysctl.

This last bit doesn't make much sense. I believe you are saying all sysctl numbers should be per architecture.

To your query about what the state of sys_sysctl is please go look there are only 79 instances of register_sysctl_table in the kernel. It's big but in an hour or two you can look at everything. Or at least read my patchset.

At this point there are no significant users of sysctl in the architecture code. The only big user of sysctl is the network stack and it uses sysctl responsibly.

The biggest blunders in using sysctl happen in the per architecture code and it is very much because the people writing the code didn't even try to get the binary interface right.

So I think making sysctl numbers per architecture is a hideous idea because that is not where the maintenance is happening, and instead it places a big burden on people who by the evidence don't care enough to get it right.

I think a much better idea is to maintain the current situation and just insist that people use CTL_UNNUMBERED for their binary number when they don't care about the binary interface. That is easy and it works.

Personally I expect the binary interface to largely remain fixed with exactly the set of non-architectural numbers we have today with any new sysctl users not allocating a sysctl number.

Eric

____