Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by ebiederm on Thu, 27 Sep 2007 19:25:48 GMT

View Forum Message <> Reply to Message

I still need to look at the code in detail but I have some concerns
I want to inject into this conversation of future sysfs architecture.

- If we want to carefully limit sysfs from going to wild code review
  is clearly not enough. We need some technological measures to
  assist us. As the experience with sysctl has shown.

  I discovered that something like 10% of the sysctl entries were
  buggy and had been for years when I added basic runtime sanity
  checks.

  I had also found one instance in the kernel and had one instance
  from outside the kernel where people had created files under
  /proc/sys not as sysctls but as using the infrastructure from
  proc_generic.c because it happened to work.

  So while it very well may be we don't want to use the kobject
  interface anymore. I expect that we want to have the sysfs_dirent
  interface not exported to modules, and only allow direct
  access from code compiled into the kernel.

  Mostly I am thinking that any non-object model users should have
  their own dedicated wrapper layer. To help keep things consistent
  and to make it hard enough to abuse the system that people will
  find that it is usually easier to do it the right way.

- The network namespace work scheduled to be merged in 2.6.24 is
  currently has a dependency in Kconfig that is "&& !SYSFS"
  because sysfs is currently very much a moving target.

  Does it look like we can resolve Tejun's work for 2.6.24?
  If not does it make sense to push my patches that allow
  multiple mounts of sysfs for 2.6.24? So I can allow
  network namespaces in the presence of sysfs.

  Outside of sysfs and the device model I'm only talk maybe 30 lines
  of code...   So I could easily merge that patch later in the
  merge window after the other pieces have gone in.

- Farther down the road we have the device namespace.
  The bounding requirements are:
  - We want to restrict which set of devices a subset of process
    can access.
  - When we migrate an application we want to preserve the device

numbers of all devices that show up in the new location.
So filesystems whose block devices reside on a SAN, ramdisks,
ttys, etc.
Other devices that really are different we can handle with
hotplug remove and add events, during the migration.

So while there is lower hanging fruit the requirements for a
device namespace are becoming clear, and don't look like something
we will ultimately be able to dodge.

For sysfs the implication is that we will need to filter the
hotplug events based upon the device namespace of the recipient, and
we will need to restrict the set of devices that show up in sysfs
based on who mounts it (as the prototype patches with the network
namespace are doing).

Also fun is that the dev file implementation needs to be able to
report different major:minor numbers based on which mount of
sysfs we are dealing with.

Eric