

Greg KH <[greg@kroah.com](mailto:greg@kroah.com)> writes:

> I would be interested in seeing what your patches look like.

Sure.

> I don't  
> think that we should take any more sysfs changes for 2.6.24 as we do  
> have a lot of them right now, and I don't think that Tejun and I agree  
> on the future direction of the outstanding ones just yet.

Sounds reasonable.

> But I don't think that your multiple-mount patches could make it into  
> .24, unless .23 is still weeks away.

Well I have posted them all earlier. At this point I it makes most sense to wait until after the big merge happen and every rebases on top of that. Then everyone will have network namespace support and it is easier to look through all of the patches. Especially since it looks like the merge window will open any day now.

I will quickly recap the essence of what I am looking at:  
On directories of interest I tag all of their directory entries with which namespace they belong to. On a mount of sysfs I remember which namespace we were in when we mounted sysfs. The I filter readdir and lookup based upon the namespace I captured at mount time. I do my best to generalize it so that the logic can work for different namespaces.

Currently the heart of the patch from the network namespace is below (I sniped the part that does the capture at mount time). Basically the interface to users of this functionality is just providing some way to go from a super block or a kobject to the tag sysfs is using to filter things.

So I get one sysfs\_dirent tree, but each super\_block has it's own tree of dcache entries.

Everything else is pretty much details in checking and propagating the tags into the appropriate places.

Eric

```

diff --git a/net/core/net-sysfs.c b/net/core/net-sysfs.c
index 5adfdc2..a300f6e 100644
--- a/net/core/net-sysfs.c
+++ b/net/core/net-sysfs.c
@@ -435,6 +437,23 @@ static void netdev_release(struct device *d)
    kfree((char *)dev - dev->padded);
}

+static const void *net_sb_tag(struct sysfs_tag_info *info)
+{
+ return info->net_ns;
+}
+
+static const void *net_kobject_tag(struct kobject *kobj)
+{
+ struct net_device *dev;
+ dev = container_of(kobj, struct net_device, dev.kobj);
+ return dev->nd_net;
+}
+
+static const struct sysfs_tagged_dir_operations net_tagged_dir_operations = {
+ .sb_tag = net_sb_tag,
+ .kobject_tag = net_kobject_tag,
+};
+
+static struct class net_class = {
+ .name = "net",
+ .dev_release = netdev_release,
@@ -444,6 +463,7 @@ static struct class net_class = {
#ifdef CONFIG_HOTPLUG
    .dev_uevent = netdev_uevent,
#endif
+ .tag_ops = &net_tagged_dir_operations,
+};

/* Delete sysfs entries but hold kobject reference until after all
--
1.5.3.rc6.17.g1911

```

---

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---