

---

Subject: Re: [PATCH 06/11] sysfs: Implement sysfs tagged directory support.  
Posted by [ebiederm](#) on Thu, 03 Jul 2008 05:11:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Tejun Heo <[htejun@gmail.com](mailto:htejun@gmail.com)> writes:

> There is rather large possibility that I'm just being dumb here  
> especially because I haven't reviewed the users of this facility, so all  
> the comments I'm making are from the POV of interfaces of sysfs and the  
> related layers. I think I've made my concerns clear by now. If you  
> still think the callbacks are the best way to go, please try to  
> enlighten me. I really don't wanna be stopping something which is  
> better from ignorance. Just give me some concrete examples or point me  
> to codes which show how and why the current interface is the best for  
> the users and switching isn't a good idea.

Currently I think a callback on to get the tag from a kobject is the best way to go. That way we don't need to add a field to struct kobject (and don't need the associated redundancy), and we can lookup up the tag when we need it.

I have been playing with the code and just about have it ready to go. I just need to refactor all of my changes into clean patches at this point, plus a bit of review and test. Ben & Daniel have given me a version of the previous patchset rebased unto the latest -mm so that should help for the unchanged parts.

Introducing the sysfs\_tag\_type thing and pushing the functions to the edges helps. It especially cleans up the ugly mount/umount situation allowing us to handle that with generic code.

Moving the kobject\_tag into struct ktype works and looks roughly as clean as what happens with attributes. So I that seems reasonable, and doesn't result in a significant change in the users.

The result of which means that I only have the helper function sysfs\_creation\_tag left in sysfs/dir.c Left in there are some of the nasties in dealing with symlinks.

At this point I believe I have achieved a nice degree of simplifying the sysfs code in the current patches without really changing the users or making it more complex for them.

I have not implemented ida tags, and I don't plan to. That is just unnecessary work right now. The users are simple and the meat of the logic would not change so it should be simple to add.

>> It looks to me like the clean solution is move kobject\_tag into  
>> kobj\_type, and have it call some higher level function.

>>  
>> We also need to remove the maintenance disaster that is  
>> kobject\_set\_name from sysfs\_rename\_dir. And push it into  
>> kobject\_rename instead. The error handling is harder in  
>> that case but otherwise we should be in good shape.  
>  
> Heh... I personally think kobject layer as a whole should just be hidden  
> under the cabinet of device driver model but I'm having difficult time  
> convincing other people of it. Anyways, fully agree the interaction  
> between kobject and sysfs is ugly at a lot of places.

I would be happy if we could remove all nonsense kobject that are there just for structural purposes but have no purpose otherwise. Things like kobjects for symlinks. The kobject layer doesn't seem to have a clear identity and purpose that I can see right now.

> Thanks a lot for your patience.

Welcome. The code reached a point a while ago where it didn't make sense to change it without review feedback.

Eric

---

Containers mailing list  
Containers@lists.linux-foundation.org  
<https://lists.linux-foundation.org/mailman/listinfo/containers>

---