

[RFC] kobject: convert some users of kobject_init to the new functions.

[RFC] kobject: convert some users of kobject_init to the new functions.

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-11/msg10295.html>

- *From:* Greg KH <greg@xxxxxxxx>
 - *Date:* Fri, 30 Nov 2007 11:54:45 -0800
-

This converts the kernel/ directory and the char_dev directory to use the new kobject_init functions as an example and test of them.

Cc: Kay Sievers <kay.sievers@xxxxxxxx>
Signed-off-by: Greg Kroah-Hartman <gregkh@xxxxxxxx>

fs/char_dev.c | 18 ++++++-----
kernel/module.c | 12 ++++-----
kernel/params.c | 5 +-----
kernel/user.c | 5 +-----
mm/slub.c | 5 +-----
5 files changed, 21 insertions(+), 24 deletions(-)

```
--- a/fs/char_dev.c
+++ b/fs/char_dev.c
@@ -509,10 +509,15 @@ static struct kobj_type ktype_cdev_dynam
struct cdev *cdev_alloc(void)
{
struct cdev *p = kzalloc(sizeof(struct cdev), GFP_KERNEL);
+ int ret;
+
+ if (p) {
- p->kobj.ktype = &ktype_cdev_dynamic;
INIT_LIST_HEAD(&p->list);
- kobject_init(&p->kobj);
+ ret = kobject_init_ng(&p->kobj, &ktype_cdev_dynamic, NULL,
+ "cdev_dynamic");
+ if (ret)
+ printk(KERN_ERR "cdev: something went wrong "
+ "initializing the dynamic kobject!\n");
}
return p;
}
@@ -527,10 +532,15 @@ struct cdev *cdev_alloc(void)
*/
void cdev_init(struct cdev *cdev, const struct file_operations *fops)
{
```

[RFC] kobject: convert some users of kobject_init to the new functions.

[RFC] kobject: convert some users of kobject_init to the new functions.

```
+ int ret;
+
memset(cdev, 0, sizeof *cdev);
INIT_LIST_HEAD(&cdev->list);
- cdev->kobj.ktype = &ktype_cdev_default;
- kobject_init(&cdev->kobj);
+ ret = kobject_init_ng(&cdev->kobj, &ktype_cdev_default, NULL,
+ "cdev_static");
+ if (ret)
+ printk(KERN_ERR "cdev: something went very wrong initializing "
+ "the static kobject!\n");
cdev->ops = fops;
}

--- a/kernel/module.c
+++ b/kernel/module.c
@@ -1218,18 +1218,14 @@ int mod_sysfs_init(struct module *mod)
err = -EINVAL;
goto out;
}
- memset(&mod->mkobj.kobj, 0, sizeof(mod->mkobj.kobj));
- err = kobject_set_name(&mod->mkobj.kobj, "%s", mod->name);
- if (err)
- goto out;
- mod->mkobj.kobj.kset = module_kset;
- mod->mkobj.kobj.ktype = &module_ktype;
mod->mkobj.mod = mod;

- kobject_init(&mod->mkobj.kobj);
+ memset(&mod->mkobj.kobj, 0, sizeof(mod->mkobj.kobj));
+ mod->mkobj.kobj.kset = module_kset;
+ err = kobject_init_and_add(&mod->mkobj.kobj, &module_ktype, NULL,
+ "%s", mod->name);

/* delay uevent until full sysfs population */
- err = kobject_add(&mod->mkobj.kobj);
out:
return err;
}

--- a/kernel/params.c
+++ b/kernel/params.c
@@ -561,10 +561,7 @@ static void __init kernel_param_sysfs_se

mk->mod = THIS_MODULE;
mk->kobj.kset = module_kset;
- mk->kobj.ktype = &module_ktype;
- kobject_set_name(&mk->kobj, name);
- kobject_init(&mk->kobj);
- ret = kobject_add(&mk->kobj);
+ ret = kobject_init_and_add(&mk->kobj, &module_ktype, NULL, "%s", name);
if (ret) {
```

[RFC] kobject: convert some users of kobject_init to the new functions.

[RFC] kobject: convert some users of kobject_init to the new functions.

```
printk(KERN_ERR "Module '%s' failed to be added to sysfs, "
"error number %d\n", name, ret);
--- a/kernel/user.c
+++ b/kernel/user.c
@@ -181,11 +181,8 @@ static int uids_user_create(struct user_
int error;

memset(kobj, 0, sizeof(struct kobject));
- kobj->ktype = &uids_ktype;
kobj->kset = uids_kset;
- kobject_init(kobj);
- kobject_set_name(&up->kobj, "%d", up->uid);
- error = kobject_add(kobj);
+ error = kobject_init_and_add(kobj, &uids_ktype, NULL, "%d", up->uid);
if (error)
goto done;

--- a/mm/slub.c
+++ b/mm/slub.c
@@ -4021,11 +4021,8 @@ static int sysfs_slab_add(struct kmem_ca
name = create_unique_id(s);
}

- kobject_set_name(&s->kobj, name);
s->kobj.kset = slab_kset;
- s->kobj.ktype = &slab_ktype;
- kobject_init(&s->kobj);
- err = kobject_add(&s->kobj);
+ err = kobject_init_and_add(&s->kobj, &slab_ktype, NULL, name);
if (err)
return err;

-
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx
More majordomo info at http://vger.kernel.org/majordomo-info.html
Please read the FAQ at http://www.tux.org/lkml/
```

[RFC] kobject: convert some users of kobject_init to the new functions.