
Subject: [RFC] Remove kern_mount() in init_devpts_fs()
Posted by [Sukadev Bhattiprolu](#) on Tue, 26 Feb 2008 00:46:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Is the kern_mount() of devpts really needed or can we simply register the filesystem type and wait for an user-space mount before being able to create PTYs ?

This is just an RFC patch that removes the kern_mount() and the 'devpts_mnt' and 'devpts_root' global variables and uses a 'devpts_sb' to store the single super block associated with devpts.

Removing the kern_mount() and relying on user-space mount could simplify cloning of PTS namespaces.

fs/devpts/inode.c | 49 ++++++-----
1 file changed, 29 insertions(+), 20 deletions(-)

Index: linux-2.6.24/fs/devpts/inode.c

=====
--- linux-2.6.24.orig/fs/devpts/inode.c 2008-02-22 14:23:53.000000000 -0800
+++ linux-2.6.24/fs/devpts/inode.c 2008-02-25 16:00:17.000000000 -0800
@@ -23,9 +23,6 @@

```
#define DEVPTS_SUPER_MAGIC 0x1cd1

-static struct vfsmount *devpts_mnt;
-static struct dentry *devpts_root;
-
static struct {
    int setuid;
    int setgid;
@@ -97,6 +94,7 @@ static const struct super_operations dev
    .remount_fs = devpts_remount,
};

+static struct super_block *devpts_sb;
static int
devpts_fill_super(struct super_block *s, void *data, int silent)
{
@@ -120,9 +118,11 @@ devpts_fill_super(struct super_block *s,
    inode->i_fop = &simple_dir_operations;
    inode->i_nlink = 2;

- devpts_root = s->s_root = d_alloc_root(inode);
- if (s->s_root)
```

```

+ s->s_root = d_alloc_root(inode);
+ if (s->s_root) {
+   devpts_sb = s;
+   return 0;
+ }

    printk("devpts: get root dentry failed\n");
    iput(inode);
@@ -136,11 +136,17 @@ static int devpts_get_sb(struct file_sys
    return get_sb_single(fs_type, flags, data, devpts_fill_super, mnt);
}

+static void devpts_kill_sb(struct super_block *sb)
+{
+ devpts_sb = NULL;
+ kill_anon_super(sb);
+}
+
static struct file_system_type devpts_fs_type = {
    .owner = THIS_MODULE,
    .name = "devpts",
    .get_sb = devpts_get_sb,
- .kill_sb = kill_anon_super,
+ .kill_sb = devpts_kill_sb,
};

/*
@@ -151,7 +157,12 @@ static struct file_system_type devpts_fs
static struct dentry *get_node(int num)
{
    char s[12];
- struct dentry *root = devpts_root;
+ struct dentry *root;
+
+ if (!devpts_sb)
+   return NULL;
+
+ root = devpts_sb->s_root;
    mutex_lock(&root->d_inode->i_mutex);
    return lookup_one_len(s, root, sprintf(s, "%d", num));
}
@@ -162,7 +173,12 @@ int devpts_pty_new(struct tty_struct *tt
    struct tty_driver *driver = tty->driver;
    dev_t device = MKDEV(driver->major, driver->minor_start+number);
    struct dentry *dentry;
- struct inode *inode = new_inode(devpts_mnt->mnt_sb);
+ struct inode *inode;
+

```

```

+ if (!devpts_sb)
+ return -ENOSYS;
+
+ inode = new_inode(devpts_sb);

/* We're supposed to be given the slave end of a pty */
BUG_ON(driver->type != TTY_DRIVER_TYPE_PTY);
@@ -181,10 +197,10 @@ int devpts_pty_new(struct tty_struct *tt
    dentry = get_node(number);
    if (!IS_ERR(dentry) && !dentry->d_inode) {
        d_instantiate(dentry, inode);
- fsnotify_create(devpts_root->d_inode, dentry);
+ fsnotify_create(devpts_sb->s_root->d_inode, dentry);
    }

- mutex_unlock(&devpts_root->d_inode->i_mutex);
+ mutex_unlock(&devpts_sb->s_root->d_inode->i_mutex);

    return 0;
}
@@ -201,7 +217,7 @@ struct tty_struct *devpts_get_tty(int nu
    dput(dentry);
}

- mutex_unlock(&devpts_root->d_inode->i_mutex);
+ mutex_unlock(&devpts_sb->s_root->d_inode->i_mutex);

    return tty;
}
@@ -219,24 +235,17 @@ void devpts_pty_kill(int number)
}
    dput(dentry);
}
- mutex_unlock(&devpts_root->d_inode->i_mutex);
+ mutex_unlock(&devpts_sb->s_root->d_inode->i_mutex);
}

static int __init init_devpts_fs(void)
{
- int err = register_filesystem(&devpts_fs_type);
- if (!err) {
-     devpts_mnt = kern_mount(&devpts_fs_type);
-     if (IS_ERR(devpts_mnt))
-         err = PTR_ERR(devpts_mnt);
- }
- return err;
+ return register_filesystem(&devpts_fs_type);
}

```

```
static void __exit exit_devpts_fs(void)
{
    unregister_filesystem(&devpts_fs_type);
- mntput(devpts_mnt);
}

module_init(init_devpts_fs)
```

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