

---

Subject: Re: [patch 1/8] add user mounts to the kernel  
Posted by [Miklos Szeredi](#) on Sun, 22 Apr 2007 16:22:40 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

```
> >> > +
> >> + uid_t mnt_uid; /* owner of the mount */
> >
> >> Can we please make this a user struct. That requires a bit of
> >> reference counting but it has uid namespace benefits as well
> >> as making it easy to implement per user mount rlimits.
> >
> > OK, can you elaborate, what the uid namespace benefits are?
>
> In the uid namespace the comparison is simpler as are the propagations
> rules. Basically if you use a struct user you will never need to
> care about a uid namespace.
```

I tried to implement it but got stuck on this: fsuid doesn't have a user\_struct in task\_struct (yet), so we'd now have to convert current->fsuid to a user\_struct. This can be done with alloc\_uid(), but this can fail, bringing in extra error handling complexity.

Also we'd have to compare current->fsuid with a user\_struct, which we don't yet know how will actually be done in the future.

So it seems, we still have to care about the uid namespace, at least if fsuid is preferred to ruid.

Anyway, here's a patch fixing the other things you brought up, and which I agree with. Does this look OK?

Thanks,  
Miklos

Index: linux/fs/namespace.c

---

```
--- linux.orig/fs/namespace.c 2007-04-22 17:48:18.000000000 +0200
+++ linux/fs/namespace.c 2007-04-22 18:19:51.000000000 +0200
@@ -252,10 +252,12 @@ static int reserve_user_mount(void)
 static void __set_mnt_user(struct vfsmount *mnt)
 {
     BUG_ON(mnt->mnt_flags & MNT_USER);
-    mnt->mnt_uid = current->uid;
+    mnt->mnt_uid = current->fsuid;
     mnt->mnt_flags |= MNT_USER;
-    if (!capable(CAP_SYS_ADMIN))
-        mnt->mnt_flags |= MNT_NOSUID | MNT_NODEV;
```

```

+ if (!capable(CAP_SETUID))
+ mnt->mnt_flags |= MNT_NOSUID;
+ if (!capable(CAP_MKNOD))
+ mnt->mnt_flags |= MNT_NODEV;
}

static void set_mnt_user(struct vfsmount *mnt)
@@ -725,10 +727,10 @@ static bool permit_umount(struct vfsmoun
if (!(mnt->mnt_flags & MNT_USER))
return false;

- if (flags & MNT_FORCE)
+ if ((flags & MNT_FORCE) && !(mnt->mnt_sb->s_type->fs_flags & FS_SAFE))
return false;

- return mnt->mnt_uid == current->uid;
+ return mnt->mnt_uid == current->fsuid;
}

/*
@@ -792,13 +794,13 @@ static bool permit_mount(struct nameidata
if (type && !(type->fs_flags & FS_SAFE))
return false;

- if (!S_ISDIR(inode->i_mode) && !S_ISREG(inode->i_mode))
+ if (S_ISLNK(inode->i_mode))
return false;

if (!(nd->mnt->mnt_flags & MNT_USER))
return false;

- if (nd->mnt->mnt_uid != current->uid)
+ if (nd->mnt->mnt_uid != current->fsuid)
return false;

*flags |= MS_SETUSER;

```

---

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

---