
Subject: [patch -mm 4/5] mqueue namespace : enable the namespace

Posted by [Cedric Le Goater](#) on Tue, 02 Oct 2007 08:46:12 GMT

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

From: Cedric Le Goater <clg@fr.ibm.com>

Move forward and start using the mqueue namespace.

The single super block mount of the file system is modified to allow one mount per namespace. This is achieved by storing the namespace in the super_block s_fs_info attribute.

Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

ipc/mqueue.c | 26 ++++++-----
1 file changed, 19 insertions(+), 7 deletions(-)

Index: 2.6.23-rc8-mm2/ipc/mqueue.c

=====

--- 2.6.23-rc8-mm2.orig/ipc/mqueue.c

+++ 2.6.23-rc8-mm2/ipc/mqueue.c

@@ -108,6 +108,8 @@ static inline struct mqueue_inode_info *
return container_of(inode, struct mqueue_inode_info, vfs_inode);
}

+static struct file_system_type mqueue_fs_type;

+

static struct mq_namespace *clone_mq_ns(struct mq_namespace *old_ns)
{

struct mq_namespace *mq_ns;

@@ -121,7 +123,12 @@ static struct mq_namespace *clone_mq_ns(
mq_ns->queues_max = DFLT_QUEUESMAX;

mq_ns->msg_max = DFLT_MSGMAX;

mq_ns->msgsize_max = DFLT_MSGSIZEMAX;

- mq_ns->mnt = NULL;

+ mq_ns->mnt = kern_mount_data(&mqueue_fs_type, mq_ns);

+ if (IS_ERR(mq_ns->mnt)) {
+ void *error = mq_ns->mnt;

+ kfree(mq_ns);

+ return error;

+ }

return mq_ns;
}

@@ -147,6 +154,7 @@ void free_mq_ns(struct kref *kref)

struct mq_namespace *mq_ns;

mq_ns = container_of(kref, struct mq_namespace, kref);

```

+ mntput(mq_ns->mnt);
  kfree(mq_ns);
}

@@ -253,7 +261,11 @@ static int mqueue_get_sb(struct file_sys
    int flags, const char *dev_name,
    void *data, struct vfsmount *mnt)
{
- return get_sb_single(fs_type, flags, data, mqueue_fill_super, mnt);
+ if (!(flags & MS_KERNMOUNT))
+   data = current->nsproxy->mq_ns;
+
+ return get_sb_single_per_data(fs_type, flags, data, mqueue_fill_super,
+   mnt);
}

static void init_once(struct kmem_cache *cachep, void *foo)
@@ -284,7 +296,7 @@ static void mqueue_delete_inode(struct i
    struct user_struct *user;
    unsigned long mq_bytes;
    int i;
- struct mq_namespace *mq_ns = &init_mq_ns;
+ struct mq_namespace *mq_ns = inode->i_sb->s_fs_info;

    if (S_ISDIR(inode->i_mode)) {
        clear_inode(inode);
    }
@@ -317,7 +329,7 @@ static int mqueue_create(struct inode *d
    struct inode *inode;
    struct mq_attr *attr = dentry->d_fsdata;
    int error;
- struct mq_namespace *mq_ns = &init_mq_ns;
+ struct mq_namespace *mq_ns = dir->i_sb->s_fs_info;

    spin_lock(&mq_lock);
    if (mq_ns->queues_count >= mq_ns->queues_max &&
@@ -709,7 +721,7 @@ asmlinkage long sys_mq_open(const char _
    struct file *filp;
    char *name;
    int fd, error;
- struct mq_namespace *mq_ns = &init_mq_ns;
+ struct mq_namespace *mq_ns = current->nsproxy->mq_ns;

    error = audit_mq_open(oflag, mode, u_attr);
    if (error != 0)
@@ -781,7 +793,7 @@ asmlinkage long sys_mq_unlink(const char
    char *name;
    struct dentry *dentry;
    struct inode *inode = NULL;

```

```
- struct mq_namespace *mq_ns = &init_mq_ns;
+ struct mq_namespace *mq_ns = current->nsproxy->mq_ns;

    name = getname(u_name);
    if (IS_ERR(name))
@@ -1319,7 +1331,7 @@ static int __init init_mqueue_fs(void)
    if (error)
        goto out_sysctl;

- init_mq_ns.mnt = kern_mount(&mqueue_fs_type);
+ init_mq_ns.mnt = kern_mount_data(&mqueue_fs_type, &init_mq_ns);
    if (IS_ERR(init_mq_ns.mnt)) {
        error = PTR_ERR(init_mq_ns.mnt);
        goto out_filesystem;

--
```

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