Subject: [patch 04/10] allow per-mount flags to be set/cleared individually
Posted by Miklos Szeredi on Mon, 16 Apr 2007 11:03:12 GMT
View Forum Message <> Reply to Message

With MS_REMOUNT, it is difficult to change mount flags individually,
without touching other mount flags or options, having to parse
/proc/mounts to get the old flag values and options.

It is also difficult to change a mount flag recursively, having to
walk the mount tree in userspace.

This patch solves this problem by generalizing do_change_type() so
that not only the propagation property can be changed, but mnt_flags
can be set/cleared individually.

From: Miklos Szeredi <mszeredi@suse.cz>

Signed-off-by: Miklos Szeredi <mszeredi@suse.cz>

Index: linux/fs/namespace.c
===
--- linux.orig/fs/namespace.c 2007-04-13 13:04:04.000000000 +0200
+++ linux/fs/namespace.c 2007-04-13 13:04:29.000000000 +0200
@@ -955,19 +956,28 @@ out_unlock:
 /*
  * recursively change the type of the mountpoint.
  */
-static int do_change_type(struct nameidata *nd, int flag)
+static int do_change_mnt(struct nameidata *nd, int flag, int mnt_flags)
 {
  struct vfsmount *m, *mnt = nd->mnt;
  int recurse = flag & MS_REC;
- int type = flag & ~MS_REC;
+ int type = flag & (MS_SHARED | MS_PRIVATE | MS_SLAVE | MS_UNBINDABLE);

  if (nd->dentry != nd->mnt->mnt_root)
   return -EINVAL;

  down_write(&namespace_sem);
  spin_lock(&vfsmount_lock);
- for (m = mnt; m; m = (recurse ? next_mnt(m, mnt) : NULL))
- change_mnt_propagation(m, type);
+ for (m = mnt; m; m = (recurse ? next_mnt(m, mnt) : NULL)) {
+ if (type)
+   change_mnt_propagation(m, type);
+
+ if (flag & (MS_SETFLAGS | MS_CLEARFLAGS))

```
+   m->mnt_flags = mnt_flags;
+  else if (flag & MS_SETFLAGS)
+   m->mnt_flags |= mnt_flags;
+  else if (flag & MS_CLEARFLAGS)
+   m->mnt_flags &= ~mnt_flags;
+ }
  spin_unlock(&vfsmount_lock);
  up_write(&namespace_sem);
  return 0;
@@ -1514,8 +1526,9 @@ long do_mount(char *dev_name, char *dir_
         data_page);
  else if (flags & MS_BIND)
   retval = do_loopback(&nd, dev_name, flags);
- else if (flags & (MS_SHARED | MS_PRIVATE | MS_SLAVE | MS_UNBINDABLE))
-  retval = do_change_type(&nd, flags);
+ else if (flags & (MS_SHARED | MS_PRIVATE | MS_SLAVE | MS_UNBINDABLE |
+     MS_SETFLAGS | MS_CLEARFLAGS))
+  retval = do_change_mnt(&nd, flags, mnt_flags);
  else if (flags & MS_MOVE)
   retval = do_move_mount(&nd, dev_name);
  else
Index: linux/include/linux/fs.h
===================================================================
--- linux.orig/include/linux/fs.h 2007-04-13 13:04:04.000000000 +0200
+++ linux/include/linux/fs.h 2007-04-13 13:04:29.000000000 +0200
@@ -127,6 +127,9 @@ extern int dir_notify_enable;
 #define MS_SHARED (1<<20) /* change to shared */
 #define MS_RELATIME (1<<21) /* Update atime relative to mtime/ctime. */
 #define MS_SETUSER (1<<22) /* set mnt_uid to current user */
+#define MS_SETFLAGS (1<<23) /* set specified mount flags */
+#define MS_CLEARFLAGS (1<<24) /* clear specified mount flags */
+/* MS_SETFLAGS | MS_CLEARFLAGS: change mount flags to specified */
 #define MS_ACTIVE (1<<30)
 #define MS_NOUSER (1<<31)


--
```

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