

[RFC] mount can figure out fstype automatically

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2006-07/msg00021.html>

- *From:* Craig Rodrigues <rodrige@xxxxxxxxxxxxxxxx>
 - *Date:* Sat, 8 Jul 2006 11:28:01 -0400
-

Hi,

One of the pet peeves I have with FreeBSD is that if I have a device with a local filesystem that I want to mount, I need to explicitly know what type of filesystem is on the device in order to mount it from the command-line.

For example,

```
mount -t cd9660
mount -t udf
mount -t ext2fs
mount -t msdosfs
```

Where this is particularly annoying is if I have multiple USB thumb drives with different filesystems on them.

What I usually end up doing is something like:
file - < /dev/ad0s4

to figure out the filesystem type, and then mount -t [whatever] to mount it.

What I would like to do is:

```
mount /dev/ad0s4 /mnt
```

and if I do not specify a filesystem type with -t, the mount program should "magically" figure out how to mount the disk. This is closer to how the mount program behaves on Linux for example.

I've come up with a patch that does this, by interpreting an fstype of "" as:
- starting with "ufs", iterate over all the local filesystem types that we know about, and try to mount the device

Comments?

[RFC] mount can figure out fstype automatically

Index: sys/kern/vfs_mount.c

```
=====
RCS file: /home/ncvs/src/sys/kern/vfs_mount.c,v
retrieving revision 1.228
diff -u -u -r1.228 vfs_mount.c
--- sys/kern/vfs_mount.c 27 Jun 2006 14:46:31 -0000 1.228
+++ sys/kern/vfs_mount.c 8 Jul 2006 14:01:52 -0000
@@ -567,6 +567,34 @@
}

static int
+vfs_domount_try(struct thread *td, char *fspath, int fsflags,
+ void *fsdata)
+{
+ struct vfsconf *vfsp;
+ int error;
+ printf("Mounting: %s first\n", "ufs");
+ error = vfs_domount(td, "ufs", fspath, fsflags, fsdata);
+ if (error == 0) {
+ printf("successfully mounted: %s\n", "ufs");
+ }
+ if (error != 0) {
+ TAILQ_FOREACH(vfsp, &vfscnf, vfc_list) {
+ if ((strcmp("ufs", vfsp->vfc_name) != 0) &&
+ !(vfsp->vfc_flags & VFCF_NETWORK) &&
+ !(vfsp->vfc_flags & VFCF_SYNTHETIC)) {
+ printf("Mounting: %s\n", vfsp->vfc_name);
+ error = vfs_domount(td, vfsp->vfc_name, fspath, fsflags, fsdata);
+ if (error == 0) {
+ printf("successfully mounted: %s\n", vfsp->vfc_name);
+ break;
+ }
+ }
+ }
+ }
+ return error;
+}
+
+static int
vfs_donmount(struct thread *td, int fsflags, struct uio *fsoptions)
{
struct vfsoplist *optlist;
@@ -596,7 +624,7 @@
*/
fstypelen = 0;
error = vfs_getopt(optlist, "fstype", (void *)&fstype, &fstypelen);
- if (error || fstype[fstypelen - 1] != '\0') {
+ if (error || (fstypelen > 0 && fstype[fstypelen - 1] != '\0')) {
error = EINVAL;
if (errmsg != NULL)
strncpy(errmsg, "Invalid fstype", errmsg_len);

```

[RFC] mount can figure out fstype automatically

[RFC] mount can figure out fstype automatically

```
@@ -606,6 +634,7 @@
error = vfs_getopt(optlist, "fspath", (void *)&fspath, &fspathlen);
if (error || fspath[fspathlen - 1] != '\0') {
error = EINVAL;
+ printf("%s:%d EINVAL\n", __FILE__, __LINE__);
if (errmsg != NULL)
strncpy(errmsg, "Invalid fspath", errmsg_len);
goto bail;
@@ -686,7 +715,14 @@
}
```

```
mtx_lock(&Giant);
- error = vfs_domount(td, fstype, fspath, fsflags, optlist);
+ if (fstypelen > 1) {
+ /* fstype was specified, go directly to vfs_domount() */
+ error = vfs_domount(td, fstype, fspath, fsflags, optlist);
+ }
+ else {
+ /* we do not know the fstype, try to probe for it */
+ error = vfs_domount_try(td, fspath, fsflags, optlist);
+ }
mtx_unlock(&Giant);
bail:
/* copyout the errmsg */
Index: sbin/mount/mount.c
```

```
RCS file: /home/ncvs/src/sbin/mount/mount.c,v
retrieving revision 1.87
diff -u -u -r1.87 mount.c
--- sbin/mount/mount.c 10 Jun 2006 01:44:57 -0000 1.87
+++ sbin/mount/mount.c 8 Jul 2006 14:14:47 -0000
@@ -200,7 +200,7 @@
```

```
all = init_flags = 0;
vfslist = NULL;
- vfstype = "ufs";
+ vfstype = "";
while ((ch = getopt(argc, argv, "adF:fo:prwt:uv")) != -1)
switch (ch) {
case 'a':
Index: sbin/mount/mount_fs.c
```

```
RCS file: /home/ncvs/src/sbin/mount/mount_fs.c,v
retrieving revision 1.2
diff -u -u -r1.2 mount_fs.c
--- sbin/mount/mount_fs.c 13 Nov 2005 01:27:57 -0000 1.2
+++ sbin/mount/mount_fs.c 8 Jul 2006 14:14:47 -0000
@@ -50,6 +50,7 @@
```

```
#include <sys/param.h>
#include <sys/mount.h>
```

[RFC] mount can figure out fstype automatically

[RFC] mount can figure out fstype automatically

```
+#include <sys/uio.h>

#include <err.h>
#include <getopt.h>
@@ -82,8 +83,9 @@
int iovlen;
int mntflags = 0;
int ch;
- char *dev, *dir, mntpath[MAXPATHLEN];
+ char *dev, *dir, mntpath[MAXPATHLEN], frompath[MAXPATHLEN];
char fstype[32];
+ //char errmsg[1024] = { 0 };
char *p, *val;
int ret;

@@ -121,15 +123,35 @@
dir = argv[1];

(void)checkpath(dir, mntpath);
+ if (realpath(dev, frompath) != NULL) {
+ dev = frompath;
+ }
(void)rmslashes(dev, dev);

build_iovec(&iov, &iovlen, "fstype", fstype, (size_t)-1);
build_iovec(&iov, &iovlen, "fspath", mntpath, (size_t)-1);
build_iovec(&iov, &iovlen, "from", dev, (size_t)-1);
-
+ //build_iovec(&iov, &iovlen, "errmsg", errmsg, sizeof(errmsg));
+ printf("fstype: %s, fspath: %s, from: %s\n", fstype, mntpath, dev);
+retry:
ret = nmount(iov, iovlen, mntflags);
- if (ret < 0)
- err(1, "%s", dev);
+ if (ret < 0 && iov[1].iov_len == 0) {
+ /*
+ * If an fstype was not specified, and nmount() failed,
+ * try again with an fstype of "ufs". This is for backwards
+ * compatibility with older kernels which do not support
+ * do_mount_try() with an fstype of "".
+ */
+ iov[1].iov_base = strdup("ufs");
+ iov[1].iov_len = 4;
+ printf("Trying again....\n");
+ goto retry;
+ }
+
+ if (ret < 0) {
+ //err(1, "%s %s", dev, errmsg);
+ //err(1, "%s", dev);
+ }
```

[RFC] mount can figure out fstype automatically

```
return (ret);  
}
```

—
Craig Rodrigues
rodrigc@xxxxxxxxxxxxxx

freebsd-arch@xxxxxxxxxxxxx mailing list
<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>
To unsubscribe, send any mail to "freebsd-arch-unsubscribe@xxxxxxxxxxxxx"