

## Re: [RFC] mount(8) can figure out fstype

---

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2007-01/msg00080.html>

---

- *From:* Craig Rodrigues <[rodrigc@xxxxxxxxxxxxxxxxx](mailto:rodrigc@xxxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 18 Jan 2007 21:37:09 -0500
- 

On Thu, Jan 18, 2007 at 10:17:49PM +0200, Rostislav Krasny wrote:

OpenBSD already has such a functionality. It uses `readlabelfs(3)` for this. What disadvantages or advantages does it have beside your implementation?

Thanks for the pointer, I did not know about `readlabelfs` on OpenBSD! From what I understand from reading the code:  
<http://www.openbsd.org/cgi-bin/cvsweb/src/lib/libutil/readlabel.c>  
<http://www.openbsd.org/cgi-bin/cvsweb/src/sys/sys/disklabel.h>

the `readlabelfs()` function tries to open the device and do a `DIOCGDINFO ioctl()` on the device to read the disklabel.

Once the disklabel is read, (i.e. `DIOCGDINFO` returns a 'struct disklabel'), the `p_fstype` member of 'struct partition' which is inside the 'struct disklabel' is converted to a string name based on the `fstypesnames` array in <`sys/disklabel.h`>.

This approach relies heavily on disklabels, and correct information being stored in disklabels.

For FreeBSD, I do not like the idea of introducing a new dependency between `mount(8)` and BSD disklabels.

The existing FreeBSD `mount(8)` program just tries to open a device, does an `nmount()` with a "fstype" parameter, and then relies on the underlying file system code (i.e. "ufs", "msdosfs", etc.) to read the superblock from the device, figure out if it can mount it, and return an error if it cannot.

That's the behavior I tried to preserve with my patch. It's kind of simplistic, but it does work.

—  
Craig Rodrigues  
[rodrigc@xxxxxxxxxxxxxxxxx](mailto:rodrigc@xxxxxxxxxxxxxxxxx)

Re: [RFC] mount(8) can figure out fstype

---

freebsd-arch@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@xxxxxxxxxxx"