

DragonFlyBSD - Bug #1537

null mount does not accept -o update

09/27/2009 06:45 PM - corecode

Status: Closed	Start date:
Priority: Low	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	
Description In contrast to all other file systems, null mounts can not be updated, using mount -u or -o update. That means that null mounts (of PFS usually) which were initially mounted ro can not be upgraded to rw, but have to be unmounted first.	

History

#1 - 10/19/2009 05:08 PM - tuxillo

Simon,

I've checked FBSD and NBSD. The former can only do update mounts for NFS exports it seems, and the latter can't at all.

Any hit for doing this?

Cheers,
Antonio Huete

#2 - 10/19/2009 05:24 PM - dillon

:Antonio Huete Jimenez <tuxillo@quantumachine.net> added the comment:

:

:Simon,

:

:I've checked FBSD and NBSD. The former can only do update mounts for NFS ex=
:ports

:it seems, and the latter can't at all.

:

:Any hit for doing this?

:

:Cheers,

:Antonio Huete

:

:-----

:status: unread -> chatting

```
hit -> hint I assume. Yes, you should be able to implement the
MNT_UPDATE stuff for null mounts. MNT_UPDATE implementations only
allow certain flags to be changed out from under the mounted filesystem.
For null mounts basically the only thing you can update is read-only
vs read-write operation.
```

```
You can see how other VFSs handle MNT_UPDATE. It is fairly
straightforward.
```

-Matt

```
Matthew Dillon
<mailto:dillon@backplane.com>
```

#3 - 11/11/2009 09:28 PM - Anonymous

Salute.

I have a patch here:

<http://leaf.dragonflybsd.org/~beket/updnull.diff>

I have tested it extensively and it works. Stuff I tried:

1. Switching between ro/rw in a local null mount.
And verified read-onlyness by trying to touch a file.
2. Switching between ro/rw in stacks of local null mounts.
And verified read-onlyness by trying to touch a file.
3. Switching between ro/rw in a NFS exported null mount.
Specifically, /pfs/@@-1:00004 was null mounted on /home. And /home was NFS shared with dfly being the server. NFS client was opensolaris build 126. I was able to write from osol when /home was exported as read-write, and failed to do so when it was exported as read-only. Client correctly reported that fs was read-only. And so did mount(8) in the server side.

I can provide copy/pastes from sessions, upon request. But better, fetch the patch and try it yourself :)

This <http://leaf.dragonflybsd.org/~beket/mountnull.png> shows some preliminary tests I did (they do not cover switching back to ro from rw, or 3.)

Best regards,
Stathis Kamperis

#4 - 11/11/2009 09:46 PM - Anonymous

Backed it off for now, I am doing a small change & publishing it again later :)

#5 - 11/12/2009 12:11 AM - Anonymous

Should be ok now. Sorry for the spam :)

Stathis

#6 - 11/19/2009 08:40 AM - Anonymous

Ping!

Anyone interested with time and clue to review the patch ?
I'd like to push it at some point, before it gets too stale, but ain't gonna happen if at least someone else gives me a thumbs up.

Cheers,
Stathis

#7 - 11/20/2009 12:30 AM - dillon

:Stathis Kamperis <ekamperi@gmail.com> added the comment:
:
:Ping!
:
:Anyone interested with time and clue to review the patch ?
:I'd like to push it at some point, before it gets too stale, but ain't gonna
:a
:happen if at least someone else gives me a thumbs up.
:
:Cheers,
:Stathis

It looks committable to me. The only possible issue is whether the type field for mount -u operations can ever be NULL. It doesn't seem to ever be NULL so I think we're ok there.

-Matt

Matthew Dillon
<dillon@backplane.com>

#8 - 11/20/2009 07:37 AM - qhwt+dfly

On Thu, Nov 19, 2009 at 08:40:46AM +0000, Stathis Kamperis (via DragonFly issue tracker) wrote:

Stathis Kamperis <ekamperi@gmail.com> added the comment:

Ping!

Anyone interested with time and clue to review the patch ?
I'd like to push it at some point, before it gets too stale, but ain't gonna

happen if at least someone else gives me a thumbs up.

I noticed that I can't remount nullfs by mount(8). The patched mount_null accepts only one argument (the mountpoint to be remounted) when MNT_UPDATE is specified either by -u or by -o update, but what mount(8) passes to filesystem-specific mount commands like this: # mount -ur /mnt/pt
-> mount_\${foo} -o ro -o update /path/to/special /mnt/pt # mount -uw /mnt/pt
-> mount_\${foo} -o noro -o update /path/to/special /mnt/pt

I think that the following patch on top of yours does this for you, although I only tested it slightly.

```
diff --git a/sbin/mount_null/mount_null.c b/sbin/mount_null/mount_null.c
index b9c595a..9e02b6d 100644
--- a/sbin/mount_null/mount_null.c
+++ b/sbin/mount_null/mount_null.c
@ -88,29 +88,20 @ main(int argc, char **argv)
argv += optind;
```

```
/*
- * Only the mount point need be specified in update mode.
+ * Resolve target and source with realpath(3). Only the mount point
+ * needs be specified in update mode, but mount(8) passes us two
+ * arguments, the second of which is the source directory.
/
- if (mntflags & MNT_UPDATE) {
-     if (argc != 1) {
-         usage();
-         / not reached */
-     }
+ if ((mntflags & MNT_UPDATE) && argc == 1) {
+     args.target = NULL;
+     checkpath(argv[0], source);
-     error = getvfsbyname("null", &vfc);
-     if (error)
-         err(1, "getvfsbyname");
-     if (mount(vfc.vfc_name, source, mntflags, &args))
-         err(1, "mount");
-     exit(0);
- }

if (argc < 2)
+ } else if (argc == 2) {
+     args.target = target;
+     checkpath(argv[0], target);
+     checkpath(argv[1], source);
+ } else
+     usage();
```

```
- /* resolve target and source with realpath(3) /
- checkpath(argv0, target);
- checkpath(argv1, source);
-
```

```
/ * Mount points that did not use distinct paths (e.g. / on /mnt) * used to be disallowed because mount linkages were stored in
@ -118,7 +109,6 @ main(int argc, char **argv) * stores mount linkages in the namecache topology and does not * have this problem, so paths
no longer need to be distinct.
*/
- args.target = target;
```

```
error = getvfsbyname("null", &vfc);
if (error && vfc.vfsloadable("null")) {
```

#9 - 11/20/2009 07:52 AM - Anonymous

I noticed that I can't remount nullfs by mount(8). The patched mount_null accepts only one argument (the mountpoint to be remounted) when MNT_UPDATE is specified either by -u or by -o update, but what mount(8) passes to filesystem-specific mount commands like this:

1. mount ~~ur /mnt/pt~~
> mount_\${foo} -o ro -o update /path/to/special /mnt/pt
2. mount ~~uw /mnt/pt~~
> mount_\${foo} -o noro -o update /path/to/special /mnt/pt

I think that the following patch on top of yours does this for you, although I only tested it slightly.

I'm in the middle of a build. Once it finishes, I'll apply your modifications on top of my patch & re-run all the tests I've done so far, plus your test cases that involve mount(8).

If everything works and given that Dillon didn't object, I'll push it in the next couple of days.

Thanks!

Cheers,
Stathis

#10 - 11/22/2009 06:19 PM - Anonymous

Fixed with 8b02b69a65f3a7c5d344957f9ee3162b20077c93