

Re: Init.c, making it chroot

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/hackers/2006-12/msg00257.html>

- *From:* Robert Watson <rwatson@xxxxxxxxxxxxx>
 - *Date:* Sat, 30 Dec 2006 12:51:08 +0000 (GMT)
-

On Sat, 30 Dec 2006, Oliver Fromme wrote:

In particular, there is no /dev, so I still get this one from the kernel:

Lookup of /dev for devfs, error: 2

But then init and everything starts up fine, so it doesn't seem to cause any harm. That raises two questions:

1- Why does the kernel try to mount /dev at all? Why not simply let init mount it in all cases, with or without `init_chroot`? Would make things simpler. There doesn't seem to be a clear reason why the kernel needs to mount it. (Or maybe there `_are_` reasons, but they don't appear during my testing.)

2- Another solution would be to let `init(8)` autodetect whether /dev needs to be mounted. However, that might not be as trivial as it sounds.

The kernel needs to mount devfs because that's how it finds the device node to mount the root file system from. The bootstrap process is a bit complex, but basically what happens is this:

- (1) The kernel mounts devfs as /, and creates a /dev -> / symlink so that lookups in /dev using standard device names will work.
- (2) The kernel then attempts to mount the requested root file system using a device node from the devfs root.
- (3) Once a real root file system is successfully mounted, it performs a "fixup", which makes the new root file system the actual root file system, and re-grafts the original devfs mount onto /dev of the new root file system. It also removes the /dev/dev symlink.

The error message you're seeing is the kernel failing to find a /dev directory on the new root file system so having no where to regraft the boot-time /dev. You can see the logic for this in `vfs_mountroot_try()`,

Re: Init.c, making it chroot

devfs_first(), and devfs_fixup().

The point of all this is that we want the mount logic in every file system to be identical whether it's being used for the root file system or not. It used to be that only certain file systems could be used as a root file system, because only they knew how to bypass the lookup procedure to find their device node, short-circuiting to the in-kernel device list.

Normally when a file system is mounted, it is given a path to a device node, which it looks up using VFS and then mounts, so the root file system required quite special behavior. Since devfs doesn't require a source device to mount, being purely synthetic, this isn't an issue, and every file system now has exactly one mount routine that can make consistent assumptions about what it's being mounted on/with. The grafting logic is entirely a property of VFS, and while not pretty, it is fairly functional, and is not in per-file system code.

init contains fallback logic to mount a devfs instance if requested (-d), but I believe this exists largely to support upgrade transitions that may or may not still be relevant. Mounting a second devfs instance is undesirable for a number of reasons, not least that you end up with an extra file system floating around (although not reachable via the name space). It's certainly not disastrous though.

Robert N M Watson
Computer Laboratory
University of Cambridge

By the way, testing the whole thing is easy. Just install qemu from ports, then run this command:

```
qemu -monitor stdio -cdrom chroot-test.iso -boot d
```

Creating the ISO (with mkisofs) takes 5 seconds, and booting it in qemu takes 10 seconds (even without the qemu kernel accelerator module), so the development and testing cycles are very short. That's how I developed my CD/DVD boot manager "eltoro"[1]. As soon as the ISO runs successfully in qemu, I write it to a CD-RW and boot it on a real PC for verification.

Best regards
Oliver

PS: [1] <http://www.secnetix.de/products/eltoro/>

--

Oliver Fromme, secnetix GmbH & Co. KG, Marktplatz 29, 85567 Grafing
Dienstleistungen mit Schwerpunkt FreeBSD: <http://www.secnetix.de/bsd>
Any opinions expressed in this message may be personal to the author
and may not necessarily reflect the opinions of secnetix in any way.

"One of the main causes of the fall of the Roman Empire was that,
lacking zero, they had no way to indicate successful termination

Re: Init.c, making it chroot

Re: Init.c, making it chroot

of their C programs."

-- Robert Firth

freebsd-hackers@xxxxxxxxxxx mailing list

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

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

freebsd-hackers@xxxxxxxxxxx mailing list

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

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