Proot: chroot made easy

Marc Espie <espie@openbsd.org>, <espie@lse.epita.fr>

July 14, 2016

Proot: chroot made easy

Marc Espie <espie@openbsd.org>, <espie@lse.epita.fr>

July 14, 2016

What's that

Proot: ports chroot builder

- Preparing chroot environments
- For ports builds on OpenBSD

Help I'm falling asleep

Actually not so much

Why bother

- Existing tools don't match the needs
- It has to be real fast
- It must be damn-fool proof

Preparation preparation preparation

Copy what exactly

- Already have tools (locatedb) that tell us what comprises the base system, so we can copy from it.
- Alternately, start from a snapshot. Also have tools for that.
- Not even close to everything: forego manpages and X server.

One copy

How to do copies

- Speed: do not copy if it didn't change.
- Use hardlinks when we can. Cool and fast cloning of existing chroot

What about the rest

Not enough for a functional system

- you need files for the network
- and ldconfig
- and also devices

ttyname bug

Horrible code

```
static int
oldttyname(struct stat *sb, char *buf, size_t len)
{
        struct dirent *dirp;
        DIR *dp;
        struct stat dsb;

        if ((dp = opendir(_PATH_DEV)) == NULL)
                return (errno);

        while ((dirp = readdir(dp))) {
                if (dirp->d_fileno != sb->st_ino)
                        continue;
                if (dirp->d_namlen > len - sizeof(_PATH_DEV)) {
                        (void)closedir(dp);
                        return (ERANGE);
                }
                memcpy(buf + sizeof(_PATH_DEV) - 1, dirp->d_name,
                    dirp->d_namlen + 1);
                if (stat(buf, &dsb) || sb->st_dev != dsb.st_dev ||
                    sb->st_ino != dsb.st_ino)
                        continue;
                (void)closedir(dp);
                return (0);
        }
        (void)closedir(dp);
        return (ENOTTY);
}
```

ttyname bug 2

practice makes perfect

```
static int
oldttyname(struct stat *sb, char *buf, size_t len)
{
        struct dirent *dirp;
        DIR *dp;
        struct stat dsb;

        if ((dp = opendir(_PATH_DEV)) == NULL)
                return (errno);

        while ((dirp = readdir(dp))) {
                if (dirp->d_type != DT_CHR && dirp->d_type != DT_UNKNOWN)
                        continue;
                if (fstatat(dirfd(dp), dirp->d_name, &dsb, AT_SYMLINK_NOFOLLOW)
                    || !S_ISCHR(dsb.st_mode) || sb->st_rdev != dsb.st_rdev)
                        continue;
                (void)closedir(dp);
                if (dirp->d_namlen > len - sizeof(_PATH_DEV))
                        return (ERANGE);
                memcpy(buf + sizeof(_PATH_DEV) - 1, dirp->d_name,
                    dirp->d_namlen + 1);
                return (0);
        }
        (void)closedir(dp);
        return (ENOTTY);
}
```

ttyname bug 3

Fixes everywhere

- database makes things okay
- so run database
- AND also fix the code!

Design notes

Must be tweakable

- As a default, we remove unknown stuff
- Never under other mount points

Action man

- set of actions, some are default
- some can be added
- ...or removed
- everything needed, writes mk.conf

One size fits all ?

Not really

- ports clusters vary immensely
- because of architectures
- and needs !
- still require 50G+ for distfiles, 50G+ for packages
- takes one day for fast architectures

One size fits all ?

Not really

- ports clusters vary immensely
- because of architectures
- and needs !
- still require 50G+ for distfiles, 50G+ for packages
- takes one day for fast architectures
- I even wrote a manpage for those choices

The future

Individual chroot

- One per port, just requires knowing distfiles and packages we need
- hence the hardlinks

The future 2

Security model

- do not need root in the chroot
- make directories writable

Questions !!!