PRoot

From Termux Wiki
Jump to: navigation, search

PRoot is a user-space implementation of chroot (https://en.m.wikipedia.org/wiki/Chroot), mount --bind, and binfmt_misc. This means that users don't need any privileges or setup to do things like using an arbitrary directory as the new root filesystem, making files accessible somewhere else in the filesystem hierarchy, or executing programs built for another CPU architecture transparently through QEMU user-mode.

You can install PRoot with this command:

```
pkg install proot
```

Termux maintains its own version of PRoot, which is compatible with the latest Android OS versions. You can find its source code at https://github.com/termux/proot (https://github.com/termux/proot).

Important note: PRoot can make program to appear under root user id due to faking system call arguments and return values, but it does not provide any way for the real privilege escalation. Programs requiring real root access to modify kernel or hardware state will not work.

PRoot vs Chroot

The main different of chroot (https://en.m.wikipedia.org/wiki/Chroot) from PRoot is that it is native. Unlike PRoot, it does not use `ptrace()` for hijacking system call arguments and return values to fake the visible file system layout or user/group IDs. It does not cause overhead and works without issues on any device. However it requires superuser permissions.

If you have rooted device and want to have a better experience with using the Linux distributions in Termux, then use `chroot`. In this case get started with Linux Deploy (https://play.google.com/store/apps/details?id=ru.meefik.linuxdeploy) app for automated distribution installation. Things like shell can be used from Termux of course.

General usage information

The main purpose of PRoot is to run the Linux distributions inside Termux without having to root device. Simplest way to start a shell in a distribution chroot is:

```
unset LD_PRELOAD
proot -r ./rootfs -0 -w / -b /dev -b /proc -b /sys /bin/sh
```

Where:

- `unset LD_PRELOAD` - Termux-exec, execve() hook, conflicts with PRoot.
- `-r ./rootfs` - option to specify the rootfs where Linux distribution was installed.
- `-0` - tells PRoot to simulate a root user which expected to be always available in Linux distributions. This option will allow you to use package manager.
- `-b /dev -b /proc -b /sys` - make file systems at /dev, /proc, /sys appear in the rootfs. These 3 bindings are important and used by variety of utilities.
- `/bin/sh` - a program that should be executed inside the rootfs. Typically a shell.

You can learn more about options supported by PRoot by executing `proot --help`.

Installing Linux distributions

Termux provides a package proot-distro (https://github.com/termux/proot-distro) which takes care of management of the Linux distributions inside Termux. You can install this utility by executing

```
pkg install proot-distro
```

For now it supports these distributions:

- Alpine Linux (edge)
- Arch Linux / Arch Linux 32 / Arch Linux ARM
- Debian (stable)
- Fedora 35
- Manjaro AArch64
- OpenSUSE (Tumbleweed)
- Ubuntu (22.04)
- Void Linux

To install distribution, just run this command (assuming proot-distro is installed):

```
proot-distro install <alias>
```

where "<alias>" should be replaced by chosen distribution, e.g. "alpine". Note that it is expected that you have a stable Internet connection during installation, otherwise download may fail.

After installation, you can start a shell session by executing next command:

```
proot-distro login <alias>
```

Here is a basic overview of the available proot-distro functionality:

- `proot-distro list` - show the supported distributions and their status.
- `proot-distro install` - install a distribution.
- `proot-distro login` - start a root shell for the distribution.
- `proot-distro remove` - uninstall the distribution.
- `proot-distro reset` - reinstall the distribution.

Run `proot-distro help` for built-in usage information. Note that each of commands (with exception of "list") has own built-in usage information which can be viewed by supplying "--help" as argument. More detailed explanation about available functions you can find at project page: https://github.com/termux/proot-distro#functionality-overview (https://github.com/termux/proot-distro#functionality-overview)

Example of installing Debian and launching shell:

```
proot-distro install debian
proot-distro login debian
```

Community scripts

The ways of installation of Linux distributions in Termux are not limited to `proot-distro` only. There are lots of community created scripts, though their quality may be lower than that of the official Termux utilities provide and third-party stuff is generally out of the official Termux support.

Here is the list of some community-provided scripts:

- Alpine Linux - https://github.com/Hax4us/TermuxAlpine (https://github.com/Hax4us/TermuxAlpine)
- Debian - https://github.com/sp4rkie/debian-on-termux (https://github.com/sp4rkie/debian-on-termux)
- Fedora - https://github.com/nmilosev/termux-fedora (https://github.com/nmilosev/termux-fedora)
- Kali Nethunter - https://github.com/Hax4us/Nethunter-In-Termux (https://github.com/Hax4us/Nethunter-In-Termux)
- Slackware - https://github.com/gwenhael-le-moine/TermuxSlack (https://github.com/gwenhael-le-moine/TermuxSlack)
- Ubuntu - https://github.com/Neo-Oli/termux-ubuntu (https://github.com/Neo-Oli/termux-ubuntu)

If you decide to use third-party scripts, take the responsibility of potential risks on your own.