2023-06-20

Beyond Gentoo

I like Gentoo very much. Its multi-repo architecture makes it naturally decentralized, and hence it requires little CDN for maintainers.

However, Python is a terrible choice for implementing system-wide things. Portage is far from ideal. Instead, I want to have a shell-based package management system.

If this idea ever evolves into a full GNU/Linux distribution, its name should be Sashimi OS.

Also, the package management tool itself does not have to be implemented in shell script. It is nice to have alternative implementations written in Rust, Kotlin, or even Lua. I only require that the design of file formats are highly optimized for shell-based implementations.

It is nice to share infrastructure projects with other distributions. If anyone wants to maintain a separate distribution with tools developed here, I would be happy.

Package Management

Designing a brave new package management system is not easy. It is beyond my ability to rewrite Portage with shell script or Rust. It makes little sense to remain compatible with ebuild scripts from Gentoo.

Out of many existing solutions, Dpkg appears to be a nice backend for any high-level package management tool to cooperate with. It can solve many detail problems such as file collision and version bump, and does not impose stupid restrictions for frontend tools such as APT.

I need to design a package management system that produces ".deb" artifacts locally on a user machine, according to build scripts which are published in repositories. Also, the system should allow users to opportunistically use shared artifacts in lieu of locally generated artifacts as an optional convenience measure, so that some users may host binary mirrors in offices or homes.

If I ever implement this frontend tool, its name should be SPM (Sashimi Package Manager).

AOSC has a toolkit that works in a similar way. Ciel makes ".deb" artifacts but its functionalities depend on Systemd. While the designs of toolkits differ, I would be able to adopt the package specification format along with some build scripts.

I will be borrowing tools from AOSC, along with package definitions. Autobuild3 should to be a good choice for building deb artifacts. SPM will be using these tools.

Different Flavors

Some people want OpenRC but some other people want Systemd. Some people want GNU but some other people want BSD.

It is possible to serve multiple flavors with meta packages under the “distro-profile” category.

The user may install the “distro-profile/gnu-linux-openrc” package which adds “sys-gnu/coreutils”, “sys-kernel/linux-src”, and “sys-init/openrc” as dependencies.