NetBSD Base System Packaging Using pkg_* Tools

Yuuki Enomoto* and Ken'ichi Fukamachi**

Chitose Institute of Science and Technology, Chitose 066-8655, Japan

* m2160020@photon.chitose.ac.jp
** k-fukama@photon.chitose.ac.jp

Abstract

Unix operating system (OS) such as Linux distributions and BSD Unix is widely used mainly as servers and embedded systems. In Unix OS, speedy update is required especially to make the system up-to-date for security. For rapid update, OS should be fine granular. So, we need to package the operating system where “package” implies a set of software, documentation, configuration files and package’s meta data and “packaging” implies the OS is assembled on a lot of small packages to add or delete on demand. In Linux distributions such as Debian, Ubuntu and Red Hat Enterprise Linux, the userland programs are already divided into many small packages. On the other hand, BSD Unix such as FreeBSD and NetBSD are behind the curve on the system packaging.

The packaged OS is considered to have the following advantages: (1) On the packaged OS, it is easy and fast to replace a small part of the system. The rapidness is very important to keep OS up-to-date for secure service in daily operation. If a software is found to be vulnerable, server administrators and service provider must update(replace) the software as soon as possible. The packaged OS can provide the fast and easy way for it. (2) The packaged OS also provides rapid development on low spec hardware since it avoids a lot of compilations. It is favorable for embedded system since, in general, embedded hardware consists of low performance CPU, RAM and ROM.

Current status on OS packaging in Unix is as follows. In Linux distributions, the system is divided into many packages. Figure 1 shows the package file format in a package named “bash_4.3-14ubuntu1.1_amd64.deb”, which “deb” format is used on Debian-like Linux. The “debian-binary” file describe the version of the format. The

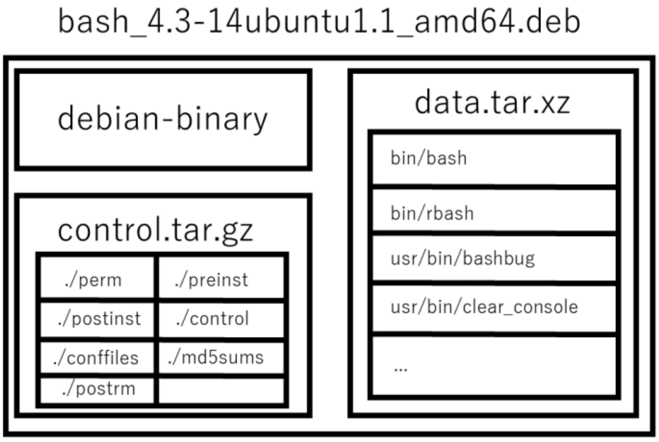

figure 1 Example of package

"control.tar.gz" file is an archive file consisting of meta information. The "data.tar.xz" file is an archive consisting of software, documentation and configuration files. The package is managed by "package manager" such as *apt* and *yum*. It is easy to update and customize the system by adding or removing packages. On the other hand, for historical reason, BSD Unix has not been packaging the system completely but in recent years OS packaging has been advancing. FreeBSD has introduced a system packaging mechanism "PkgBase"[1] which is under development. NetBSD had the "syspkg" mechanism similar to "PkgBase" but "syspkg" development has been stagnant[2] these years.

We focused on NetBSD for portability and license. Firstly, NetBSD supports a wide range of platforms (more than any other UNIX OS's). Secondly, NetBSD is shipped under the BSD (2-clause) license whereas Linux is GNU GENERAL PUBLIC LICENSE (GPL). Under the BSD license it is not necessary to open the source code of NetBSD and the software derived from NetBSD. Hence, NetBSD is considered to be suitable for commercial software product and service.

We have developed the software called "basepkg"[3] to make NetBSD base system packaged. This software is a substitute software for "syspkg". The current "basepkg" can provide the fundamental function for OS packaging. It divides the NetBSD system into 816 packages. It provides a simple mechanism to add or delete a part of OS base system rapidly by using a package management framework as the same way of 3rd party package management one based on "*pkg_** tools" software such as *pkg_add* and *pkg_delete*. In addition, "basepkg" is a shell script instead of "syspkg" makefile framework so that "basepkg" is small, readable and can be well maintained.

We have a lot of further development.

1. *sysinst* needs to be updated to use *pkg_add* during installation.
2. Evaluate how fast our framework works.
3. Construct the NetBSD base system package delivery network and verify the effectiveness of our packaging framework. We have been constructing it for the architecture "ARM" which is a typical example of embedded system.
4. Write the document about each package.
5. Develop kernel package it can easy upgrade and rollback.
6. "*pkg_** tools" must continuously be backward-compatible against previous versions.
7. Import to pkgsrc which is a framework for building third-party software on NetBSD and other Unix-like systems.

References

[1] https://wiki.freebsd.org/PkgBase (2017/09/17 accessed)
[2] https://wiki.netbsd.org/projects/project/syspkgs (2017/09/17 accessed)
[3] https://github.com/user340/basepkg (2017/09/17 accessed)