Maintain the NetBSD Base System Using pkg_* Tools

Yuuki Enomoto* Ken'ichi Fukamachi†

Abstract

This paper describes the script "basepkg.sh" for base system packaging to make NetBSD base system more granular.

Today, fine granular systems are expected to provide more rapid security update and more flexible customization in creating a very small base system for sensor network.

In "NetBSD", base system packaging mechanism called "syspkg" has been developed, but now, its development is stagnant. In addition, it is troubled to deal with "syspkg" consisting of a lot of Makefiles and shell scripts.

Thus we developed a shell script simpler than "syspkg" framework. This script uses src/distrib/sets/lists files and pkg_create command to generate a fine granular base package. We verified our system can provide minimum functionality that our package can replace a part of NetBSD base system. It provides the first step for more granular NetBSD base system.

1 Introduction

"Unix" such as Linux distribution and *BSD is used mainly as servers. It is very important to keep operating systems up to date for security. If a software is found to be vulnerable, server administrators must update the software as soon as possible.

This updating process should be rapid. So, it must be useful to replace a vulnerable software with the latest binary by using some rapid mechanism e.g. package managers.

Many Unix systems have its own package manager(s) to manage software. The scope managed by the package manager depends the system. Some developers consider that the whole system consists of packages, another consider that the operating system consists of the base and optional packages.

For example, Debian Linux packages the whole system to update and upgrade it easily. FreeBSD 11.0 introduced a framework called PkgBase[FreeBSD 2017]. to divide the

*Chitose Institute of Science and Technology, 758-90, Bibi, Chitose, Hokkaido, 066-8655, mailto: m2160020@photon.chitose.ac.jp https://e-yuuki.org/

†Chitose Institute of Science and Technology, 758-90, Bibi, Chitose, Hokkaido, 066-8655, mailto:k-fukama@photon.chitose.ac.jp http://www.nsrg.fml.org/

base system into 757 packages where pkg(8) can manage the FreeBSD base system.

On the other hand, NetBSD had developed "syspkg" under the same concept described above, but its development is stagnant[NetBSD 2017]. Currently updating a base software on NetBSD needs building from the source code (or replacing the binary if the major upgrade is available). It is no quick response. So, NetBSD base system packaging is required.

2 Comparison of Package Management Systems

"Package" is a collection of software, configuration files and documents. This collection is usually packed into a single archive file in the format .tar.gz, .zip, et.al. "Package Manager" is a software that manages package installation, deinstallation and updating.

Almost all Unix clone systems consist of a base mandatory system, auxiliary systems (e.g. compilers, documents, X11 Window) and optional 3rd party packages. The package manager can modify the system by adding or deleting optional binary packages.

2.1 Linux Distribution

"Debian" Linux and the derived distributions use "dpkg", "apt-get" and "apt" commands for package managements.

"CentOS" and the derived distributions use "rpm" and "yum" commands as package management systems.

There are a few Linux distributions (e.g. "Gentoo") where the package managers (e.g. "Portage") compile the source code and install it into the system.

2.2 FreeBSD

"FreeBSD" has a package system called "ports" to handle optional 3rd party packages[FreeBSD 2017]. "ports" consists of a lot of Makefiles to describe configurations. "pkg" command manages actual installation/deinstall action et.al.

By default, "ports" do not install the binary package directly, but "ports" can get and compile the source code, and install it into the system using the "pkg" command.

Also, "ports" can fetch and install binary packages provided by FreeBSD.

Instead of using "ports", FreeBSD users can use "pkg" command to manipulate packages.

2.3 NetBSD

NetBSD has a portable package system called "pkgsrc"[NetBSD 2017] derived from the FreeBSD "ports" system. "pkgsrc" is mainly developed for NetBSD but also can be available on a lot of platforms: NetBSD, FreeBSD, OpenBSD, DragonFlyBSD, Solaris, Linux, Darwin and other commercial UNIX systems. Some systems such as DragonFlyBSD use "pkgsrc" as the package management system by default.

"pkgsrc" is similar to FreeBSD "ports" but different to use another utilities called "pkg_*" to install or remove compiled software.

2.3.1 NetBSD pkg_* Utility

In NetBSD, the pkg_* utility consists of the following commands.

- pkg_add
- pkg_admin
- pkg_create
- pkg_delete
- pkg_info

"pkg_create" creates a binary package and saves it in "tgz" format. "pkg_add" installs and upgrades package(s) created using pkg_create. "pkg_delete" removes the specified package(s) from the system. "pkg_info" displays the package information. "pkg_admin" executes management tasks for the package system.

2.4 OpenBSD

OpenBSD package system is similar to NetBSD one. But the package utility is written in Perl not C. Commands such as pkg_add and pkg_delete are the same Perl script, which changes the behavior by the called argument.

OpenBSD users can write arbitrary package management scripts by using these Perl modules in /usr/libdata/perl5/OpenBSD/.

3 Base System Packaging

3.1 Base System Packaging

There are Linux distributions managing the base system as a collection of packages. Debian Linux is a typical one. Debian's "Essential" package provides the minimum functionality required to generate a small base system[Debian 2017]. Software and functions in the base system can be easily added and deleted by handling them as a set of packages.

In the case of BSD Unix(s), FreeBSD 11.0 introduced a mechanism called PkgBase. FreeBSD divides the base system into 757 packages for pkg(8) to be able to manage the base system.

Today NetBSD does not support base system packaging. NetBSD costs us a little bit of time to update a base software. We need to build from the source code and install it. Or we need to fetch and extract the binary distribution to overwrite the binary if the latest major distribution is available.

For example, consider that you want to install or remove only "groff" on your system.

We have roughly granular tar-balls on NetBSD where the base system is not packaged. We download text.tgz from the ftp site and unpack it in the root directory. So software other than groff will be installed together.

Instead of using text.tgz, we can install groff from "pkgsrc" (pkgsrc/textproc/groff). But in this case, after further updates, two "groff" will be installed such as in /usr/bin/groff and /usr/pkg/bin/groff. It may confuses users. Such a problem can be solved by packaging the base system with fine granularity.

NetBSD had developed "syspkg" under the same ideas described above, but its development is stagnant. Theoretically "syspkg" can manage the base system, but it is incomplete and not used well. We can run build.sh with the option -syspkgs to create a compiled base package, but the compiled package cannot be installed.

3.2 Benefits of Base System Packaging

A system packaging a base system with fine granularity must has advantages to enable rapid update and flexible customization of base system.

Typical examples needed for rapid security updates in base system are some critical libraries such as "OpenSSL" and fundamental daemons such as "OpenSSH" and "Postfix".

From the other point of view on flexibility, it looks that demand to create a small base system easily increases since cheap tiny devices for sensor network are wished. For example, omega2[Omega2 2016] is a $5 micro computer ready for Linux and FreeBSD. Also, Intel at CES in 2017 released a credit card size computer[Intel 2017]. But $5 must be too expensive to use billions of devices over the world. Devices with more small storage and memory are preferrable, so easily customized small operating systems must be required.

4 Yet Another NetBSD Base System Packaging

Instead of "syspkg", We have developed a "basepkg.sh"[Enomoto 2016]. It is a just shell script simpler than "syspkg" framework.

4.1 problems on "syspkg" framework

NetBSD source tree contains incomplete "syspkg" framework, but its development is stagnant.

The style of "syspkg" framework is traditional. It mainly consists of a lot of Makefiles under usr/src/distrib/sets/ and usr/src/distrib/syspkg/. It looks beyond our control to reconstruct the current "syspkg". We determined that it would be easier to develop another script that creates base packages.

Instead of traditional Makefile style, our script is a wrapper to use pkg_ tools based on information in usr/src/distrib/sets/lists/. Currently pkg_ tools is used only for "pkgsrc", but these tools are used for all software in the base system if we can prepare proper configuration files for pkg_* tools.

4.2 basepkg.sh

"basepkg.sh" creates a binary package based on information under "usr/src/distrib/sets/lists/". The script runs in two steps. (1) prepare configuration files based on "usr/src/distrib/sets/lists/". (2) call pkg_* tools to create packages actually.

4.2.1 Preparation of Configuration Files

Files under "usr/src/distrib/sets/lists/" contains program and configuration paths with classified information. Our script resolves the package name based on "usr/src/distrib/sets/lists/".

In addition, a configuration file is necessary to create a compiled package. basepkg.sh uses the following setting files, which is same as "pkgsrc" ones.

- +BUILD_INFO
  Information on the environment in which the package was created

- +COMMENT
  Comment of the package
- +CONTENTS
  Information on the path of the file to be stored in the package
- +DESC
  Description of the package

4.2.2 Package Creation

"basepkg.sh" calls pkg_create command to create the corresponding compiled packages in using configuration files created above.

The created compiled package can be installed using the pkg_add command. We can use the pkg_delete command to remove it from the system. pkg_info command shows information on the binary package. However, at this moment, useful information is not displayed since our package does not contain the corresponding +DESC file.

We confirmed that we can install compiled packages basepkg.sh creates on NetBSD-7.0.2/amd64 by using pkgsrc/pkgtools/pkg_install version 20160410.

5 Future Work

As a result of development, it has turned out that difficulty of base system packaging comes from package classification rather than package creation process.

Firstly, we need to investigate that the number of packages are proper or not. Currently "basepkg.sh" generates 879 binary packages as classified by usr/src/distrib/sets/lists. The number is almost same as one of FreeBSD base packages. But it is not clear that the graduality is proper for operation. We need to find the criterion of the graduality and classification. It is also required when we add a new software.

Secondly, packaging should be enhanced to run a hook since installation/deinstallation of shared library package needs the corresponding daemon restarting. We can use PRE-INSTALL and POST-INSTALL scripts of pkg_add command, but we should prepare the scripts, which are not covered by usr/src/distrib/sets/lists/. We need to arrange another configurations for such scripts.

Lastly, we need to fix the master database for package creation process. There are some empty packages since there are not corresponding binaries in +CONTENTS. For example, "comp-obsolete" package contains /usr/bin/atf-compile command, but there is no /usr/bin/atf-compile in the binary set of NetBSD-7.0.2/amd64.

6 Conclusion

We have developed a script "basepkg.sh" for base system packaging to make NetBSD base system more granular. It provides the first step for more granular NetBSD base system.

Fine granular systems become more important for rapid update and flexible customization of system in the future Internet and sensor networks. We believe that our system helps one small step for it.

Acknowledgments

I would like to thanks Ken'ichi Fukamachi, NetBSD developers and Japan

NetBSD Users Group.

References

[FreeBSD 2017]
https://www.freebsd.org/ports/

[NetBSD 2017]
http://www.netbsd.org/docs/software/packages.html

[Debian 2017]
https://www.debian.org/doc/manuals/debian-reference/ch02.html

[FreeBSD 2017]
https://wiki.freebsd.org/PkgBase/

[NetBSD 2017]
http://wiki.netbsd.org/projects/project/syspkgs/

[Omega2 2016]
https://www.kickstarter.com/projects/onion/omega2-5-iot-computer-with-wi-fi-powered-by-linux/description

[Intel 2017]
http://www.intel.com/content/www/us/en/compute-card/intel-compute-card.html

[Enomoto 2016]
https://github.com/user340/basepkg