Maintain the NetBSD Base System Using pkg_* Tools

Yuuki Enomoto
Ken’ichi Fukamachi
Chitose Institute of Science and Technology
Abstract

• This presentation explains “basepkg.sh”.

• Fine granular system can be easily updated and customized.

• “syspkg” consisting a lot of Makefiles and shell script.

• We have developed a script simpler than syspkg framework.
Index

1. Background
2. Case Study
3. Our Goals
4. Proposal Scripts
5. Outstanding Issues
6. Suggestions to Improve NetBSD Source Tree
7. Conclusion
Index

1. Background
2. Case Study
3. Our Goals
4. Proposal Scripts
5. Outstanding Issues
6. Suggestions to Improve NetBSD Source Tree
7. Conclusion
• In Unix, it is important to keep it up to date for security.

• Usually, we use “Package Manager” for up-to-date system.
  Ex. pkgsrc, ports, apt, yum, ...

• Third-party softwares can be updated using package manager.
Background 2/3

• But, how do we keep the base system up-to-date?

• This picture shows NetBSD-7.0.2 amd64 base binaries.
• This picture shows part of NetBSD base.tgz.

• It is an example of the base system.

• How do we manage these files?
Index

1. Background
2. Case Study
3. Our Goals
4. Proposal Scripts
5. Outstanding Issues
6. Suggestions to Improve NetBSD Source Tree
7. Conclusion
Case Study -- Debian GNU/Linux 1/2

• Debian is an example of base system management.

• Debian manages the base system and third-party software as a package.

• Debian package’s document can be viewed from the web.
Case Study -- Debian GNU/Linux 2/2

Package: base-passwd (3.5.37) [essential]

Debian base system master password and group files

These are the canonical master copies of the user database files (/etc/passwd and /etc/group), containing the Debian-allocated user and group IDs. The update-passwd tool is provided to keep the system databases synchronized with these master files.


Other Packages Related to base-passwd

- depends
- recommends
- suggests
- enhances

- dep: libc6 (>= 2.17) [arm64, ppc64el]
  GNU C Library: Shared libraries
  also a virtual package provided by libc6-udeb

- dep: libc6 (>= 2.8) [not arm64, ppc64el]

- dep: libdebcconfclient0 (>= 0.145)
  Debian Configuration Management System (C-implementation library)
Case Study -- FreeBSD 1/3

• FreeBSD has “PkgBase” since FreeBSD-11.0.

• This mechanism can manage the base system using “pkg”.

• outstanding issues are remaining.
  - # pkg delete -a
Case Study -- FreeBSD 2/3

$ make buildworld
$ make buildkernel
$ make package

FreeBSD PkgBase
Initial Setup

FreeBSD-acct-11.0_1.txz
FreeBSD-acpi-11.0_1.txz
package site.txz

'user'

$ make buildworld
$ make buildkernel
$ make package
Case Study -- FreeBSD 3/3
Case Study -- OpenBSD 1/2

• OpenBSD can’t manage the base system yet.

• But, OpenBSD’s package management mechanism serves as a useful reference.

• OpenBSD provides Perl modules. pkg_* tools developed by the modules.
Case Study -- OpenBSD 2/2

- OpenBSD’s `pkg_add` and `pkg_delete` command scripted by the same Perl code.

- Command jobs judged by the command name.

```
pkg_add
pkg_delete
pkg_info
pkg_create
...
```

OpenBSD Perl Modules

any works
In NetBSD…

- The mechanism called “syspkg” like PkgBase.

- The syspkg’s process included in the build.sh.
  - $ build.sh syspkg

- But syspkg project is not achieved.
syspkg 1/2

• syspkg creates many packages of the base system.

• NetBSD wiki(wiki.netbsd.org) says
  “There has been a lot of work in this area already, but it has not yet been finalized.”
  -- http://wiki.netbsd.org/projects/project/syspkgs/
syspkg 2/2

• syspkg consisting of a lot of Makefiles and shell scripts.
  - src/distrib/syspkg (Makefile -- bsd.syspkg.mk)
  - src/distrib/sets (database and shellscript)

• syspkg mechanism is difficult for the above reason.

• We thought we could develop another base system packaging system simpler than syspkg.
Index

1. Background
2. Case Study
3. Our Goals
4. Proposal Scripts
5. Outstanding Issues
6. Suggestions to Improve NetBSD Source Tree
7. Conclusion
Our Goals

• Develop a base system packaging program simpler than syspkg for NetBSD.

• Clarify the problems of base system packaging of NetBSD.

• It is possible to ...
  • easy to update.
  • granular custom installation more than current installer.
Index

1. Background
2. Case Study
3. Our Goals
4. Proposal Scripts
5. Outstanding Issues
6. Suggestions to Improve NetBSD Source Tree
7. Conclusion
Proposal for another Script

• We have developed “basepkg.sh” for packaged NetBSD. ([https://github.com/user340/basepkg](https://github.com/user340/basepkg))

• This script makes packages from NetBSD binaries.

• Packages can be installed to NetBSD (base) system.
Easy to make a base package.
  • Prepare information files, then running pkg_create command.

NetBSD’s base package use the following files.
  • +BUILD_INFO -- Information of the environment.
  • +COMMENT -- Comment of the package.
  • +CONTENTS -- Information on the path of the file.
  • +DESC -- Description of the package.
<table>
<thead>
<tr>
<th>Base Package</th>
<th>Arch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cobalt</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>emips</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>ews4800mips</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>hp300</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>hpcarm</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>hpcmips</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>hpcsh</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>hppa</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>i386</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>landisk</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>luna68k</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>mac64k</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>macppc</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>mipsco</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>mmeye</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>mvme68k</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>mvmepc</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>netwinder</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>news68k</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>newsmips</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>next68k</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>ofppc</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>pmax</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>prep</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>rs6000</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>sandpoint</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>sbmips</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>sgimips</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>shark</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>sparc</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>sparc64</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>sun2</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>sun3</td>
<td></td>
<td>md</td>
</tr>
<tr>
<td>vax</td>
<td></td>
<td>shl</td>
</tr>
<tr>
<td>x68k</td>
<td></td>
<td>ml</td>
</tr>
<tr>
<td>module.mi</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.ad.arm</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.ad.m68k</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.ad.m68k.shl</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.i386</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.mac68k</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.macppc</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.mi</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.shark</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.sparc</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.sparc64</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.sun2</td>
<td></td>
<td>rescue</td>
</tr>
<tr>
<td>rescue.vax</td>
<td></td>
<td>shl</td>
</tr>
<tr>
<td>shl.mi</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
About basepkg.sh 3/4

- The files are classified.

- basepkg.sh makes hierarchical directories from “list” file.

<table>
<thead>
<tr>
<th>Line</th>
<th>Directory</th>
<th>Root Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>.</td>
<td>base-sys-root</td>
</tr>
<tr>
<td>19</td>
<td>./altroot</td>
<td>base-sys-root</td>
</tr>
<tr>
<td>20</td>
<td>./bin</td>
<td>base-sys-root</td>
</tr>
<tr>
<td>21</td>
<td>./bin/[</td>
<td>base-util-root</td>
</tr>
<tr>
<td>22</td>
<td>./bin/cat</td>
<td>base-util-root</td>
</tr>
<tr>
<td>23</td>
<td>./bin/chgrp</td>
<td>base-util-root</td>
</tr>
<tr>
<td>24</td>
<td>./bin/chio</td>
<td>base-util-root</td>
</tr>
<tr>
<td>25</td>
<td>./bin/chmod</td>
<td>base-util-root</td>
</tr>
<tr>
<td>26</td>
<td>./bin/cp</td>
<td>base-util-root</td>
</tr>
<tr>
<td>27</td>
<td>./bin/cpio</td>
<td>base-util-root</td>
</tr>
<tr>
<td>28</td>
<td>./bin/csh</td>
<td>base-util-root</td>
</tr>
<tr>
<td>29</td>
<td>./bin/date</td>
<td>base-util-root</td>
</tr>
<tr>
<td>30</td>
<td>./bin/dd</td>
<td>base-util-root</td>
</tr>
<tr>
<td>31</td>
<td>./bin/df</td>
<td>base-util-root</td>
</tr>
<tr>
<td>32</td>
<td>./bin/domainname</td>
<td>base-nis-root</td>
</tr>
<tr>
<td>33</td>
<td>./bin/echo</td>
<td>base-util-root</td>
</tr>
</tbody>
</table>
How to install basepkg.sh

• Requirements
  – NetBSD Source (For src/sys/conf/osrelease.sh)
  – NetBSD Binaries tarball (From build.sh)
    Or download tarball from NetBSD ftp server, and extract tarballs at basepkg/work
      tar zxf base.tgz -C basepkg/work/base
      tar zxf comp.tgz -C baepkg/work/comp
      ...
  – pkgtools/pkg_install (From pkgsrc)
  – devel/git

$ git clone git://github.com/user340/basepkg
How to run basepkg.sh

$ ./basepkg.sh dir
$ ./basepkg.sh list
$ ./basepkg.sh pkg

• Packages are placed at basepkg/packages

• Use pkg_add to install package
basepkg.sh demonstration

1. Create packages using basepkg.sh

2. Install packages to NetBSD system on QEMU.

3. Reboot, and try to delete the part of base packages.
Index

1. Background
2. Case Study
3. Our Goals
4. Proposal Scripts
5. Outstanding Issues
6. Suggestions to Improve NetBSD Source Tree
7. Conclusion
basepkg.sh outstanding issues 1/5

- Overwrite files

- This picture shows that `/etc/rc.conf` is overwritten by "etc-sys-rc" package

- `rc_configured=NO`
basepkg.sh outstanding issues 2/5

• This picture shows that /etc/group and /etc/passwd files are overwritten by "etc-sys-sys" package.
“basepkg.sh” generates 879 (base) packages.

Is it clear that the granularity is proper for managing system?

What is criterion of the granularity and classification of packages?
There are some empty packages named "obsolete".

For example, "comp-obsolete" package.

We have to fix the empty packages.
basepkg.sh outstanding issues 5/5

• Where is base packages document like as Debian?

• It is desirable to be able to be browse on the web.

• Hard work.
Index

1. Background
2. Case Study
3. Our Goals
4. Proposal Scripts
5. Outstanding Issues
6. Suggestions to Improve NetBSD Source Tree
7. Conclusion
Suggestions to Improve of NetBSD Source Tree

- In pkgsrc, its functions in the its tree.

- But, syspkg’s functions not in the src/distrib/syspkg.

- List files which is categorize base system files are located at src/distrib/sets/lists.

- Why pull together syspkg’s functions into one tree?
Index

1. Background
2. Case Study
3. Our Goals
4. Proposal Scripts
5. Outstanding Issues
6. Suggestions to Improve NetBSD Source Tree
7. Conclusion
Conclusion

• We could manage the NetBSD base system using pkg_*.  
• But, outstanding issues are remaining.  
  • Unexpected overwrite files  
  • Granularity of packages  
  • Empty packages  
  • Documentation  

• Thank you for listening.