

Split debug symbols for pkgsrc builds

GSoC 2016 Project Proposal

Leonardo Taccari

\$Id: gsoc-debugpkg,v 1.10 2016/03/25 14:36:22 leot Exp \$

Rationale

The ability to debug software is important not just when developing but also when using it, e.g. post-mortem analysis. NetBSD provides `MKDEBUG` and `MKDEBUGLIB` variables that can be set in `mk.conf` in order to split debugging symbols for user-land applications and libraries. Resulting split debugging symbols are then available via the `debug.tgz` and `xdebug.tgz` installation sets. NetBSD also provides Rump and to some extent also `ddb(4)`, `ktrace(1)` and `DTrace` that ease analysis, tracing and debugging. All these features make NetBSD a great operating system in this regard. However, in pkgsrc it is possible only to generate packages with debugging symbols by providing proper `CFLAGS` for debugging and setting the `INSTALL_UNSTRIPPED` flag. This make debugging, especially for binary packages users, not very feasible.

About the project

A more convenient way - like what RPM and Debian package manager do - is to provide a `<package>-{debuginfo,dbg}` for `<package>` (where applicable) that includes all stripped debugging symbols. Thus, the project consists to add a mechanism in the pkgsrc infrastructure in order to make the generation of the `<package>-{debuginfo,dbg}` possible and transparent.

Providing `<package>-{debuginfo,dbg}` separately is also worth because debugging symbols have a cost in term of disk space needs. As an example, after quickly analyzing the space for NetBSD/amd64 -current installation sets it can be observed that `{debug,xdebug}.tgz` sets are (in MB):

```
$ ls -sk *debug.tgz | \
  awk '{ s += $1 } END { print (s / 1024) }'
```

492.578

...while all the other installation sets (not considering `kern-XEN3*.tgz` kernels and `{debug,xdebug}.tgz` installation sets) are (in MB):

```
$ ls -sk [bcegmt]* x[bcefs]* kern-GENERIC.tgz | \
  awk '{ s += $1 } END { print (s / 1024) }'
```

419.744

When extracted `{debug,xdebug}.tgz` need approximately 1.5GB.

Ideally all the packages that have `USE_LANGUAGES c` and/or `c99` and/or `c++` (and maybe also other programming languages) can be compiled with debugging symbols and it will be needed to automatically generate the `PLIST` for the `<package>-{debuginfo,dbg}` given the `PLIST` of the `<package>`. The `<package>-{debuginfo,dbg}` will be generated if a `mk.conf` variable is defined (e.g. `PKGSRC_MKDEBUG`) and also automatically installed. In order to handle them the `<package>-{debuginfo,dbg}` will `DEPENDS` on `<package>` (of course this can be argued because - strictly speaking - the `<package>-{debuginfo,dbg}` will not depends on any other packages but in practice they are useful only if the `<package>` is installed).

All the above would - hopefully - not need any change in the packages' Makefiles. For packages that for some reasons the debugging symbols can not be generated it will be needed to add a (per-package) Makefile variable in order to avoid the generation and splitting of the debug symbols (e.g. the various `emulators/suse*` packages). Also any package that presents some other problems or when generating debugging symbols isn't logical can use that variable to inhibit the generation of debug symbols.

Related works

As stated above NetBSD already supports splitting the debugging symbols. There are also some existing package management systems like RPM or the Debian package manager that support that. Both NetBSD `MKDEBUG*`, RPM and Debian package manager (and maybe also others) can probably be taken as a good source for inspiration and design.

Neither FreeBSD ports nor OpenBSD ports support splitting debug symbols in separate packages.

Deliverables

- design and implementation of the infrastructure in `pkgsrc` that handles the compilation, split of the debug symbols and generation of the `<package>-{debuginfo,dbg}` packages. Always pay attention to keep the `pkgsrc` portability in mind making the infrastructure compiler/debugger-agnostic as much as possible in order to be easily extensible (although due to limited GSoC time-frame it will be practically addressed only NetBSD-current with base `gcc` compiler).
- documentation of the interface in “The `pkgsrc` guide” targeting `pkgsrc` users, `MAINTAINERS` and developers. Also provide useful documentation for the on-line `pkgsrc` documentation via the “`help`” target.
- run a bulk build with the strip debug functionality turned on for at least a significant subset of packages on NetBSD in order to verify that the implemented infrastructure correctly works. This part will probably reveals problematic packages (e.g. package that ignore `CFLAGS`). Fixing these packages will also indirectly improve the hygiene of the `pkgsrc` ecosystem.

Project schedule

April 22, 2016 - May 22, 2016 (Community Bonding)

- get in contact with the mentor(s)
- get an overview regarding debugging symbols, debugging data formats and tools involved in handling them
- read and study pertinent documentation and code regarding how split debugging symbols are generated via `MKDEBUG*` in `src/share/mk`
- investigate and research existing solutions for other package management systems
- start to familiarize with the `pkgsrc` internals investigating possible sub-systems involved for the design and implementation of the debug strip functionality infrastructure
- discuss with the mentor(s) regarding any progress done and start brainstorming with her/him/them.

May 23, 2016 - June 20, 2016 (Students Work on their Project)

- start to design and implement an initial version of the infrastructure that automatically strip debug symbols from a `<package>` and generate a `<package>-{debuginfo,dbg}`
- verify that the implemented infrastructure works with few packages (without needing a bulk build)

June 20-27, 2016 (Midterm Evaluations)

June 27, 2016 - August 15, 2016 (Students Continue Coding)

- extend the implemented infrastructure as needed starting to test more packages
- document the interface for `pkgsrc` users, MAINTAINERS and developers in “The `pkgsrc` guide” and provide on-line documentation via the “help” target for all visible variables and targets
- start running bulk builds with the strip debug symbols functionality turn on for a significant subset of `pkgsrc` packages trying to address possible problems (e.g. packages that ignores `C*FLAGS`). Unfortunately the GSoC time-frame will probably not permit to fix a lot of them but at least a modus operandi regarding how fixing most common problems should be demonstrated.
- (only if time permits) empirically compare the difference in time and (disk) space needed by the bulk builds with and without the strip debug functionality
- (only if time permits) document a practical example demonstrating the installation of `<package>` and `<package>-{debuginfo,dbg}` and a sample debug session to illustrate the usage from a user perspective

August 15-23, 2016 (Students Submit Code and Evaluations)

- polish code and documentation if needed

I consider this potential GSoC project a great opportunity to work full-time on NetBSD and pkgsrc and, in particular, to become more familiar with internals and various subsystems, not just of pkgsrc. Last but not least, it's also a chance to cooperate with the NetBSD and pkgsrc community in a not-so-short-term project.

About me

I am studying for a Master Degree in Computing and Automation Engineering at Università Politecnica delle Marche in Ancona, Italy. I am also a recent NetBSD developer and have experience in particular as a package maintainer. I've also sporadically contributed in htdocs and src areas too.

Where to find this proposal and how to contact me

This proposal was first discussed on tech-pkg@ mailing list:

<http://mail-index.netbsd.org/tech-pkg/2016/03/14/msg016568.html>

It was polished and modified and the current version is available via the following URL:

<https://netbsd.org/~leot/misc/gsoc2016/gsoc-debugpkg>

In order to ease the review RCS is used and the corresponding RCS file is available via the following URL:

<https://netbsd.org/~leot/misc/gsoc2016/gsoc-debugpkg,v>

For questions, comments and suggestions please contact me via "leot at NetBSD dot org".