Add multi-packages support to pkgsrc
GSoC 2017 Project Proposal

Leonardo Taccari

April 3, 2017

Rationale

Multi-packages support permits the generation of multiple binary packages from a single pkgsrc package. [1]

This feature is needed to generate `<package>-debugpkg`s (packages containing debug symbols for `<package>`); improve organization and/or avoid code duplication (e.g. `tex-*` packages and their corresponding `tex-*-doc` packages) and avoid *extract*, *configure* phases of certain[2] packages multiple times.

About the project

During GSoC 2016 "Split debug symbols for pkgsrc builds" project [3] an initial attempt to provide multi-package support in pkgsrc was done for a very trivial package. However, what was done is far for being reviewable and useful in general.

I would like to continue the work started during GSoC 2016 in order to finally have debugpkgs support available in the short term and other benefits of `SUBPACKAGES` (the term multi-packages in pkgsrc is already used by packages that can be built with multiple versions of Apache, Python, PHP, etc.).

To do that several pkgsrc package variables and files will be need to be per-subpackage (e.g. `PKGNAME.<subpackage>`, `DEPENDS.<subpackage>`, `COMMENTS.<subpackage>`, `DESCR.<subpackage>`, etc.). Several variables and files that need to be adapted per subpackage are listed in GSoC 2016 debugpkg diary:

`https://www.NetBSD.org/~leot/gsoc2016-diary/debugpkg.txt`

The initial work done in GSoC 2016 consisted in adapting uses of such variables/files in an alternative code path that is traversed when the `SUBPACKAGES` variable (variable that contains a list of all possible subpackages for a given package) is defined. In that way there will be a `SUBPACKAGES` case and a `!SUBPACKAGES` case (i.e. for packages that doesn't have any subpackages).

This need to be properly completed, initially not considering any `{BUILD,TOOLS,}_DEPENDS`.

When that is done we can switch to an implicit (and hidden) subpackage in order to get rid of code duplication and having a single control flow relative to `SUBPACKAGES` [4].

[1] E.g. from a possible `misc/frobnitzes` in the pkgsrc tree, generation of `frobnitzes-foo`, `frobnitzes-bar`, `frobnitzes-baz` binary packages

[2] . . . the ones that can be converted to multi-packages.

[3] `https://summerofcode.withgoogle.com/archive/2016/projects/5609947500904448/`

[4] In other words: every package will always have at least one subpackage.

This will permit to have enough support for subpackages for `*-debugpkg`s and other trivial possible subpackages like `tex-*-doc`.

After that, a way to handle dependencies and buildlink3 should be designed and implemented in order to provide support for general subpackages.

For further information and what was done and what is needed to do please also give a look to "Split debug symbols for pkgsrc builds, final term GSoC 2016 notes":

`https://www.NetBSD.org/~leot/gsoc2016-diary/final-term-notes.txt`

Related works

OpenBSD ports support this feature and it is known as `MULTI_PACKAGES` and can be taken as a source for inspiration. Other package management systems that support multi-packages are RPM and the Debian package manager.

Deliverables

- design and implementation of the infrastructure in pkgsrc to support splitting of a package in multiple binary packages ("subpackages")

Project schedule

May 4, 2017 - May 29, 2017 (Community Bonding)

- get in contact with the mentor(s)
- sync "debugpkg" branch [5] with pkgsrc-current
- investigate and research existing solutions used in other package management systems
- discuss with the mentor(s) regarding any progress done and start brainstorming with her/him/them.

May 30, 2017 - June 30, 2017 (Phase 1)

- continue initial work done for `SUBPACKAGES` support considering the `SUBPACKAGES` and `!SUBPACKAGES` cases separately. This will duplicate the control flow although this will not impact existent `!SUBPACKAGES` flow and ease the initial implementation of subpackages support.

July 1, 2017 - July 28, 2017 (Phase 2)

- merge `SUBPACKAGES` and `!SUBPACKAGES` flows introducing an implicit (and hidden) subpackage for every packages. This will permit introducing trivial subpackages support in pkgsrc that does not have any per-subpackages dependencies like `*-debugpkg`s and `tex-*-doc` packages.

[5] `https://github.com/iamleot/pkgsrc/commits/debugpkg?author=iamleot`

July 29, 2017 - August 29, 2017 (Phase 3)

- design and implement a way to handle dependencies (`{BUILD,TOOLS,}_DEPENDS` and buildlink3 inclusions) granularly per-subpackage. E.g. a package `foo` can have an optional X11 support and install an X11 frontend in a separate binary and that binary can be splitted in a subpackage `foo-x11`. Only `foo-x11` will depends on `x11/libX11`.

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 NetBSD developer mainly working in pkgsrc.

I have partecipated in GSoC 2016, with The NetBSD Foundation, working on “Split debug symbols for pkgsrc builds” project.