

A large, light pink circular brushstroke graphic is centered in the background, partially overlapping the blue banner and the text below.

debug.debian.net


大統一 Debian 勉強会

岩松 信洋

iwamatsu@debian.org

2012年6月23日

Agenda

- 自己紹介
 - Debian デバッグ情報用パッケージの現状
 - Debian でのデバッグ情報パッケージについて
 - 実装内容について
 - 考えられる問題
 - 今後の課題
 - 質疑応答
- 



自己紹介

自己紹介

- 岩松信洋 (twitter: @iwamatsu)
- Debian Project Official Developer
- Bluetooth, Mozc, OpenCV, libpng, Renesas/SH porter
- 大統一 Debian 勉強会実行委員長... なんだぜ？

こんなことありませんか？

- パッケージにバグがあったのでデバッグしてやろうと思ったらデバッグ情報がなかった

こんなことありませんか？

- パッケージにバグがあったのでデバッグしてやろうと思ったらデバッグ情報がなかった
- 「しょうがない、デバッグ情報を有効にしてやるか」と思い、再ビルドしたら再現しなかった

こんなことはありませんか？

- パッケージにバグがあったのでデバッグしてやろうと思ったらデバッグ情報がなかった
- 「しょうがない、デバッグ情報を有効にしてやるか」と思い、再ビルドしたら再現しなかった
- 古いパッケージだったので、ビルド用の環境構築がめんどろ




どうしたらいいのか、数秒
考えてみた。

A large, stylized pink brushstroke graphic that forms a circular shape with a smaller circle inside, resembling a swirl or a decorative element. It is positioned in the background, partially overlapping the text.

とりあえず実装してみた、
というのが今回の発表です。

デバッグ情報とは

- 内部シンボル
 - 型の情報
 - ソースコードの行番号
 - etc
- 

デバッグ情報があると良い点

- デバッガ（GDB）を使ったデバッグでより詳細なデバッグ情報を得ることができる
- デバッグ情報を含めたバイナリを再度ビルドする必要がない
- バイナリベースディストリビューションの場合、実際のバイナリとデバッグ情報が常に対になるので、バグの再現性が高くなる

デバッグ情報があると悪い点

- 実行ファイルにデバッグ情報が含まれるので実行ファイルのサイズが大きくなる
- デバッグしない人にとっては不要なものが含まれることになる

ようするにだな

- すべての実行ファイルのデバッグ情報が提供されており
- ユーザにとって必要のない情報がインストールされない仕組みがあれば
- デバッグ情報パッケージはとても有益なものになるはず !!



Debian デバ
ッグ情報用
パッケージ
の現状

Debian デバッグ情報用パッケージの現状



Debian デバッグ情報用パッケージの現状

- Debian ではいくつかのソースパッケージからデバッグ情報を含んだパッケージが提供されている
- パッケージには`-dbg` というサフィックスが付いている
- 特にデバッグ情報パッケージの提供に関するポリシーは決まっておらず、提供に関してはパッケージメンテナ次第


Debian デバッグ情報用パッケージの現状

2012/06/20 の時点での main

- バイナリパッケージ数 : 37520 個
- ソースパッケージ数 : 18306 個
- -dbg パッケージ数 : 1736 個
- -dbg を提供しているソースパッケージ数 : 1401 個

Debian デバッグ情報用パッケージの現状

なぜ今まで提供されていなかったのか



Debian デバッグ情報用パッケージの現状

なぜ今まで提供されていなかったのか

- ディスク容量の問題
- 回線の問題
- そんなもん自分でビルドしろ、という風潮があった（ように思う）

これらはすでに過去の話。

Debian デバッグ情報用パッケージの現状

他のディストリビューションは？



Debian デバッグ情報用パッケージの現状

他のディストリビューションは？

- Fedora では `-debuginfo` というサフィックスを持ったパッケージが提供されている
- Gentoo ではデフォルトでこれらの情報を生成し管理する仕組みがある `features = splitdebug` を指定

デバッグ情報の提供という点に関して他のディストリビューションに遅れを取っている。

Debian デバッグ情報用パッケージの現状

他のディストリビューションは？

- Fedora では `-debuginfo` というサフィックスを持ったパッケージが提供されている
- Gentoo ではデフォルトでこれらの情報を生成し管理する仕組みがある `features = splitdebug` を指定

デバッグ情報の提供という点に関して他のディストリビューションに遅れを取っている。

これは問題。



Debian での
デ バグ
情 報 パッ
ケージにつ
いて

Debian でのデバッグ情報パッケージについて

- Debian ではデバッグ情報パッケージは `-dbg` というサフィックスがついたパッケージ名を持つ

Debian でのデバッグ情報パッケージについて

- Debian ではデバッグ情報パッケージは `-dbg` というサフィックスがついたパッケージ名を持つ
- `foo` というアーキテクチャ依存のパッケージがあった場合、`foo` のデバッグ情報を持ったパッケージ名は `foo-dbg` になる

Debian でのデバッグ情報パッケージについて

- Debian ではデバッグ情報パッケージは `-dbg` というサフィックスがついたパッケージ名を持つ
- `foo` というアーキテクチャ依存のパッケージがあった場合、`foo` のデバッグ情報を持ったパッケージ名は `foo-dbg` になる
- これらのパッケージに含まれるデータはどのように作成されるのか？

Debian でのデバッグ情報パッケージについて

デバッグ情報ファイル作成手順

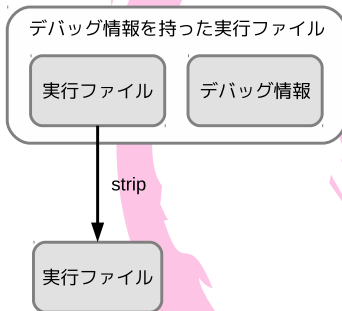
デバッグ情報を持った実行ファイル

実行ファイル

デバッグ情報

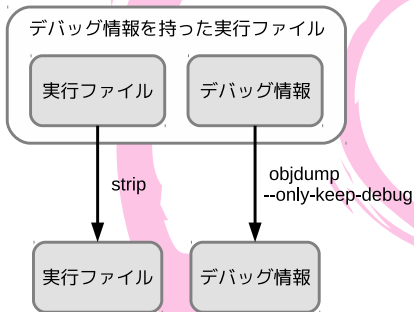
Debian でのデバッグ情報パッケージについて

デバッグ情報ファイル作成手順



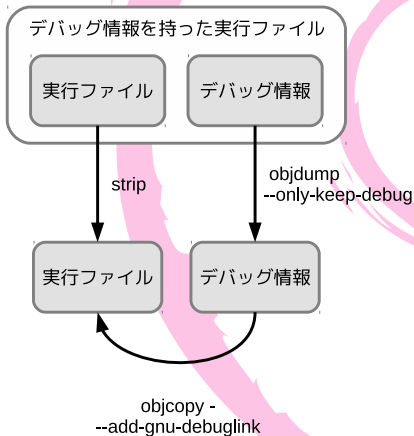
Debian でのデバッグ情報パッケージについて

デバッグ情報ファイル作成手順



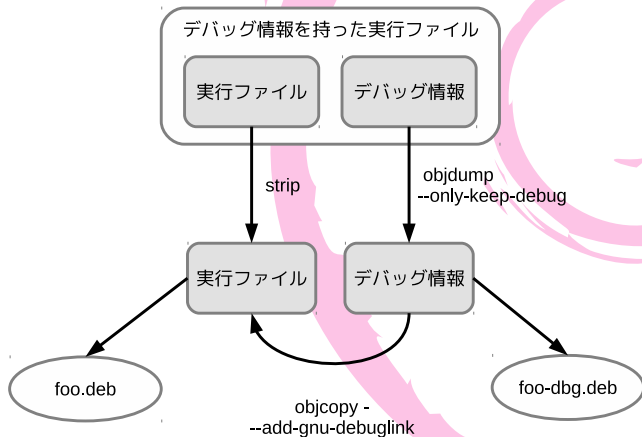
Debian でのデバッグ情報パッケージについて

デバッグ情報ファイル作成手順



Debian でのデバッグ情報パッケージについて

デバッグ情報ファイル作成手順





実装内容について

全ての実行ファイルのデバッグ情報を提供する
には？



全ての実行ファイルのデバッグ情報を提供する には？

全てのパッケージで strip された実行ファイルとデバッグ
情報ファイルを持ったパッケージを構築すればよい

Debian パッケージ構築順序



Debian パッケージ構築順序

- パッケージ作成過程のどこかに手を加える必要がある

Debian パッケージ構築順序

- パッケージ作成過程のどこかに手を加える必要がある
- キーとなる処理の動作を簡単に説明

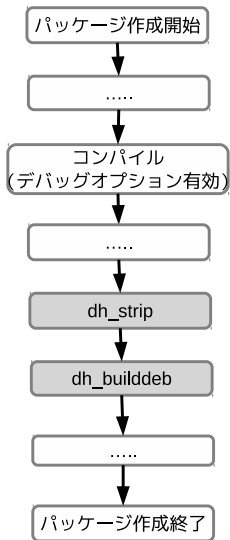
Debian パッケージ構築順序

- パッケージ作成過程のどこかに手を加える必要がある
- キーとなる処理の動作を簡単に説明
 - `dh_strip`

Debian パッケージ構築順序

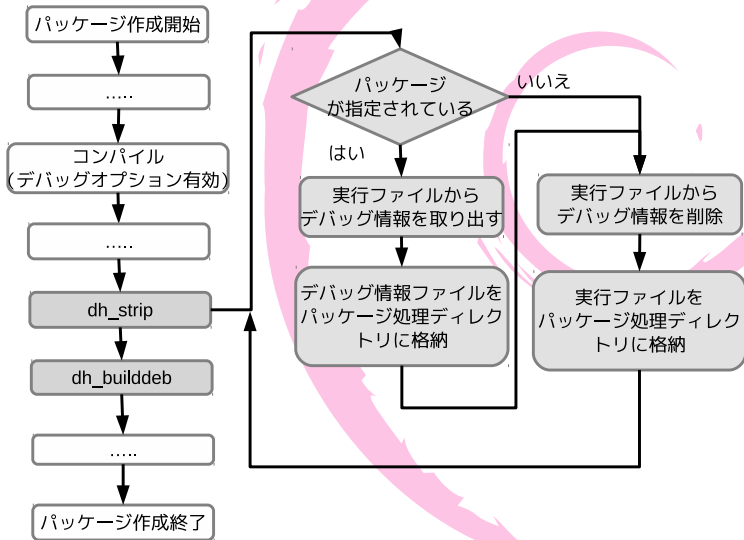
- パッケージ作成過程のどこかに手を加える必要がある
- キーとなる処理の動作を簡単に説明
 - dh_strip
 - dh_builddeb

Debian パッケージ構築順序



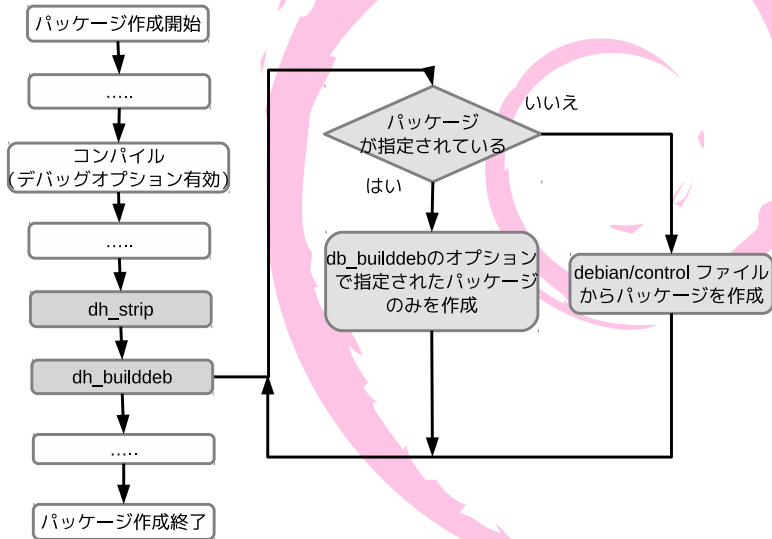
Debian パッケージ構築順序

dh_strip



Debian パッケージ構築順序

dh_builddeb



問題 1

自動生成したいデバッグ情報パッケージの情報をどのように生成するか

問題 1

自動生成したいデバッグ情報パッケージの情報をどのように生成するか

- Debian パッケージは `debian/control` に記述されている情報を元に生成される
- `debian/control` に生成されるパッケージ情報が書かれている必要がある

対策: 自動生成したいデバッグ情報パッケージの情報をどのように生成するか

- `dh_strip` の処理の先頭で `debian/control` ファイルにデバッグ情報パッケージファイルに関する情報を追記する処理を追加
- デバッグ情報パッケージは対象のパッケージがアーキテクチャ依存 (Architecture: all ではない) である事とパッケージ名さえわかれば、パッケージ情報は自動生成できる

対策: 自動生成したいデバッグ情報パッケージの情報をどのように生成するか

```
Package: hoge
Architecture: any
Depends: ${misc:Depends}
Description: hoge
```

```
Package: hoge-dbg
Architecture: any
Section: debug
Priority: extra
Depends: hoge (= ${binary:Version}), ${misc:Depends}
Description: debugging symbols for hoge
    This package contains the debugging symbols for hoge
```

問題2

dh_strip でデバッグ情報パッケージが既に指定されている場合、どのようにしてデバッグ情報用パッケージを自動生成するか

問題2

dh_strip でデバッグ情報パッケージが既に指定されている場合、どのようにしてデバッグ情報用パッケージを自動生成するか

- dh_strip は指定されているパッケージのみ、デバッグ情報ファイル作成処理を行う
- 指定されていない場合は処理されない
- 既にデバッグ情報用パッケージが提供されている場合でも、すべて提供されているわけではない。(ライブラリ用は提供されているが、ツール用は提供されていない場合がある)

対策: `dh_strip` でデバッグ情報パッケージ指定されている場合、自動生成したい `-dbg` パッケージ用のファイルをどのように生成するか

- アーキテクチャ依存の Debian パッケージは必ず `dh_strip` が呼ばれるため、ここで処理をフックしてしまえば、デバッグ情報を提供するパッケージと用のデータと `strip` されたバイナリデータを分けることができる

対策: `dh_strip` でデバッグ情報パッケージ指定されている場合、自動生成したい `-dbg` パッケージ用のファイルをどのように生成するか

- アーキテクチャ依存の Debian パッケージは必ず `dh_strip` が呼ばれるため、ここで処理をフックしてしまえば、デバッグ情報を提供するパッケージと用のデータと `strip` されたバイナリデータを分けることができる
- 今回は `dh_strip` の中身を改造し、全てのアーキテクチャ依存のパッケージ用のデータを作成するように変更

対策: `dh_strip` でデバッグ情報パッケージ指定されている場合、自動生成したい `-dbg` パッケージ用のファイルをどのように生成するか

- アーキテクチャ依存の Debian パッケージは必ず `dh_strip` が呼ばれるため、ここで処理をフックしてしまえば、デバッグ情報を提供するパッケージと用のデータと `strip` されたバイナリデータを分けることができる
- 今回は `dh_strip` の中身を改造し、全てのアーキテクチャ依存のパッケージ用のデータを作成するように変更
- `dh_strip` でパッケージが指定されていても無視するように処理。なんでも処理をするように変更

問題3

自動生成したいデバッグ情報パッケージそのものをどのように生成するか

問題3

自動生成したいデバッグ情報パッケージそのものをどのように生成するか

- `dh_strip` で指定されているものしかデバッグパッケージ処理を行わない
- `dh_strip` 内でデバッグ情報が削除されてしまうため、削除される前にデバッグ情報を取得する必要がある

対策: 自動生成したいデバッグ情報パッケージ そのものをどのように生成するか

- デバッグ情報パッケージの作成は `dh_strip` 内で `dh_builddeb` を呼び出すことで対応。
- パッケージ名とパッケージ作成に必要なデータは揃っているため、`dh_builddeb -p デバッグ情報パッケージ名` を実行することで、パッケージが作成させる

これって debhelper 依存じゃないの？



¹<http://people.debian.org/~cjwtatson/dhstats.png>

これって debhelper 依存じゃないの？

- これらの処理が行える前提条件として、debhelper に依存しているパッケージが対象になる
- 現在ほとんどのパッケージが debhelper か CDBS に依存しており、CDBS は debhelper と同時に使う事が多いため、問題はない。¹

¹<http://people.debian.org/~cjwtatson/dhstats.png>

パッケージサイズへの対応



パッケージサイズへの対応

- デバッグ情報は非常に大きい

パッケージサイズへの対応

- デバッグ情報は非常に大きい
- strip されたバイナリの数倍以上のサイズになることもめずらしくない

パッケージサイズへの対応

- デバッグ情報は非常に大きい
- strip されたバイナリの数倍以上のサイズになることもめずらしくない
- どれくらい大きいのか？

パッケージサイズへの対応

- デバッグ情報は非常に大きい
- strip されたバイナリの数倍以上のサイズになることもめずらしくない
- どれくらい大きいのか？
- libjpeg8 で提供される libjpeg.so.8.4.0 のファイルの場合

パッケージサイズへの対応

- デバッグ情報は非常に大きい
- strip されたバイナリの数倍以上のサイズになることもめずらしくない
- どれくらい大きいのか？
- libjpeg8 で提供される libjpeg.so.8.4.0 のファイルの場合
 - strip 前 : 約 1.3MB

パッケージサイズへの対応

- デバッグ情報は非常に大きい
- strip されたバイナリの数倍以上のサイズになることもめずらしくない
- どれくらい大きいのか？
- libjpeg8 で提供される libjpeg.so.8.4.0 のファイルの場合
 - strip 前 : 約 1.3MB
 - strip 後 : 236KB

パッケージサイズへの対応

- デバッグ情報は非常に大きい
- strip されたバイナリの数倍以上のサイズになることもめずらしくない
- どれくらい大きいのか？
- libjpeg8 で提供される libjpeg.so.8.4.0 のファイルの場合
 - strip 前 : 約 1.3MB
 - strip 後 : 236KB
 - デバッグ情報ファイル : 1.1MB

パッケージサイズへの対応

- パッケージサイズが大きく異なるので、パッケージを分けてもユーザが利用しているパッケージリポジトリと同じ場所・方法で提供してしまうと、ミラーに時間がかかるようになる
- ユーザに不要なデータが格納されたりリポジトリ情報を持たせることになる

パッケージサイズへの対応

- この問題を回避するためにリポジトリを分けることで対応
- unstable で strip されているパッケージ（通常のパッケージ）は unstable とだけ指定し、デバッグ情報を提供するパッケージは unstable/debug を指定する

```
deb http://cdn.debian.or.jp/debian/ unstable main non-free  
deb http://cdn.debian.or.jp/debian/ unstable/debug main
```


- デバッグ情報が必要なユーザは unstable/debug を apt-line に追加することによってデバッグ情報用のパッケージが利用できるようになる

パッケージサイズへの対応

- パッケージが格納されるディレクトリのパスを変更することによって、デバッグ情報のみを提供するミラーを構築することができ、debug 情報をミラーしないミラーサーバの負荷も今までと変わらない

実装後について

- これらを実装したシステムを `reprepro + sbuild + rebuildd` で構築した。
- 現在 `stable/amd64` のみをターゲットテストとして動作中。



考えられる
問題

セキュリティの問題

- ソースパッケージから作成されたバイナリパッケージの一覧は .changes ファイルにファイルのハッシュと共に記述され、どのソースパッケージ (orig.tar.gz, dsc.diff.gz) から作成されたのか分かるようになっている

セキュリティの問題

- ソースパッケージから作成されたバイナリパッケージの一覧は `.changes` ファイルにファイルのハッシュと共に記述され、どのソースパッケージ (`orig.tar.gz`, `dsc.diff.gz`) から作成されたのか分かるようになっている
- 現時点での実装はバイナリパッケージ作成の課程で自動生成されるためのこれらの情報とリンクしない

セキュリティの問題

- ソースパッケージから作成されたバイナリパッケージの一覧は .changes ファイルにファイルのハッシュと共に記述され、どのソースパッケージ (orig.tar.gz, dsc.diff.gz) から作成されたのか分かるようになっている
- 現時点での実装はバイナリパッケージ作成の課程で自動生成されるためのこれらの情報とリンクしない
- 仮に Debian で提供される場合、BuildD 上でデバッグ情報ファイルパッケージが生成されるので個人的に問題ないと思っているが、これらを紐付けるシステムがあるほうがより安全と言える

バイナリの不一致

- 既存のシステムでは、strip されたバイナリと デバッグ情報が一致しないため、オフィシャルのバイナリと混ぜて使えないことが考えられる

バイナリの不一致

- 既存のシステムでは、strip されたバイナリと デバッグ情報が一致しないため、オフィシャルのバイナリと混ぜて使えないことが考えられる
- 私が提供しているされているデバッグ情報パッケージを使う場合、私が提供して通常のパッケージも利用しないと意味がない

バイナリの不一致


- 既存のシステムでは、strip されたバイナリと デバッグ情報が一致しないため、オフィシャルのバイナリと混ぜて使えないことが考えられる
- 私が提供しているされているデバッグ情報パッケージを使う場合、私が提供して通常のパッケージも利用しないと意味がない
- 今はこれをバージョンによる依存関係で回避

バイナリの不一致


- 既存のシステムでは、strip されたバイナリと デバッグ情報が一致しないため、オフィシャルのバイナリと混ぜて使えないことが考えられる
- 私が提供しているされているデバッグ情報パッケージを使う場合、私が提供してる通常のパッケージも利用しないと意味がない
- 今はこれをバージョンによる依存関係で回避
- 最終的には buildd に入れてもらうことでデバッグ情報パッケージを自動生成することを考えている



大きな落とし穴




ヤッター！デキタヨー！思っていたら大きな落とし穴が待っていたのである...



そもそも `debug.debian.net` があるんじゃない？

そもそも debug.debian.net があるんじゃない？

私のような凡人が考えるようなことは先人達はすでに考えているわけでした....



そもそも debug.debian.net があるんじゃない？

私のような凡人が考えるようなことは先人達はすでに考えているわけでして....

既に <http://debug.debian.net> というサービスがあり実装され、そして終了していました。

そもそも `debug.debian.net` があるんじゃないか？

- 実装と考えもほとんど同じ
`dh_strip` ですべてを処理するという考え

そもそも debug.debian.net があるんじゃないか？

- 実装と考えもほとんど同じ
dh_strip ですべてを処理するという考え
- 大きく違うところはデバッグ情報パッケージのサフィックスが -dbg ではなく、 -dbgsym である

そもそも debug.debian.net があるんじゃない？

- 実装と考えもほとんど同じ
dh_strip ですべてを処理するという考え
- 大きく違うところはデバッグ情報パッケージのサフィックスが -dbg ではなく、-dbgsym である
- デバッグ情報を作成する部分が dh_builddeb ではなく、dpkg-deb を使っている

そもそも debug.debian.net があるんじゃない？

- 実装と考えもほとんど同じ
dh_strip ですべてを処理するという考え
- 大きく違うところはデバッグ情報パッケージのサフィックスが -dbg ではなく、-dbgsym である
- デバッグ情報を作成する部分が dh_builddeb ではなく、dpkg-deb を使っている
- dh_strip を直接変更するのではなく、シンボリックリンクで機能をオーバーライドさせている

そもそも debug.debian.net があるんじゃない？

- 実装と考えもほとんど同じ
dh_strip ですべてを処理するという考え
- 大きく違うところはデバッグ情報パッケージのサフィックスが -dbg ではなく、-dbgsym である
- デバッグ情報を作成する部分が dh_builddeb ではなく、dpkg-deb を使っている
- dh_strip を直接変更するのではなく、シンボリックリンクで機能をオーバーライドさせている
- ファイル拡張子が ddeb

そもそも debug.debian.net があるんじゃない？

- こちらのほうが debhelper に手を加えなくて済むのでこちらに乗り換え

そもそも debug.debian.net があるんじゃない？

- こちらの方が debhelper に手を加えなくて済むのでこちらに乗り換え
- 彼に連絡を取り、debug.debian.net 再稼働させた

そもそも debug.debian.net があるんじゃない？

- こちらの方が debhelper に手を加えなくて済むのでこちらに乗り換え
- 彼に連絡を取り、debug.debian.net 再稼働させた
- この発表が行われている頃には稼働しているはず、まだしていなかった

<http://debug.debian.net> ヘデータコピーできてない

<http://debug.nigauri.org> でビルド結果を参照可能



今後の課題

今後の課題

- このままスタンドアロンでデバッグ情報パッケージを提供しても無駄なバイナリを生成するだけ

今後の課題

- このままスタンドアロンでデバッグ情報パッケージを提供しても無駄なバイナリを生成するだけ
- buildd にこのシステムを入れてもらい、dak で ディストリビューション/debug として処理してもらうことが今後の大きな課題

今後の課題

- このままスタンドアロンでデバッグ情報パッケージを提供しても無駄なバイナリを生成するだけ
- buildd にこのシステムを入れてもらい、dak で ディストリビューション/debug として処理してもらうことが今後の大きな課題
- このことについて来月開催される Debconf 12 で BOF または FTP チーム、wanna-buildd チームと話をする予定

今後の課題

- このままスタンドアロンでデバッグ情報パッケージを提供しても無駄なバイナリを生成するだけ
- buildd にこのシステムを入れてもらい、dak で ディストリビューション/debug として処理してもらうことが今後の大きな課題
- このことについて来月開催される Debconf 12 で BOF または FTP チーム、wanna-buildd チームと話をする予定
- 誰か速いマシン貸してください

A large, stylized pink circular brushstroke graphic that frames the text on the right side of the image.

質疑応答

なにか質問はありますか？

ありがとうございました