

A large, stylized pink brushstroke graphic that forms a circular shape, resembling a smile or a large letter 'D', is positioned in the background of the slide.

東京エリア Debian 勉強会

資料

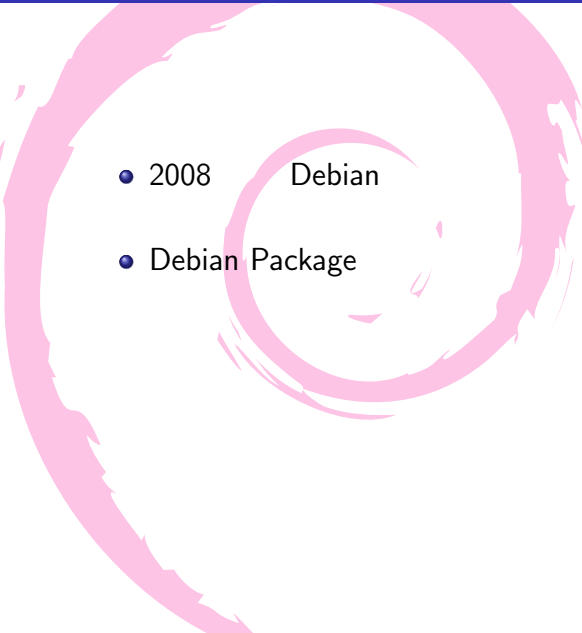
上川 純一 dancer@debian.org
IRC nick: dancerj

2008年1月19日



設営準備に
ご協力くだ
さい

Agenda

- 注意事項
 - 飲食禁止
 - 政治/宗教/営利活動禁止
 - quiz
 - 最近の Debian 関連のイベント
 - 前回
 - 2008 年度 Debian 勉強会企画
 - Debian Package 管理の流れ
- 

前回のアジェンダ

- 注意事項
 - 飲食禁止
 - 政治/宗教/営利活動禁止
- quiz
- 最近の Debian 関連のイベント
 - 前回
- Debian 勉強会資料は
いかにつくられているか
- 2007 年度の勉強会を
ふりかえる
- 2008 年度計画ワーク
ショップ

前々回のアジェンダ

- quiz
- 最近の Debian 関連のイベント
 - 前回
 - OSC Tokyo/Fall
 - KOF
 - Biella 宴会
- bluetooth
- livehelper
- tomoyo kernel module
- KOF
- OSC Tokyo/Fall
- 今後の計画

Debian 常識クイズ

Debian の常識、もちろん知ってますよね？ 知らないなんて
恥ずかしくて、知らないとは言えないあんなことやこんな
こと、みんなで確認してみましよう。

今回の出題範囲は、

<http://lists.debian.org/debian-devel-announce/> に
ある最近のアナウンス文書です。

問題1. Debian Miniconf 7が開催される、Linux Conf Au はいつ開始か

-  A 1月28日
-  B 1月29日
-  C 1月19日

問題1. Debian Miniconf 7が開催される、Linux Conf Au はいつ開始か

-  A 1月28日
-  B 1月29日
-  C 1月19日

答えは:



A

問題2. lintian.debian.org は全アーカイブに対して実行した lintian の実行結果を報告してくれているページだが、メンテナ単位のウェブページのURLが変更になった。どうかわったか？



A

<http://www.youtube.com/email.html>



B

<http://people.ubuntu.com/liw/lintian/gutsy-i386-main/reports/maintainer/email.html>



C

<http://lintian.debian.org/reports/maintainer/email.html>

問題2. lintian.debian.org は全アーカイブに対して実行した lintian の実行結果を報告してくれているページだが、メンテナ単位のウェブページのURLが変更になった。どうかわったか？

答えは:



A

<http://www.youtube.com/email.html>



B

<http://people.debian.org/~gutsy-i386-main/reports.html>






C




<http://lintian.debian.org/maintainer/email.html>



問題3. dpkg のシンボルファイルで何が実現できるか？

-  A 共有ライブラリなんてつかってられないので全部DLLにしてみた依存関係
-  B 共有ライブラリはどんなバージョンでもよくなるような依存関係
-  C 共有ライブラリのバージョン付きシンボルをベースにした依存関係

問題3. dpkg のシンボルファイルで何が実現できるか？

- 
- 
- 

A 共有ライブラリなんてつかってられないので全部DLLにしてみた

B 共有ライブラリバージョンごとの依存関係


C 共有ライブラリ付きシンボル付きの依存関係

答えは:



C

問題 4. <http://wiki.debian.org/HelpDebian/Start> には何がかかっているか

-  A なぜ Debian に貢献するべきか
-  B Debian はなぜ存在するのか
-  C GNU の存在意義

問題 4. <http://wiki.debian.org/HelpDebian/Start> には何がかかっているか



A なぜ Debian に貢献するべきか



B Debian I






C GNU の

答えは:






A

問題5. d-iでの翻訳作業の省力化のための工夫を Christian Perrier が実施した。それはメッセージを使われかたの優先度によって分割するものだったが、何分割にしたか

-  A 5
-  B 4
-  C 3



問題5. d-iでの翻訳作業の省力化のための工夫を Christian Perrier が実施した。それはメッセージを使われかたの優先度によって分割するものだったが、何分割にしたか

-  A 5
-  B 4
-  C 3




答えは:



A

問題6. Fonts task force

(<http://wiki.debian.org/Fonts>) は週次で何を確認するスクリプトを用意したか？

-  A Debian パッケージとしてはたしてどれだけのフォントファイルが存在するのかを棚卸する
-  B いかかわしい形のフォントを検出する
-  C うまく表示できないフォントを検出する

問題6. Fonts task force

(<http://wiki.debian.org/Fonts>) は週次で何を確認するスクリプトを用意したか？



A Debian パッケージとしては
たしてどれだけのフォントフ
ファイルが存在しているかを知
る



B いかか
を検出する






C うまく
トを検出する

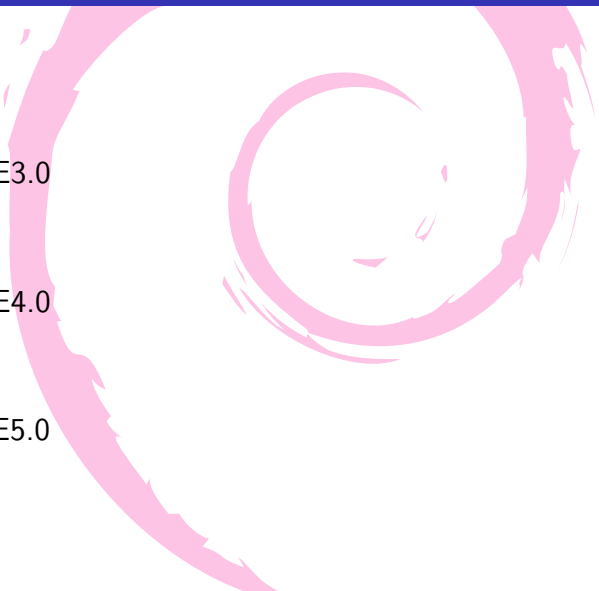


答えは:




A

問題7. Lenny リリースに標準として含まれる予定のKDEはどれか

-  A KDE3.0
-  B KDE4.0
-  C KDE5.0



問題7. Lenny リリースに標準として含まれる予定のKDEはどれか




-  A KDE3.0
-  B KDE4.0
-  C KDE5.0

答えは:






A

問題8. Debian-i18n meeting で main,contrib,non-free/i18n/Translation-* に対し て何がおきたか

-  A Grisu が悟りを開いた
-  B AUTOBYHAND の仕組み
を利用するようになった
-  C あきらめて変更しないこと
にした

問題8. Debian-i18n meeting で main,contrib,non-free/i18n/Translation-* に対し て何がおきたか




答えは:

-  A Grisu が悟りを開いた
-  B AUTOE
を利用する
-  C あきらめ
にした






B

問題9. FOSS.inの開催期間中にDDになったインド人は何人目のインド人DDか？

-  A 1
-  B 2
-  C 4



問題9. FOSS.inの開催期間中にDDになったインド人は何人目のインド人DDか？




-  A 1
-  B 2
-  C 4

答えは:



C

問題 10. insserv では何が実現できるか？

-  A 自動でデータベースに insert してくれる
-  B 依存関係から init スクリプトの実行順序を計算して実行
-  C ネームサービスの提供

問題 10. insserv では何が実現できるか？



A 自動でデータベースに insert してくれる



B 依存関係
トの実行順



C ネーム空間



答えは:

B

A large, stylized pink circular graphic element, resembling a brushstroke or a swirl, is positioned on the right side of the page, partially overlapping the text.

事前課題の 紹介

事前課題問題

- ① 「こんな Debian パッケージを作成してみました・作成してみようとしてみたらここまでしかできませんでした」

2008 年の Debian 勉強会は Debian パッケージ作成について連続して企画をする予定です。現在どれくらいできるものなのか、どこらへんでつまっているのかを教えてください。(200-800 文字)

- ② 「2008 年の Debian の目玉を大胆に予想する」
2008 年にどういう事が Debian の目玉になるのか、大胆に予想してみてください。(200-800 文字)

「こんな Debian パッケージを作成してみました・作成してみようとしてみたらここまでしかできませんでした」私は、たまに必要があると野良パッケージを作っています。野良パッケージを作る理由としては、

- ① 使いたいアプリ/機能が公式パッケージにないから
- ② 複数マシンにインストールしたり、環境再構築時に楽だから
- ③ 新規パッケージは statble に入らないから
- ④ その他

といったところです。基本的には私のレベルは `dh_make` して `s` で作るだけのレベルです。その程度ですが、某 LUG で入門として話してみたことがあります。deb 作成は rpm 作成と比べて作業や決まり事が多くて少し面倒とおもいます。が、仕事で作っていた rpm に比べて、まだ deb を作り慣れていないせいも多いと思います。

「2008年のDebianの目玉を大胆に予想する」lennyリリース！というのは皆書きそうなので... 某社からDebianプリインストール機が出る... とかあると面白いですね（笑）。

こんな Debian パッケージを作成してみました
仕事でも趣味でも必要があれば自分でもパッケージングを
やってます。debian ディレクトリのテンプレートは、perl なら
dh-make-perl を、そうでなければ dh_make を使っています。
ビルドは debuild で lintian でチェックしています。perl
モジュールの Net::SSH::Perl のパッケージングを仕事でやっ
たときが一番大変でした。(ssh を perl で実装してあるため、
大量の依存モジュールが) C のヘッダの型情報と動的ライブ
ラリを使って C の関数を Scheme の処理系である gauche から
呼び出す c-wrapper ライブラリとか、C のソースコードを
解析して Objective Caml の抽象木にしてくれる cil ライブラ
リとかを趣味でパッケージングしました。

小ネタ: 2008 年の **Debian** の目玉を大胆に予想する
experimental にはもう有るようですが perl-5.10 あたりで
しょうか。ぜんぜん大胆じゃない。python と perl が両方と
も parrot の上に乗るとか。

「こんな **Debian** パッケージを作成してみました・作成してみようとしてみたらここまでしかできませんでした」ターボールから野良ビルドする時は、必ずパッケージにすることにしています。ただ、パッケージを2つ以上に分割する方法は、見様見まねで、よくわかっていません。あと、新しいポリシー (3.7.3) の変更点もいまいち理解できていないと思います。

「**2008 年の Debian の目玉を大胆に予想する**」
lenny フリーズ！リリースは再来年に持ち越し！

「こんな Debian パッケージを作成してみました・作成してみようとしてみたらここまでしかできませんでした」

- JD という 2ch ブラウザをパッケージに。パッケージとしてはあまりいじらなくても良い状態になっています。upstream が活発なのでそれに併せて更新予定です。
- eclipse-nls-sdk は、upstream が音沙汰無いのでどうしたものか。分割方法を考えた方がいいのが課題。とりあえず lintian の warning だけ何とかします。そう言えば今思い出しましたが、Pleiades をパッケージ化できないかを upstream な人に KOF で尋ねられていたのでした。すっかり忘れてた。これも検討しないといかんですね。
- ccspatch (linux-patch-tomoyo)、ccstools (tomoyo-ccstools) については一段落しました。backports を作って、upstream に利用を明示する必要があります。

山根

- mirmon というパッケージを作って ITP しましたが、mentors でスポンサーしてあげるよといった DD がそれっきり音沙汰無いのでどうしたものかと思っています。delel@jp で聞くべきかも。
- フォントパッケージを増やしています。ttf-vlgothic、ttf-konatu、ttf-kiloji が今まで accepted。次は ttf-togoshi-gothic、ttf-ume が手元では出来ているので、細かな所を upstream に確認ののち、ITP → upload 予定。確認出来ているフリーなフォントはどんどん追加予定です。
- sylph-searcher をパッケージにしたいな、と思ったのですが、libsylph を作らないといけないのですね。library package は今のところ作って無いのでどうしたものか。

すでにパッケージを upload した人は、lintian.debian.org で自分のパッケージの warning を確認してどんどん潰すべきですな

「こんな **Debian** パッケージを作成してみました・作成してみようとしてみたらここまでしかできませんでした」
年末年始でようやく一部のサーバを Etch 化し始めました。
ついでに自宅環境で使っているスクリプトの配布には、
Debian パッケージにしてみようか、と思ったわけですね。
で、じゃあスクリプトだけをパッケージ化するのはどうやるのかと、調べようとする前に、scp で転送してしまい、やらずじまいでした…。orz 何かのついでに始めてみようというのは、ダメですね…。今度の日曜（今月の Debian 勉強会の翌日）ちゃんと時間を割いてやります。

「**2008** 年の **Debian** の目玉を大胆に予想する」
きっと誰かが Wii も Debian にして、Wii Balance Board で APT シェルを自由自在に操っているに違いない。w

こんな Debian パッケージを作成してみました作成してみようとしてみたらここまでしかできませんでした
最近作った Debian パッケージは

- Macbook の LED を Ethernet のアクセスに合わせて点灯させるソフトウェア
- u-boot 用のイメージを作成するソフトウェア

です。

両方とも Debian パッケージ化を行い、手元で使っています。
パッケージ化がむずかしいソフトウェアではないので、特にハマることはありませんでした。

なので、いじられるところは特にありません。

以前、Ruby on Rails で Oracle を使うために ruby-oci8 を install しようとしたら deb package が無かったので、package を作ってみようとした事がありました。

<http://www.debian.org/doc/manuals/maint-guide/index.ja.html#contents> 辺り（古い？）を斜め読みしながらやったのですが、

```
$ dpkg-buildpackage -r fakeroot
```

した所で、何故か /usr/local 配下に install されてしまいました。その時は、普通に install したのと同じ結果になっただけなので、まあいっか、とゆ一事で終わりにしてしまっただけなのですが、良い機会なので少し調べてみました。

```
$ tar xvfz ruby-oci8-1.0.0.tar.gz
$ cd ruby-oci8-1.0.0
$ dh_make -e mail_address -f ruby-oci8-1.0.0.tar.gz
$ dpkg-buildpackage -r fakeroot
(snip)
dh_installdirs
# Add here commands to install the package into debian/tmp
/usr/bin/make DESTDIR=/home/noriaki/tmp/deb/ruby-oci8-1.0.0/debian/tmp install
make[1]: ディレクトリ '/home/noriaki/tmp/deb/ruby-oci8-1.0.0' に入ります
ruby setup.rb install
---> lib
mkdir -p /usr/local/lib/site_ruby/1.8/
install oci8.rb /usr/local/lib/site_ruby/1.8/
Permission denied - /usr/local/lib/site_ruby/1.8/oci8.rb
Try 'ruby setup.rb --help' for detailed usage.
```

ということで、いまさら Makefile を眺めてみた所、そもそも DESTDIR がなく、install 先が簡単に変更出来ない感じでした。Makefile から呼び出している setup.rb という script を読んで、Makefile を書き換えないとダメっぽいです。今回は時間がなく、ここまでで断念しました。そもそも、package の作り方はまだ良く分かっていない所が多いので、今日、ばっちり勉強して帰って、再度 try したいと思います。

「こんな Debian パッケージを作成してみました・作成してみようとしてみたらここまでしかできませんでした」
作ってみたパッケージは GNU hello です。debian/rules ファイルを雛形のまま編集せずに debuild が通ったので実力は全くついていませんが、dh_make などの基本的なパッケージ作成コマンドの使い方は概ね把握しました。これからパッケージ作成で学びたいことは、debian/rules ファイルのスタイル・書き方についてと gpg 署名です。

「2008 年の Debian の目玉を大胆に予想する」初心者なので、2007 年の目玉がなんだったのかさえ知りません:-)

「こんな Debian パッケージを作成してみました・作成してみようとしてみたらここまでしかできませんでした」
Debian パッケージは作成した事が無いです。以前作ったものの自体をパッケージ化したほうがいいか？と考えたときに即座に必要なないと判断したのでパッケージ化するネタが無いです。こんなのを作りたいなぁと妄想だけはあるんですけどね。

こんな Debian パッケージを作成してみました
Debian パッケージになっているソフトウェアは数多くあるし、非公式のパッケージを作っている人も多いので自分でパッケージングする必要に迫られることは少ないのだが、今までパッケージングしてみた数少ない例としては以下のものが挙げられる。

* g nubg

gnubg は公式パッケージになっているが、公式パッケージが全然更新されていなかった時期に CVS 版をビルドしてパッケージングしたことはあった。autoconf を使用したものであったのでインストール可能なレベルのパッケージを作るのは比較的容易だった。

* gtkipmsg

IP Messenger クライアントとしては、xipmsg は公式パッケージとして存在するが、GTK 版クライアントの gtkipmsg のパッケージは存在しなかったのでパッケージングしてみた。これも autoconf 対応だったのでインストール可能なパッケージは比較的容易に作成できた。autoconf モノは比較的容易にパッケージングできることがわかった。が、autoconf を使っていないものについてはまだ経験がない。

「こんな Debian パッケージを作成してみました・作成してみようとしてみたらここまでしかできませんでした」

作ったパッケージ：

仕事である編集業や、趣味である日曜プログラミングでの必要性や興味から、次の野良パッケージを作って使っています。

`vfddata-otf-ptex` [http:](http://psitau.at.infoseek.co.jp/otf.html)

[//psitau.at.infoseek.co.jp/otf.html](http://psitau.at.infoseek.co.jp/otf.html) OTF は、OpenType フォントを pTeX で使うための TeX マクロパッケージおよびフォントデータです。OTF 版のヒラギノやモリサワを Debian で使いたかったので作りました。

`ideotype` <http://ideotype.sourceforge.net/>

IdeoType は、商業出版の現場で編集制作を支援するためのツールです。XHTML 形式の原稿を本 (PDF) に変換します。

これからパッケージ化しようと考えているもの：

libxsl-ruby <https://rubyforge.org/projects/libxsl/>
libxslt を Ruby で使うためのライブラリです（1文字違いの libxslt-ruby とは異なる）。Ruby の拡張ライブラリを Debian パッケージにする際の作法がよく分からず、まだ眺めている程度です。

c-wrapper <http://homepage.mac.com/naoki.koguro/prog/c-wrapper/index-j.html> Scheme インタプリタ Gauche から、C/Objective-C で書かれたライブラリを利用できる仕組み（FFI）です。

知りたいこと：

他のパッケージ形式との間での変換の仕方（例えば RubyGems の Gem から dpkg 形式への変換や、dpkg 形式から RPM や MacPorts への変換など）。Gem の場合、ほぼ自動での変換が可能らしいことは分かったのですが、ruby-pkg-tools の使い方が分からず挫折中です。

パッケージ化されてほしいその他のソフトウェア：

[rushcheck](#) Ruby 用のテストツール

[rspec](#) Ruby 用のテストツール

[cruisecontrol.rb](#) continuous integration サーバ

[pdumpfs-rsync](#) rsync 越しの pdumpfs

[pdumpfs-clean](#) pdumpfs によるバックアップデータの整理・
削除

[color-moccur.el](#), [moccur-edit.el](#) Emacs で複数バッファを一括
して検索・編集



2008
計画

年

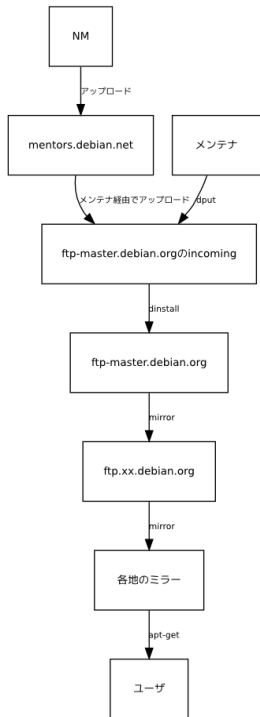


2008年計画

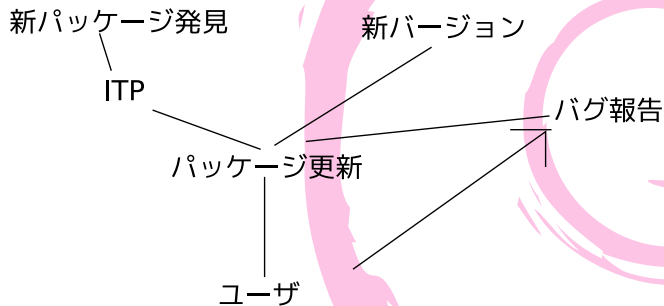
- ① 新年会「気合を入れる」
- ② Open Source Conference Tokyo (3/1)
- ③ データだけのパッケージを作成してみる、ライセンスの考え方 (David)
- ④ バイナリーつのパッケージを作成してみる (吉田@板橋)
バージョン管理ツールを使い Debian パッケージを管理する (git)
アップストリームの扱い (svn/git/cvs)(岩松 信洋さん)
- ⑤ バイナリの分けたパッケージの作成。(前田さん)
バイナリの分け方の考え方、アップグレードなどの運用とか。
- ⑥ パッケージ作成 (dpatch/debhelper で作成するパッケージ)(小林儀匡さん)
man の書き方 (roff or docbook)(でんさん)
- ⑦ パッケージ作成 (kernel patch、kernel module)、Debconf 発表練習
- ⑧ Debconf アルゼンチン、共有ライブラリパッケージ作成
- ⑨ Open Source Conference Tokyo/Fall、デーモン系のパッケージの作成、latex、emacs-lisp、フォントパッケージ
- ⑩ パッケージの cross-compile の方法、amd64 上で i386 のパッケージとか、OSC-Fall 報告会、Debconf 報告会
- ⑪ 国際化 po-debconf / po 化 / DDTP
- ⑫ 忘年会



Debian パ
ツケ ー ジ
作成



開発者のロジックの流れ





アップ スム
トリー ム
取得



パッケージ
作成

BTS の対応

- `reportbug-ng`
- `reportbug`
- `dpkg-dev-el`



- devscripts \mathcal{D} dch
- dpkg-dev-el \mathcal{D} debian-changelog.el
- vim \mathcal{D} debchangelog


- `debuild`
- `lintian` / `linda`
- `debc` / `debdiff`
- `debi`
- `pbuilder`





アップロード

レポジトリ作成

- dpkg-scanpackages / dpkg-scansources
 - apt-ftparchive
 - mini-dinstall
 - apt-move
 - debarchiver
 - reprepro
 - dak
- 

宴会場所

- 宴会場所
本日の宴会は「???'」です。参加者は1Fに集合し、全員で移動しましょう。
- 片付け
部屋を片付けるのにご協力ください。