東京エリア デビアン 勉強会

Debian勉強会幹事 上川純一

2008 年 1 月 19 日

1 Introduction

上川 純一

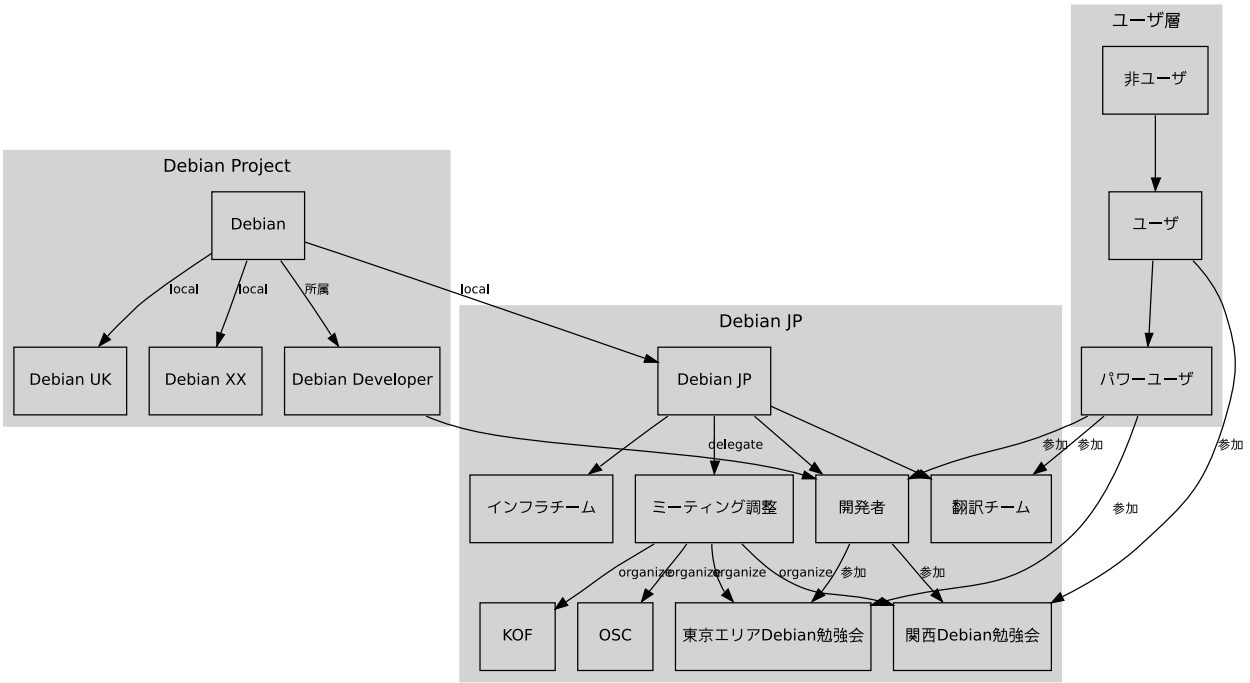
今月の Debian 勉強会へようこそ。これから Debian の世界にあしを踏み入れるという方も、すでにどっぷりとつかっているという方も、月に一回 Debian について語りませんか？

Debian 勉強会の目的は下記です。

- Debian Developer (開発者) の育成。
- 日本語での「開発に関する情報」を整理してまとめ、アップデートする。
- 場 の提供。
  - 普段ばらばらな場所にいる人々が face-to-face で出会える場を提供する。
  - Debian のためになることを語る場を提供する。
  - Debian について語る場を提供する。

Debian の勉強会ということで究極的には参加者全員が Debian Package をがりがりと作るスーパーハッカーになった姿を妄想しています。

情報の共有・活用を通して Debian の今後の能動的な展開への土台として、「場」としての空間を提供するのが目的です。



デビアン勉強会

目次

2 事前課題

上川 純一

今回の事前課題は以下です。

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)

この課題に対して提出いただいた内容は以下です。

2.1 吉田@板橋

2.1.1 「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」

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。

2.1.2 「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）。

2.2 日比野 啓

2.2.1 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。

2.2.2 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。

2.3 山本

2.3.1 「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」

タール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。

2.3.2 「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！

2.4 山根

2.4.1 「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」

- 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そうです。
- 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

2.5 前田 耕平

2.5.1 「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」

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。

2.5.2 「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

2.6 岩松信洋

2.6.1 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

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。

2.7 Noriaki Sato

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、とゆー事で終わりにしてしまったのですが、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。

2.8 高橋 昭之

「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:-(

2.9 Aya Komuro

「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。

2.10 橋本 徹

2.10.1 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。

* gnubg

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 を使用したものだ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。

2.11 森田尚

2.11.1 「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」

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。

vfdata-otf-ptex `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 用のテスティングツール

rspec　Ruby 用のテスティングツール

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

3 Debian Trivia Quiz

上川 純一

ところで、みなさん Debian 関連の話題においついていますか？Debian 関連の話題はメーリングリストをよんでいると追跡できます。ただよんでいるだけでははりあいがないので、理解度のテストをします。特に一人だけでは意味がわからないところもあるかも知れません。みんなで一緒に読んでみましょう。

今回の出題範囲は `debian-devel-announce@lists.debian.org` に投稿された内容からです。

3.1 問題

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 2. lintian は Debian パッケージのよくある間違いを検出してくれるツールである。 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 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 4. http://wiki.debian.org/HelpDebian/Start には何がかかれているか

☐ 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 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 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 いかが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 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 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 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 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

4 最近の Debian 関連のミーティング報告

上川 純一

4.1 東京エリア Debian 勉強会 35 回目報告

今回の参加者は山本 浩之さん、石原怜美さん、あけどさん、吉田@板橋さん、でんさん、前田耕平さん、小林さん、小室文さん、本庄さん、キタハラさん、上川の 11 人でした。

まず、クイズを今回も実施しました。今回も、debian-devel-announce の内容から出題しました。ニュース Wiki が開始したのでそこからの出題が中心になりました。2 問くらいで全滅するという正答率の低さですが、その度に勝ち残った参加者に景品をさしあげました。

最近の話題として出たのが、qmail や djbdns がオープンになったのが出ました。これはインパクトでかいです。qmail を仕方なく使っている人たちや djbdns が軽いから愛用しているという人たちは修正が出ない現状に課題を感じていた部分が多かったと思いますが、それが解消されることに期待です。Debian 関連の最近の話題で一番大きなものとしては、Debian Maintainers 制度の発足でしょうか。Debian Project のメンバーの要件を再定義してしまい、大幅に活性化されることにつながるのではないかと睨んでいます。

今回はいきなり Debian 勉強会の資料の作り方について語りました。LaTeX と whizzytex と emacs と yatex と git の使い方について簡単に説明しました。これでみんな資料がかけるようになったと思います。LaTeX については LaTeX の用意されている内容を使うことができるとよいのだが、綺麗に整形しようとすると複雑になりがちなのでスタイルファイルでうまく定義できるようにするのがポイントだね、という議論をしました。LaTeX は大半の人が使えるので、Debian 勉強会の資料も Debian 勉強会用に定義したコマンドを一部説明さえすればみんな作成できるようだということがわかりました。git と latex を利用しているワークフローについては、共同作業の際にマージするのや差分を確認するのに便利なので、普段 Word で困っている人などにおすすめです。ハードルは上がりますが、Word でせっかくフォームを利用して自動化しているのに直接編集されて困ったという経験がある方、latex で頑張った方がよい結果がでるかもしれません。

忘年会らしく、2007 年をふりかえってみました。まず Debian JP の位置づけ、および東京エリア Debian 勉強会の存在を整理しました。ユーザは Debian JP の「会員」ではないという点については違和感があったようですが、debian-users などに参加するのとは別に選挙などの運営に関わるのが「会員」であるという点で説明しました。Debian JP の各種の会議の頻度や、伝達の流れについても説明し、ディスカッションし、大筋こんなものだろう、ということが分かりました。「図は何で書いているのか」という質問がありましたが、 GraphViz の dot です。日本語を正しく出すのにはコツがあります。

Debian JP の背景を見たところで、Debian JP の目的について議論しました。最初に白紙だったところに、全員で議論して記入しました。どうやら目的はこんなもんだったようです。

- Debian Developer (開発者) の育成。
- 日本語での「開発に関する情報」を整理してまとめ、アップデートする。

- 場の提供。
  - 普段ばらばらな場所にいる人々が face-to-face で出会える場を提供する。
  - Debian のためになることを語る場を提供する。
  - Debian について語る場を提供する。

Debian 勉強会の参加者の推移と実施内容について確認しました。みたところ、平均の参加人数は１６人程度から１８人程度になっているだけなので大きく人が増えているわけではないことがわかりました。また、実施内容を見ると、2005 年は debhelper 連載、2006 年については policy 連載があったのでテーマ感があったのですが、2007 年の開催内容は開発者になるための情報という観点からは一貫性がなく情報量も少なかったのではないかという傾向が見えました。

Debian 勉強会に影響のありそうな過去のイベントと将来のイベントを出してみようということで、トレンドをみんなで議論して出し合いました。携帯世代の台頭、高速移動体通信の普及ということで PC ベースの Debian の普及には逆風が吹いており、64bit コンピューティング環境の普及の観点では今後数年間に大きく普及するだろう、というようなトレンドが出てきました。

SWOT 分析を行いました。できたこと・できなかったこと・チャンスとなるもの・脅威となるものを出し合い、それらに対しての施策を考えてみました。各種オープンソース化の動きと仮想化のトレンドが直近では大きいためそれを受けた施策などが必要だろう、Debian の人材が流出していくのではないか、人を育てる必要がある、というようなものが出てきました。

その後、2005 年、2006 年の最初のように勉強会にテーマ感をもたせてみよう、ということで、パッケージの作成ができるための施策をまずうつ必要がある、という認識の元、各種のパッケージを作成する流れを順番にやっていって一年間やってみよう、という計画をたてました。

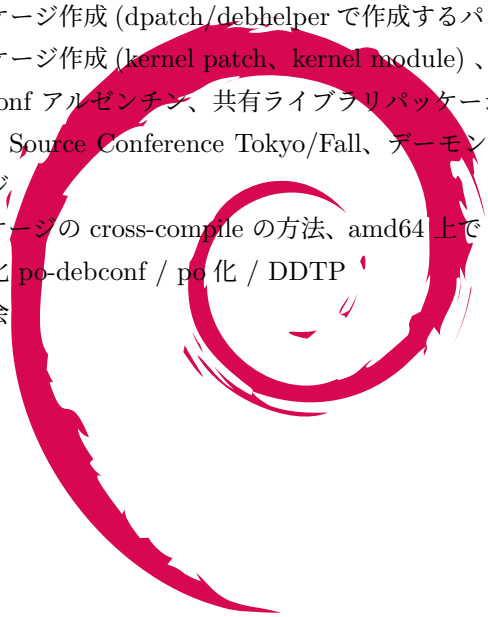
今回は宴会は「時の居酒屋 刻　荻窪店」にて開催しました。

5 2008 年度 Debian 勉強会企画

上川 純一

2007 年 12 月に実施した 35 回 Debian 勉強会で、一年分の実施したい内容を出してみました。そこで策定した計画は以下です。

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
4. バイナリ一つのパッケージを作成してみる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)
5. バイナリの分けたパッケージの作成。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) man の書き方 (roff or docbook)
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



6 Debian Package 管理の流れ

上川 純一

2008 年の Debian 勉強会では対応する種類別に Debian パッケージの作成の個別の内容を検討していくことになります。Debian パッケージの作成の技術的詳細にはいる前に、まず基本に立ち返って全体的な作業の流れを確認してみましょう。

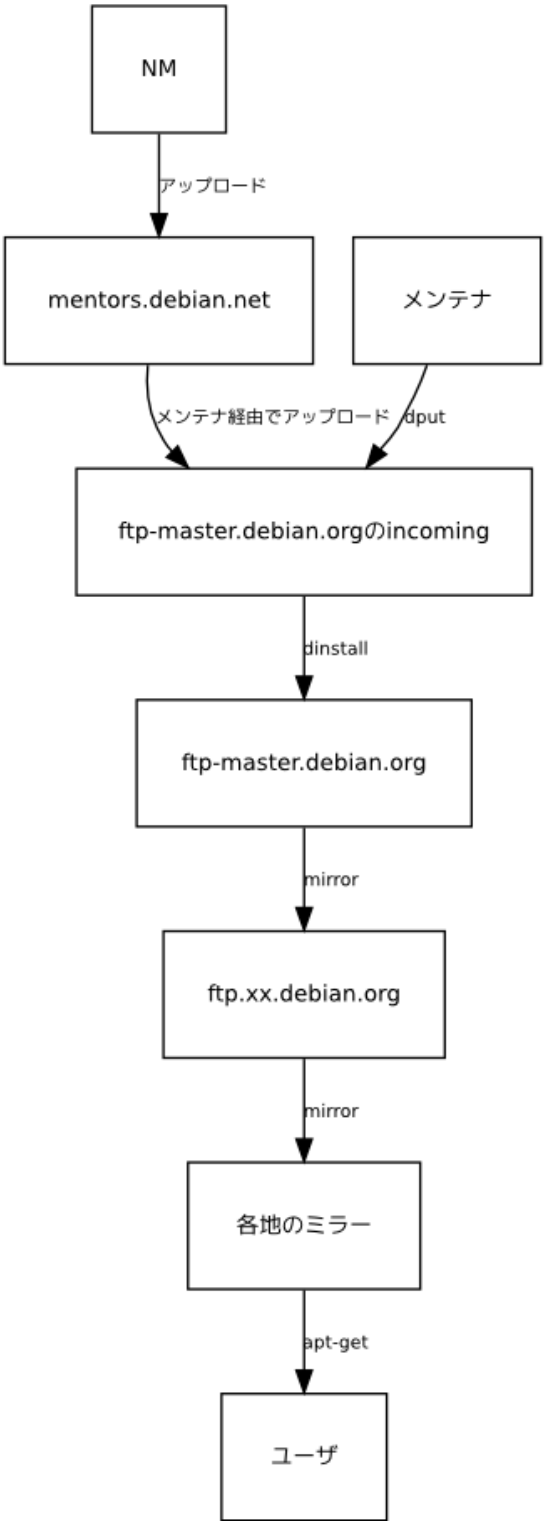
6.1 開発者からユーザにパッケージが到達するまで

Debian Developer はどこからかソースパッケージを取得してきて、パッケージをアップロードします。

公式パッケージの場合は、開発者が dput コマンドで ftp-master.debian.org にパッケージをアップロードすると da-katie が処理し、一日二回のバッチのタイミングで ftp.debian.org ミラーネットワークにパッケージが配信されます。日本のユーザであれば、 cdn.debian.or.jp を利用し、apt-get update / aptitude update をしたタイミングで新しいパッケージが取得できるようになります。

Debian Developer でない場合には、自分で dput を実行して直接 ftp-master.debian.org にアップロードするということはできません。Debian Developer の他の誰かに依頼して実施します。その際にこうしなければならないという定型の方法はありませんが、最近は Debian Mentors という仕組みが普及しています。`http://mentors.debian.net/` にパッケージをアップロードし、Debian Mentors の ML [*1]などで聞いてみるとよいでしょう。

[*1] `http://lists.debian.org/debian-mentors/`



6.2 開発者のパッケージングの契機

開発者がパッケージを作成するタイミングとはどういうものがあるでしょうか。簡単にいうと三つあります:

- Debian に存在しない全く新しいパッケージ (ITP からの一連の流れをふむ)
- Debian にすでに存在しているパッケージの新しいアップストリームバージョンがリリースされた
- Debian にすでに存在しているパッケージにユーザがバグ報告をし、BTS に登録されたバグ報告の内容に対応するにはパッケージの修正が必要になった

- 依存関係が更新されたから更新

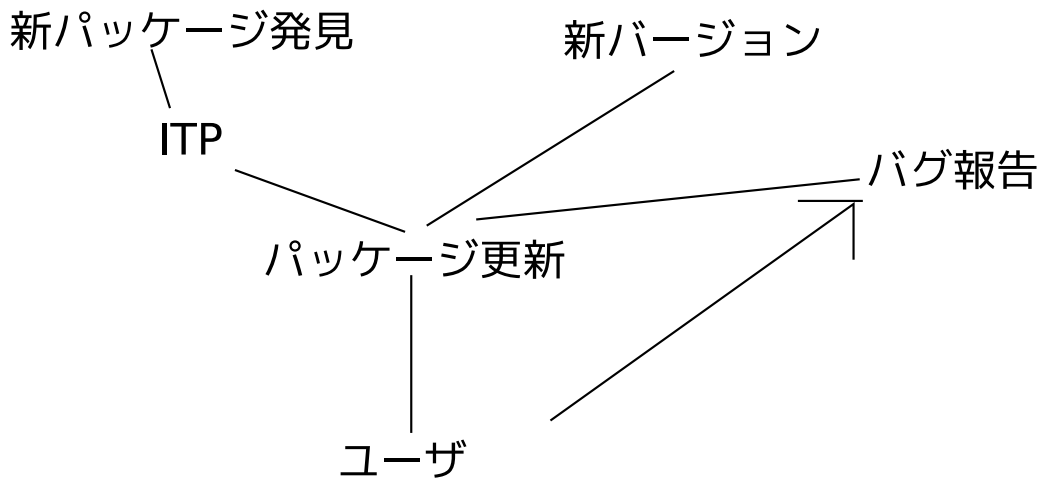


図 1　開発者のロジックの流れ

6.3　作業の流れ

6.3.1　アップストリームの取得

新しいソースパッケージを取得します。tar.gz で公開されていればそのまま使えます。tar.bz2 であったり、よくわからないアーカイブフォーマットだったり、git レポジトリでしか公開されていなかったりするので、それなりに加工します。

最初は適当に取得するわけですが、二回目移行はスクリプトで自動化して取得するようにします。watch という仕組みがあり、 uscan / uupdate コマンドで新しいバージョンをダウンロードしてきて Debian パッケージ用のパッチを適用することができます。

```
version=3
http://ftp.imendio.com/pub/imendio/giggle/src/giggle-(.*)\.tar\.gz
```

図 2　giggle パッケージの debian/watch ファイル

6.3.2　パッケージング

二度めからは差分の適用などで対応できるのですが、最初の新規パッケージの場合は ITP などの処理を必要とします。また、新規にパッケージを作成するので、なんらかのテンプレートパッケージから debian/ ディレクトリを取得してきて修正するか、 dh_make コマンドを利用してテンプレートを作成します。

この段階で debhelper を利用するのか、 cdbs を利用するのかの選択を迫られます。cdbs は debhelper より新しい仕組みですが、2003 年の登場から 5 年もたっているのでそろそろこなれています。簡単なパッケージであれば cdbs を利用すればよいでしょう。現状の cdbs のフレームワークで不具合があるような場合であれば debhelper で詳細に設定するのがよいでしょう。

```
$ dh_make

Type of package: single binary, multiple binary, library, kernel module or cdbs?
 [s/m/l/k/b] b

Maintainer name : Junichi Uekawa
Email-Address   : dancer@debian.org
Date            : Fri, 18 Jan 2008 22:22:51 +0900
Package Name    : tttt
Version         : 2.2
License         : blank
Type of Package : cdbs
Hit <enter> to confirm:
Could not find tttt_2.2.orig.tar.gz
Either specify an alternate file to use with -f,
or add --createorig to create one.
[22:22:55]coreduo:tttt-2.2>
```

6.3.3 BTS との対応

BTS に登録されているバグ報告に対してメールで対応したりします。報告されている不具合をちゃんと修正してみたりします。

BTS は閲覧は Web でできますが、制御は Web ではできません。BTS はメールで命令を直接うって制御できます。メールコマンドはすぐに忘れてしまうので、`/usr/share/doc/debian/bug-*` あたりを参照しながら作業します。BTS の操作に便利なツールがいくつかあるので、そちらを利用することをおすすめします。

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

6.3.4 ChangeLog の管理

パッケージの新しいバージョンを作成する場合には、debian/changelog に対応内容を記述します。ここで活用できるツールには次があります。

- devscripts の dch
- dpkg-dev-el の debian-changelog.el
- vim の debchangelog

バージョン番号を自動で増加させてくれたり、いろいろなエントリーの補完などをしてくれます。BTS を参照して修正されたバグのタイトルから ChangeLog のエントリを生成してくれたりもします。

```
pbuilder (0.177) unstable; urgency=low

  [ Loic Minier ]
  * Run apt-get autoremove after upgrade.

  [ Junichi Uekawa ]
  * python-apt/gdebi based pbuilder-satisfydepends-gdebi (closes:
    #453388)
  * Fix devpts mount permissions (closes: #453862)
  * Document pbuilder-satisfydepends-gebi in manpage

 -- Junichi Uekawa <dancer@debian.org>  Wed, 26 Dec 2007 20:53:24 +0900
```

6.3.5 ビルド・テスト

debian/以下のファイルを微調整して、パッケージをビルドします。dpkg-buildpackage を直接呼び出してもよいですが、devscripts パッケージの debuild コマンドを利用すればよいでしょう。

lintian / linda で生成されたパッケージにあきらかな間違いがないことを確認します。

debc, debdiff で生成されたパッケージにあきらかな間違いや意図しない大きな変化がないことを確認します。

debi でパッケージをインストールして正常に動作することを確認します。

pbuilder でビルドしてみて問題ないことを確認します。

pbuilder のテストスクリプト pbuilder-test を利用してインストール・アンインストール・アップグレードの試験をしてみるとさらによいでしょう。

6.3.6 アップロード

署名してアップロードします。dpkg-buildpackage コマンドを利用してパッケージをビルドすると自動で gpg で署名することになります。あとで署名しなおす場合には、 debsign を利用します。

アップロードは dput コマンドを利用します。

6.4 非公式パッケージレポジトリの作成方法

テスト用に非公式のパッケージレポジトリを準備すると便利です。

Packages.gz ファイルなどを生成して HTTP 経由でアクセスできるようにしておくと、ユーザが /etc/apt/sources.list に追記することで apt を利用してパッケージが取得できるようになります。

- dpkg-scanpackages / dpkg-scansources を直接利用する方法。
- apt-ftparchive を利用する方法。
- mini-dinstall を利用する方法。
- apt-move を利用する方法。
- debarchiver
- reprepro
- dak

2007 年 9 月号で apt-ftparchive が紹介されていたので再掲します。

6.4.1 apt-ftparchive

Sources.gz / Packages.gz などのパッケージ情報用ファイルを作成するためのツール。自分で作ったパッケージを apt-line として公開したいときに使います。

```
% apt-ftparchive packages . | gzip -9 > Packages.gz
% apt-ftparchive sources . | gzip -9 > Sources.gz
% apt-ftparchive release . > Release
```

6.4.2 apt-sortpkgs

Packages ファイル および Sources ファイルをソートします。apt-ftparchive で作成したものはソートされていなかったりするので、アルファベット順にソートするときに使います。

```
% apt-sortpkgs Packages > Packages.sort
```

6.5 参考文献

必読の文献として、以下があります。一通り読んだ後は、パッケージとしてインストールしておき、手元で検索できるようにしておくと便利です。sid を利用しているのであれば、パッケージとしてインストールしておくと最新版に勝手に更新されるため、便利です。

内容に誤記などがあれば、ぜひ BTS にバグ報告を登録してください。

- debian-policy: Debian Policy Manual。パッケージの準拠すべきルールが文書化されています。
- developers-reference: Developers Reference。
- doc-debian: Debian Project Documentation。 BTS のメールインタフェースなどの文書がある。
- maint-guide: Debian New Maintainer's guide。New Maintainer として知るべきマナーのようなものが文書

化されています。

Debian 勉強会資料

2008 年 1 月 19 日　　初版第 1 刷発行
東京エリア Debian 勉強会 （編集・印刷・発行）
