



## 第3回 東京エリア Debian 勉強会 事前資料\*

非営利個人 上川純一†

2005年4月9日

---

\* 機密レベル public: 一般開示可能

† Debian Project Official Developer

## 目次

1	Introduction To Debian 勉強会	2
1.1	講師紹介 . . . . .	2
1.2	事前課題紹介 . . . . .	2
2	Debian Weekly News trivia quiz	5
2.1	2005 年 11 号 . . . . .	5
2.2	2005 年 12 号 . . . . .	6
2.3	2005 年 13 号 . . . . .	7
2.4	2005 年 14 号 . . . . .	8
3	最近の Debian 関連のミーティング報告	10
3.1	東京エリア Debian 勉強会 2 回目報告 . . . . .	10
4	debhelper 論 その 3	11
4.1	debhelper の実行フロー . . . . .	11
5	個人提案課題	15
6	グループ提案課題	16
7	Keysigning Party	17
8	次回	18

## 1 Introduction To Debian 勉強会

上川純一



3 回目の Debian 勉強会へようこそ．これから Debian のあやしい世界に入るという方も，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？

目的として下記の二つを考えています．

- メールではよみとれない，もしくはよみとってられないような情報を情報共有する場をつくる
- まとまっていない Debian を利用する際の情報をまとめて，ある程度の塊として出してみる

また，東京には Linux の勉強会はたくさんありますので，Debian に限定した勉強会にします．Linux の基本的な利用方法などが知りたい方は，他でがんばってください．Debian の勉強会ということで究極的には参加者全員が Debian Package をがりがりを作りながらスーパーハッカーになれるような姿を妄想しています．

Debian をこれからどうするという能動的な展開への土台としての空間を提供し，情報の共有をしたい，というのが目的です．次回は違うこと言ってるかもしれませんが，御容赦を．

### 1.1 講師紹介

- 山根さん Debian の翻訳関連についてはマスターだそうです
- 上川純一 Debian Developer です．元超並列計算機やっていて，今は音楽関係とか，気づいたら canna とか．あと，pbuilder とか，libpkg-guide とか，Debian の品質向上目指してます．

### 1.2 事前課題紹介

3 回目の事前課題は「sarge がリリースするまでに日本語化しておきたい Debian のここらへん」というタイトルで 200-800 文字程度の文章を書いてください．というものでした．その課題に対して下記の内容を提出いただきました．

#### 1.2.1 Eto Yasuo

最低限 aptitude やら インストーラに含まれるもの。Debian Specific なものについての po、man、マニュアルについてはやりたいと思っていましたが、能力が不足していることが解り、英語や日本語の能力を上げるための努力をする気も時間もないので、自分は一切やらないことにしました。自分が手を動かさないことをぐちぐち言ってもしょうがないので、「sarge がリリースするまでに日本語化しておきたい Debian のここ」はなにもありません。

### 1.2.2 Junji Sawada

sarge のリリースまでに日本語化したいのは、パッケージ説明の日本語化である。パッケージを検索・詳細表示する場合、やはり英語で説明が書かれているよりも日本語で書かれている方が理解しやすい。これを実現するためには、まず各パッケージについて control ファイル中のパッケージ説明を日本語化した control.ja を用意する。次に Packages を作成する際には言語ごとに作成する。control.ja がないパッケージについては control を利用する。apt-get update 時にはロケールに応じて適切な Packages を取得する。これにより、利用言語に応じたパッケージ説明を見ることが可能である。問題点として、root のロケールを ja にしておかないといけないうこと、現状のパッケージングシステムを大きく変える必要があること、言語ごとの Packages を以下に効率よく作成するか (UTF-8 化して一つのファイルに含めてしまうのがよい?) 挙げられる。

### 1.2.3 中島清貴さん

はじめから日本語で作ってほしい。なので他の言語を日本語化してほしくない。日本語化というと既存の言語を日本語にするわけだから、かなりの手間だ。それであれば始めから日本語で作って日本語で使うべきだ。

英語も英語で書いて英語で読むべきだ。書けなくても英語で書かれたものは英語で読むくらいのリテラシーは必要だ。逆に日本語で作られたものは日本語で使うべきだし英語化するべきではない。

読めなければ使えないという状態に持っていけば本当に使いたければ気合で読むはずなので、気合で自分の頭で日本語化してもらうのがよい。自分もワープロが苦手だけど気合いで使っている。

日本語にも対応というスタンスでいい。言語の一つとして日本語にも対応する。もちろん当然その他の言語にも対応する。

日本語化するのがいけないとは言わない。ユーザが自分の頭で日本語に変換するべきだ。それであれば誰も手間はかからない。

第一そんな手間暇かかることやっていたら時間がかかるし生産性がない。こんなことコンピュータのエキスパートがやるとは思えないのだ。

これからは始めから日本語で作るべき。そうでなければ日本語にも対応する。もし他の言語で作られていればユーザが個人の頭で勝手に日本語化する。これでいい。

### 1.2.4 Nobuhiro Iwamatsu

- Debian Installer / マニュアル 完全日本語対応してほしいなと思います。
  - debconf の po
  - package の description 日本語化してるものとしてないものがあるのは見た目がカッコ悪い。ちょっと時間があるときにできるので時間をみてやってます。
  - 日本語化にはあまり関係ないかもしれませんが aptitude の 日本語 (マルチバイト) の文字化け
- こんなところです。

### 1.2.5 KURASHIKI Satoru

個人的に欲しいのは、SELinux の導入手順文書でしょうか。まだ公式の扱いじゃないようですが、他の distro に比べて、敷居が高いように見えます。また、d-i の FAQ、Wiki あたりも日本語になるといいかなあ

と思います。ただまあ、他の作業よりもさっさとリリースしちゃうのが Debian には一番必要なんじゃないかとか思ったりも。

#### 1.2.6 上川

日本語に訳されているけど、現状に内容がおいついていない、一部の古いマニュアルがあります。そういうマニュアルを利用すると間違った利用をしてしまう可能性があるんで、できるだけそういう状況をなくしたいですね。

翻訳の衝突とか、翻訳を採用してもらう手順とか、現在面倒なところもいくつかあると思うので、地道に翻訳するインフラを整備するのも大切じゃないかな、と考えています。

特に、リリース前の翻訳のためのテキストフリーズという概念をディストリビューション全体で採用できると理想なのですが。

## 2 Debian Weekly News trivia quiz

上川純一



ところで、Debian Weekly News (DWN) は読んでいますか？Debian 界限でおきていることについて書いている Debian Weekly News. 毎回読んでいるといろいろと分かって来ますが、一人で読んでいても、解説が少ないので、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 DWN を読んでみましょう．

漫然と読むだけではおもしろくないので、DWN の記事から出題した以下の質問にこたえてみてください．後で内容は解説します．

### 2.1 2005 年 11 号

問題 1. Debconf5 に投稿されたプロポーザルの数は

- A 22
- B 33
- C 44

問題 2. Debian logo のライセンスの変更するためになにがおきたか

- A 原作者から SPI に著作権の移管
- B 著作権法の改正
- C ロゴの作り直し

問題 3. ドキュメントや辞書などのデータに GPL ライセンスを適用することにより問題となりうるのはなにか

- A ライセンス文が大きすぎることによるデータ量の増大
- B GPL でないアプリケーションから利用する場合の検討
- C 利用するまえに Free software song を歌う必要がでてくる

問題 4. USB ストレージデバイスで GPG キーを利用する際に、VFAT 上でループバックデバイスを利用しているのはなぜか．

- A 暗号化したいから
- B ループバックでしか VFAT はマウントできない
- C GPG がループバックデバイスしか対応していない

問題 5. Debian の wanna-build サーバの SSH サーバに関して変更が発生したのはなぜか

- A コネクションキャッシングしないと負荷が高くて死んでしまうから
- B SSH をビルドすることがかっこよいから
- C make コマンドをたたく練習がしたかったから

問題 6. Joey Hess が NEW キューに関して提案した内容はなにか

- A NEW キューを無くしてしまおう
- B NEW のパッケージに関して人気投票をしてみよう
- C 新しいパッケージは作成しないことにしよう

問題 7. DPL debate はどこで開催されたか

- A Vancouver
- B メッセンジャー
- C IRC の Freenode ネットワーク上

問題 8. Steve Langasek の出した SCC の提案とはなにか

- A 多くのアーキテクチャをリリースプロセスから除外する
- B 世界に残存する m68k を廃棄する
- C etch のリリースアーキテクチャは 20 種類を目指す

問題 9. Asia Debian Miniconf で達成したのはなにか

- A debian-zh IRC チャンネルの復活
- B 全員がお腹をこわした
- C 中国から台湾へのネットワークでの接続

## 2.2 2005 年 12 号

問題 10. GPLv3 が出て来ることでもっともおそれられているのは

- A GPLv2 のみに対応するというソースコードが増えて、プロジェクトのフォークが増えること
- B ライセンス文が長文になりすぎてオープンソースプロジェクトのファイルサイズが大きくなりすぎる事
- C 誰も読めない言語で書かれていること

問題 11. Creative Commons 2.0 のライセンスについて debian-legal が出した判定結果は

- A DFSG Free です
- B Derived Work について制約が多いので、DFSG Free ではない
- C GPL と同じです

問題 12. Enrico Zini はどんなアンケートを実施したか

- A 一日の生活パターンについて
- B 人生の目標について
- C Debian の利用目的と方法について

問題 13. 300000 番目のバグレポートはどういう内容か

- A SPAM
- B セキュリティーホール
- C Description の文法的な修正

問題 14. DPL Vote システムに暗号化したメールで投票したらどう処理されるか

- A gpg で復号化し, devotee システムが処理する
- B /dev/random から投票結果を生成する
- C 暗号には現在対応していない

問題 15. autoconf を build 時に呼ぶ理由でないものはなにか

- A そこに autoconf があるから
- B ソースパッケージのサイズを削減できる
- C autoconf が変更したあとで実行してもパッケージがビルドできないという状況をさけられる

## 2.3 2005 年 13 号

問題 16. 3 月 18 日にあたらしく増えた FTP-master メンバーは誰か

- A Jeroen van Wolffelaar と Joerg Jaspert
- B James Troup と Ryan Murray
- C Anthony Towns と Branden Robinson

問題 17. 新しくできたメーリングリスト debian-dak はなにについて語る場所か

- A Katie さんについて語る場所
- B Debian のパッケージインフラソフトウェアについて語る場所
- C Anthony Towns の好きな歌手について語る場所

問題 18. libtool1.4 に build-depend しているパッケージにたいしてなにがおきるか

- A あたらしい libtool を使うように推奨
- B 即刻パッケージを削除
- C libtool を使わないように修正

問題 19. ITP されているものの放置されているパッケージに関して推奨された処理は



- A メールアドレスを調査して SPAM 送付
- B バグレポートにメールを追記して、作業を開始する
- C BTS サーバに新入して、ITP バグレポートをこっそり削除

問題 20. Description の文書でよく間違っていると Florian Zumbiehl が指摘したのは

- A 英語になっていない
- B 略語の前の 'a' が 'an' であるべき場所がある
- C ギリシャ文字で書いてある

問題 21. PHP 4 のライセンスで問題となっているのは、なにか

- A GPL が嫌いだ、と書いてある
- B 問題はない
- C 変更したものは、PHP4 という名前を利用できない

問題 22. Marcus Brinkmann が Hurd/Mach をより Hurd/L4 がよいという理由は

- A l4 のほうが Linux のエミュレーションが優れている
- B Mach の VM 管理がださい
- C l4 のほうがデバイスドライバが書きやすい

## 2.4 2005 年 14 号

問題 23. GNU Hurd のライブ CD を qemu で起動するコマンドラインはどれか

- A `qemu -cdrom hurd-live-cd.iso -boot d`
- B `./gnu-hurd`
- C `dd if=hurd-live-cd.iso of=/dev/hdc && reboot`

問題 24. SCC 提案に対して、John Goerzen の提案した内容はなにか

- A 一部のアーキテクチャはバイナリを配付せずにソースのみを配付する
- B gentoo に全員移動
- C 全員 i386 以外のアーキテクチャは使わない事にする

問題 25. 遅いアーキテクチャの対応として提案されたツールの案で、エミュレータなのはどれか

- A qemu
- B scratchbox
- C distcc

問題 26. chroot 内でデーモンの起動をしてほしくないときにはなにを使えばよいか

- A invoke-rc.d のための policy-rc.d スクリプトを作成する
- B init.d 以下のスクリプトを全部即 exit0 で終了するものに変更する
- C 起動しないように祈りの踊りをささげる

問題 27. 起動スクリプトの出力をログにとるためにはどうするか

- A 画面をみながら出力のメモをとる
- B dmesg コマンドの出力にでるのでそれを使う
- C echo 'BOOTLOGD\_ENABLE=Yes' > /etc/default/bootlogd

問題 28. testing からパッケージが削除された場合の情報はどこに流れるか

- A apt-cache show パッケージ名
- B 秘密なので教えられない
- C 今後は debian-testing-changes に流れる予定

問題 29. gluck.debian.org になにがおきたか

- A ネットワーク障害による通信の断絶
- B ディスク障害によるファイルシステムの崩壊
- C 人的災害によるシステムの破壊

### 3 最近の Debian 関連のミーティング報告

上川純一



#### 3.1 東京エリア Debian 勉強会 2 回目報告

今回は朝早くから実施しましたが、予想したよりたくさんの方が無事に到着しました。今日の参加者は合計 8 名。

最初は DWN クイズを実施しました。15 問中 14 問正解が最高で、12 問くらいまでは正解している方がいらっしゃいました。

Debian Social Contract についての話を松山さんがしました。Debian Social Contract をまじめによんでみる、という機会をもってみると、本当にこの通りにできているのか、という疑問をもつ点がありますが、その疑問をもつということも大切でしょう。upstream との関係がうまくとれているのか、フリーの定義とは何かなど、Debian の精神の基本となる文書でありながらも議論の余地の多い文書であることがわかりました。

debhelper について上川が語りました。現在 debhelper パッケージに入っている全コマンドについて一覧を概説し、それぞれのコマンドが debian/package.XXX のような名前のファイルを入力としてとりいれて、それから postinst.debhelper ファイルなどをつくり、postinst に書いてある#DEBHELPER#という部分を置き換える、ということについて説明しました。#DEBHELPER#の意味を知らない方が意外と多くておどろきです。

グループワークとして、Debian パッケージシステムの改善案について議論しました。出て来た要望としては、「現在はパッケージ名と説明文でしか検索できないので、パッケージの機能を検索したい」「dpkg でインストールするファイルを全部 dpkg で管理できるようにしてほしい」「unofficial な sources.list のエントリなどを一括管理できるシステムや GUI などがあればよいな」などでした。即実行できる内容もあれば、まだまだ難しいだろうと思われるものもありました。

その後、デニーズにて懇親会。次回はぜひ翻訳について語って頂きたいところです。あとは、Debian パッケージを debhelper を使ってビルドできるまでのフローについて解説したいと思います。

## 4 debhelper 論 その 3

Debhelper の処理の流れ .

---



### 4.1 debhelper の実行フロー

debhelper の一般的な実行フローにおいてどのファイルがどういう順番でコピーされていくのか、という部分について追跡してみます .

#### 4.1.1 debian/rules clean

debhelper としては、Debian パッケージのビルドに利用した一時ファイルを削除します .  
debian/以下のディレクトリの削除を実施します .

- *packagename* ディレクトリ
- *packagename*\*.debhelper
- files
- tmp

また、一般的な一時ファイルの削除も実施します .

- `###`
- `*~`
- `DEADJOE`
- `*.orig`
- `*.rej`
- `*.bak`
- `*.SUMS`
- `TAGS`
- `core`
- `*/*.deps/*.P`
- `autotm4te.cache`

#### 4.1.2 debian/rules build

ここでは、debhelper はほとんどなにもしません . 上流のソースコードに対して Make し、コンパイルする作業が主です .

カレントディレクトリが正しい場所か、ということを確認します .

#### 4.1.3 debian/rules install

ここでは、debhelper を利用して、インストール先のディレクトリを作成し、そのディレクトリにソフトウェアをインストールします。

dh\_installdirs で、必要なディレクトリを *debian/package* 以下に作成します。

*debian/package* ディレクトリにソフトウェアをインストールします。autoconf/automake を利用しているソフトウェアであれば、`make install DESTDIR=$(PWD)/debian/package/` のようなコマンドでインストールできます。

#### 4.1.4 debian/rules binary

ここでは、インストールしたソフトウェアを Debian パッケージにするための最終的な微調整を実施します。dh\_testroot で root であることを確認します。

dh\_installdocs でドキュメントファイルを *debian/package/usr/share/doc/XXX* にコピーします。

dh\_installexamples でドキュメントファイルを *debian/package/usr/share/doc/examples/XXX* にコピーします。

dh\_installman で *debian/package/usr/share/man/manX/XXX.X* にコピーします。

dh\_link で必要なシンボリックリンクを作成します。

dh\_strip で *debian/package/usr/bin/XXX* や *debian/package/usr/lib/XXX* に存在している実行ファイルのデバッグ情報を strip します。

dh\_compress で *debian/package/usr/share/doc/XXX* や *debian/package/usr/share/man/manX/XXX.X* などにある、ポリシーで gzip 圧縮しておくべきとされているファイルを gzip 圧縮します。

dh\_fixperms で *debian/package/usr/share/doc* 以下の実行権限をはずしたりして、アクセス権を修正します。

dh\_installdeb で制御ファイルを DEBIAN ディレクトリーにコピーします。*debian/package.XXX* を *debian/package/DEBIAN/XXX* にコピーします。ここで対象となるのが、下記です。

- postinst
- preinst
- postrm
- prerm
- shlibs
- conffiles

*debian/package/etc* 以下にファイルがある場合は、*debian/package/DEBIAN/conffiles* に追記されます<sup>\*1</sup>。postinst/preinst/postrm/prerm に関しては、#DEBHELPER#と記述されている部分は *debian/package.XXX.debhelper* ファイルの中身で置換されます<sup>\*2</sup>。

dh\_shlibdeps で、共有ライブラリの依存関係を解析します。*debian/package* 以下にある実行ファイルと共有ライブラリの一覧を dpkg-shlibdeps に渡します。*debian/package.substvars* に出力させます。内容とし

---

<sup>\*1</sup> debconf V3 以上

<sup>\*2</sup> Debian/Debhelper/Dh\_Lib.pm

ては、依存するパッケージの一覧を `shlibs:Depends` 変数の定義として出力します。

`dh_gencontrol` は、`dpkg-gencontrol` コマンドを利用します。`debian/changelog` ファイルを解析し、バージョンを調べ、`debian/control` のパッケージ部分に対して、`debian/package.substvars` にある変数置換を実施し、`debian/package/DEBIAN/control` を生成します。また、`Installed-Size` 情報を `du` を実行して作成します。

```
--- /tmp/control      2005-04-08 07:49:13.236346992 +0900
+++ debian/whizzytex/DEBIAN/control  2005-04-08 06:17:20.000000000 +0900
@@ -1,6 +1,11 @@
Package: whizzytex
+Version: 1.2.2-2
+Section: tex
+Priority: optional
Architecture: all
Depends: emacs, tetex-bin (>= 2.0.2-17), advi | xdvi | gv
+Installed-Size: 584
+Maintainer: Junichi Uekawa <dancer@debian.org>
Description: a WYSIWYG emacs environment for LaTeX
 WhizzyTeX is an emacs minor mode for incrementally
 (TeXing and) previewing a LaTeX file while editing at real-time.
```

`dh_md5sums` で `debian/package/DEBIAN/md5sum` に `debian/package` 以下にあるファイルの `md5sum` の結果を記録します。

`dh_builddeb` で `dpkg-deb --build` コマンドを呼び出し、パッケージをビルドします。`debian/package/DEBIAN` を制御情報として利用し、`debian/package/`以下をパッケージのデータとして利用し、`../package_XXXX_XXX.deb` などを作成します。

#### 4.1.5 参考資料

`debhelper` で作成される `debian/rules` ファイルを参考のため載せておきます。

```

#!/usr/bin/make -f
# -*- makefile -*-
# Sample debian/rules that uses debhelper.
# This file was originally written by Joey Hess and Craig Small.
# As a special exception, when this file is copied by dh-make into a
# dh-make output file, you may use that output file without restriction.
# This special exception was added by Craig Small in version 0.37 of dh-make.

# Uncomment this to turn on verbose mode.
#export DH_VERBOSE=1

CFLAGS = -Wall -g

ifneq (, $(findstring noopt, $(DEB_BUILD_OPTIONS)))
    CFLAGS += -O0
else
    CFLAGS += -O2
endif

configure: configure-stamp
configure-stamp:
    dh_testdir
    # Add here commands to configure the package.

    touch configure-stamp

build: build-stamp
build-stamp: configure-stamp
    dh_testdir

    # Add here commands to compile the package.
    $(MAKE)
    #docbook-to-man debian/test-source.sgml > test-source.1

    touch build-stamp

clean:
    dh_testdir
    dh_testroot
    rm -f build-stamp configure-stamp

    # Add here commands to clean up after the build process.
    -$(MAKE) clean

    dh_clean

install: build
    dh_testdir
    dh_testroot
    dh_clean -k
    dh_installdirs

    # Add here commands to install the package into debian/test-source.
    $(MAKE) install DESTDIR=$(CURDIR)/debian/test-source

# Build architecture-independent files here.
binary-indep: build install
# We have nothing to do by default.

# Build architecture-dependent files here.
binary-arch: build install
    dh_testdir
    dh_testroot
    dh_installchangelogs
    dh_installdocs
    dh_installexamples
#    dh_install
#    dh_installdmenu
#    dh_installddebconf
#    dh_installogrotate
#    dh_installemacsen
#    dh_installdpam
#    dh_installdmime
#    dh_installdinit
#    dh_installdcron
#    dh_installdinfo
    dh_installdman
    dh_link
    dh_strip
    dh_compress
    dh_fixperms
#    dh_perl
#    dh_python
#    dh_makeshlibs
    dh_installddeb
    dh_shlibdeps
    dh_gencontrol
    dh_md5sums
    dh_builddeb

binary: binary-indep binary-arch
.PHONY: build clean binary-indep binary-arch binary install configure

```

## 5 個人提案課題

---



名前 \_\_\_\_\_

下記の空欄を埋めてください:

Debian の ( )  
に注目し日本人のために ( )  
を実現します .

企画案の図 :



## 6 グループ提案課題



名前 \_\_\_\_\_  
名前 \_\_\_\_\_  
名前 \_\_\_\_\_

名前 \_\_\_\_\_  
名前 \_\_\_\_\_  
名前 \_\_\_\_\_

下記の空欄を埋めてください:

Debian の ( )  
に注目し日本人のために ( )  
を実現します .

企画案の図 :

## 7 Keysigning Party

上川純一



### 事前に必要なもの

- 自分の鍵の fingerprint を書いた紙 (gpg --fingerprint XXXX の出力.)
- 写真つきの公的機関の発行する身分証明書, fingerprint に書いてある名前が自分のものであると証明するもの

### キーサインで確認する内容

- 相手が主張している名前の人物であることを信頼できる身分証明書で証明しているか<sup>\*3</sup>.
- 相手が fingerprint を自分のものだと言っているか
- 相手の fingerprint に書いてあるメールアドレスにメールをおくって, その暗号鍵にて復号化することができるか

### 手順としては

- 相手の証明書を見て, 相手だと確認
- fingerprint の書いてある紙をうけとり, これが自分の fingerprint だということを説明してもらう
- (後日) gpg 署名をしたあと, 鍵のメールアドレスに対して暗号化して送付, 相手が復号化してキーサーバにアップロードする (gpg --sign-key XXXXX, gpg --export --armor XXXX )

---

<sup>\*3</sup> いままで見た事の無い種類の身分証明書を見せられてもその身分証明書の妥当性は判断しにくいので, 学生証明書やなんとか技術者の証明書の利用範囲は制限される. 運転免許証明書やパスポートが妥当と上川は判断している

## 8 次回



---

次回は 5 月 21 日土曜日の夜を予定しています．内容は本日決定予定です．  
参加者募集はまた後程．