



## 第 6 回 東京エリア Debian 勉強会 事前資料\*

Debian 勉強会会場係 上川純一†

2005 年 7 月 2 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	4
2.1	2005 年 24 号 . . . . .	4
2.2	2005 年 25 号 . . . . .	5
2.3	2005 年 26 号 . . . . .	6
2.4	2005 年 27 号 . . . . .	6
3	最近の Debian 関連のミーティング報告	7
3.1	東京エリア Debian 勉強会 6 回目報告 . . . . .	7
4	glibc and toolchains	8
4.1	toolchain とは . . . . .	8
4.2	Debian ではどうなっているか . . . . .	8
4.3	今後どうするか . . . . .	8
5	dpatch をつかってみよう	9
5.1	dpatch とは . . . . .	9
5.2	ファイル構成 . . . . .	9
5.3	道具 . . . . .	9
5.4	作業フロー . . . . .	10
5.5	今後の開発 . . . . .	11
6	個人提案課題	13
7	Keysigning Party	14
8	次回	15

## 1 Introduction To Debian 勉強会

上川純一



今月の 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 をがりがりと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

- gotom さん Debian の glibc とかをやってます．
- 上川純一 宴会の幹事です．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 や，libpkg-guide などを通して，Debian の品質向上を目指しています．

### 1.2 事前課題紹介

今回の事前課題は「いままで toolchain をつかってみて/今後 toolchain に期待すること」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 makoyuki さん

えー、すいません、当方、そもそも「toolchain」といわれて何のことかわかりませんでした。大体「開発環境一式」とかなあ?という感じだったのですが、正確に理解しているかわかりません。

んで、仮にそうだと、ある程度のコンパイルとかはしていても、当方コード書きではまったく無い上、少なくとも Debian においては一度も手動でソースコードからコンパイルしたことがありませんし、プログラミング周りのことについてはド素人なのでド素人なりのことを書いてみます。

開発環境であるという前提で書きますが...正直に包み隠さず言いますと、当方は実際のところプログラミン

グっばいことがしたくないから Debian を使っています。コードは書きたくないのです。万が一それが楽しいと思うことがあれば今後絶対にやらないということはありませんが、少なくとも今、私がやりたいのはプログラミングではありません。なので、「toolchain を使う」ということは、当面ないと思うのです。いままでも。そしてこれから当分の間は。

勿論、勉強にもなりますし、理解を深める一助にもなるでしょうから、それについての勉強をしたいとは思っています。その上で、限度はあるにせよ、ドキュメントの充実、ツールやインターフェースの充実により、レベルは下げずに敷居を下げるものであって欲しいという浅はかな望みはありますが、それはなんとなく今回のお題にはそぐわない気がします。

そもそもが toolchain が何のことか理解していないかもしれませんので完全にハズしているかもしれませんがご勘弁を。

### 1.2.2 澤田さん

toolchain に期待すること、ということいろいろ妄想してみた。

まず現実的なもの。最適化は結構ブラックボックスなイメージがあるので、最適化を行ったときにどの部分にどんな最適化を施したのか、逆にここはなんで最適化をかけなかったのか（時間と空間のトレードオフとか）、最適化をかけることでどんな効果が得られるのかのログがあるといいなと思った。

もうひとつまあ現実的なもの。リンク時にどのライブラリをリンクしないといけないのか考えるのが面倒なので、ライブラリが export しているシンボルを保存しておき、リンク時に参照する機能があるといいなと思った。この場合、どのライブラリを使っているかわからなくなってしまうので、デフォルトオフで `-library-auto-detect` とかいうオプションがあるといいだろうか。

次に実現できるかよくわからないもの。コンパイル時には inline 展開なるものがあるわけだが、リンク時（共有ライブラリ使ってる場合はロード時）にも inline 展開してくれるとうれしいことがあるかもしれないと思った。

最後はかなり非現実的なもの。最適化を行う際に一番問題なのはやはりどの最適化オプションを使えばいいのかわからないということであろう。これを解決する方法として、こんなコードに対してはこんな最適化が有効だったというデータベースを構築し、最適化時にそれを参照、実行してデータベースに反映とかすると何ができるのではないかなと思った。

### 1.2.3 上川

etch にむけて g++ の transition があるらしい。ABI がまた変わるとのこと。gcc 3.0 になるときに「これで最後だから」といってみておきながら、また浮気するのですか？あぁ...

woody から sarge にあがる際に、c++ の ABI が一旦変更になっています。アップグレードの際にこまったりしませんでしたか？

## 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 24 号

6 月 14 日版です。

問題 1. Alioth が新しいサーバに移動するということが各ユーザ宛のダイレクトメールで出た。Branden Robinson が Alioth のアナウンスメールについてコメントを出した。その主旨は。

A 匿名で各ユーザにメールを出すのではなく、debian-devel-announce に出そう、さらに誰が出したのかを明確にして出そう

B メールは信頼できないプロトコルなので WEB ページに掲示したらよいだろう

C SMTP より、SNMP トラップのほうがみんな気づく

問題 2. Andreas Barth は etch に向けてリリースポリシーについてどういう変更が必要だと説明したか

A ダメな人禁止

B ダメなコード禁止

C 共有ライブラリはダイナミックにリンクすべきであり、ソースコードをコピーするのは禁止しよう。

問題 3. etch にむけて c++ の ABI が変更になるため、Matthias Klose は何を宣言したか

A C++ でかいたプログラムの肅正

B とりあえず今 C++ のライブラリであたらしい ABI バージョンのものについてはアップロードしないようにフリーズする

C java の利用禁止

問題 4. Debconf での講演の予定の組み方について今回はどういうところみがなされたか

A 意志決定方法としてさいころを導入してみた

B 参加者の都合が悪い日をできるだけ選んだ

C 参加したい講演の投票をおこない、参加者ができるだけ参加できるようにスケジューリングをする

問題 5. Scott James Remnant 曰く , dpkg 1.13 で新しく追加されたアーキテクチャ変数名は

- A DPKG\_HOST\_ARCH\_OS
- B DEBIAN-ARCHITECTURE
- C uname -a

問題 6. Roberto Sanchez の作成した FAQ には何がかいてあるか

- A Debian パッケージをインストールする方法がかいてある
- B Debian パッケージをカスタマイズしてビルドする手順がかいてある
- C Debian メンテナを招集する方法がかいてある

問題 7. selinux の進捗についてはどうなっているか

- A coreutils などのパッケージが libselinux1 に依存するようになってきた
- B NSA の陰謀であることが判明したので排除
- C もうほとんどのユーザが SELINUX を使うようになった

## 2.2 2005 年 25 号

6 月 21 日版です.

問題 8. Oskuro が GNOME について宣言したのは何か

- A 今後は KDE に移行する
- B Gnome 2.10 が unstable に入った
- C GNOME3 がもうすぐリリース

問題 9. woody から sarge にアップグレードするのにもっとも大きい問題は何か

- A 相互に依存するような依存関係があるパッケージのアップグレード
- B woody では存在したけど sarge で消えたパッケージの処理
- C etch に無いパッケージ

問題 10. Aurelien Jarno が BSD 移植版について報告した問題は

- A GNU Hurd についてはもうあきらめたので考えなくて良い
- B FreeBSD についてはもうあきらめた .
- C libselinux1 に依存するパッケージは , selinux が linux のみなので , Linux のみでリンクするようにして

ほしい

問題 11. Bill Allombert が menu について報告したのは

- A menu のセクションの部分については国際化できるようになった
- B menu はもう使われていないので廃止する
- C 対応する Window Manager が増えた

問題 12. 矢吹さん曰く大阪でなにかあるらしい, 何?

A 10 月に mini Debian Conference を開催する

B 9 月にたこやきはやくい競争

C 阪神優勝祈念会

## 2.3 2005 年 26 号

6 月 28 日版です.

## 2.4 2005 年 27 号

### 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 6 回目報告

前回開催した第 6 回目の勉強会の報告をします。

DWN クイズを実施しました。今回は 3 週間分だったので、分量がすくなかったかな。

えとーさんが update-alternatives と dsys について説明しました。update-alternatives が 実は root 権限がなくても使えるようになっていたり、新しい発見がありました。dsys については、alternatives の Description(説明文) というものがないのでその情報が欲しい、さらに国際化するには何が必要だろう、という話題が出ました。

武藤さんが debian-installer の構造について話をしました。国際化はまだ etch にむけて改善できるだろう、ということです。udeb をつかっていたり、anna をつかっていたり、cdebconf を使っているところとか。1st-stage と 2nd-stage があって、全然違うとか。インストーラが動いているときには DEBCONF\_PRIORITY は HIGH なのにインストール後は MEDIUM に強制変更するためいろいろとひずみがでているとか。なお、自動認識されないハードウェア情報については、バグレポートが重要なので、lspci, lspci -n の出力とともにおくってくださいとのこと。lspci -n の出力はしらなかったけど便利。

GPGKeysigning party をしました。キーサイングに参加するのは 10 人なので、あまり時間かからないかなあ、と思ったら 30 分くらいかかってしまいました。

はなの舞で宴会、デニーズでデザート。



## 4 glibc and toolchains

gotom



- 
- 4.1 toolchain とは
  - 4.2 Debian ではどうなっているか
  - 4.3 今後どうするか

## 5 dpatch をつかってみよう

上川純一



### 5.1 dpatch とは

Debian のソースパッチを管理するツールです。Debian パッケージでは、ソースパッケージは以下の構成になっています。

- .orig.tar.gz: オリジナルの tarball
- .diff.gz: Debian で作成した差分
- .dsc: dpkg 用制御ファイル

この中で、.diff.gz は一つの大きな差分ファイルとして管理されるため、どの部分がどういうパッチであるか、ということ进行管理してはくれません。その部分を実装するのが dpatch です。

通常の.diff.gz であると、debian/ディレクトリ以下の Debian パッケージング用の情報とそれ以外のソフトウェア自体への修正が混合しています。それを整理するというものです。

### 5.2 ファイル構成

dpatch では、それぞれの小さな変更をそれぞれ独立したパッチとして扱います。それぞれのパッチを debian/patches/xx\_patchname.dpatch という名前で管理します。例えば、debian/patches/01\_configure.dpatch という名前になります。そして、パッチの一覧を debian/patches/00list に適用する順番に記述します。そこでは、01\_configure というような形式で記述できます。

この形式を採用しているため、Debian においての変更点が debian/ディレクトリ以下に集まり、また変更点を debian/patches ディレクトリで分類して管理できる、という利点があります。

### 5.3 道具

dpatch を利用するための道具を紹介します。

#### 5.3.1 dpatch

dpatch コマンドは、パッチの適用とパッチをはずすという処理を実施してくれるコマンドです。従来は、makefile から include する Make スクリプトとして実装されていましたが、dpatch 2.0 からは実体が /usr/bin/dpatch シェルスクリプトになっています。

古いdebian/rules では下記の内容を記述しています。

```
include /usr/share/dpatch/dpatch.make
```

最近の dpatch を利用するソースでは、dpatch コマンドを呼ぶように実装すればよいことになっています。  
(/usr/share/doc/dpatch/examples/rules/rules.new.dh.gz から抜粋)

```
#!/usr/bin/make -f
#
# Sample dpatch rules file. Only example. Nothing else. :)
# This one uses the new way with dpatch from dpatch 2.x

export DH_COMPAT = 4

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

build: build-stamp
build-stamp: patch
    @echo "--- Compiling"
    dh_testdir
# Do something to build your package here
    touch build-stamp

clean: clean1 unpatch
clean1:
    @echo "--- Cleaning"
    dh_testdir
    dh_testroot
    dh_clean -k
# Clean your build tree

install: build-stamp
    dh_testdir
    dh_testroot
    dh_clean -k
    dh_installdirs
# Install it here

# Build architecture-independent files here.
binary-indep: build install

# Build architecture-dependent files here.
binary-arch: build install
    dh_testdir
    dh_testroot
    dh_installdocs
# And all the other dh.* stuff you need for your package.

# And now the simple things for dpatch. Here we only apply/unapply the patches.
# You can do more things with dpatch, like having patches only applied on
# a special architecture - see the non-dh version of the sample for this!
patch: patch-stamp
patch-stamp:
    dpatch apply-all
    #dpatch call-all -a=pkg-info >patch-stamp

unpatch:
    dpatch deapply-all
    rm -rf patch-stamp debian/patched

binary: binary-indep binary-arch
.PHONY: binary clean binary-indep binary-arch build install patch unpatch \
    clean1
```

### 5.3.2 dpatch-edit-patch

dpatch で利用するためのパッチを作成するコマンドです。

### 5.3.3 dpatch.el

emacs 上で dpatch を使うために必要な、00list ファイルや、.dpatch ファイルの編集用のモードです。  
まだまだ未完成です。

目標は dpatch-edit-patch の実装ですが、コアの dpatch の部分の変更しているばかりで時間が過ぎていきそうです。

## 5.4 作業フロー

### 5.4.1 あたらしい upstream が出た時

- 前のバージョンから debian/ディレクトリをコピーしてくる

- `debian/patches/xx_patchname.dpatch` をそれぞれ `patch --dry-run -p1` で適用できるかどうか確認する
- `debian/patches/00list` を編集
- `debian/rules patch`
- 適用できないパッチの再作成

#### 5.4.2 あたらしく package をつくる時

- `debian/rules` を変更し, `/usr/share/dpatch/dpatch.make` を include し, `clean` と `configure` のルールで `patch` と `unpatch` ルールが呼ばれるようにする (`patch-stamp` 等を利用)
- `debian/patches/00list` を作成する
- `dpatch-convert-diffgz` を実行, とりあえず `dpatch` ファイルに変換する
- `dpatch` ファイルを適当に編集して複数ファイルに分割
- `debian/rules patch` と `debian/rules unpatch` が成功する
- `dpkg-buildpackage` で生成される `diff.gz` の `diffstat` をとり, `debian` 以下以外の場所が `diff` に含まれていないことを確認

#### 5.4.3 あたらしく patch を作成

- `dpatch-edit-patch` パッチ番号\_パッチ名とする.  
例: `dpatch-edit-patch 03_automake`  
もし他のパッチに依存する変更をするなら, そのパッチ名を入力します.  
例 `dpatch-edit-patch 03_automake 02_autoconf`.  
これを実施すると, `/tmp` 以下に適当なディレクトリでソースが編集できるようにシェルが起動します.
- ソースを適当に編集します.
- `exit` すると, パッチファイルが `debian/patches` ディレクトリ以下に生成されます.
- 生成されたパッチファイルに適当なコメントを書いておきます
- `debian/patches/00list` を適当に修正します
- `debian/rules unpatch`  
`debian/rules patch`  
`debian/rules unpatch` として 一応パッチが動作することを確認します

#### 5.4.4 すでにあるパッチを編集

- `dpatch-edit-patch` パッチ番号\_パッチ名とする. 例: `dpatch-edit-patch 03_automake`
- 編集する
- `exit`

### 5.5 今後の開発

がんばれ. といいたいところですが, とりあえず現状の仕様をあらいだして, テストを作成して, 全部の機能が動作することを確認するところからやろうと考えています. 今は謎の機能が多すぎておいそれと触れま

せん .

## 6 個人提案課題

---



名前 \_\_\_\_\_

下記の空欄を埋めてください:

Debian glibc の ( )  
に注目し , 今後の Debian の toolchain は ( )  
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>\*1</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>\*1</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 8 月 6 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．