



東京エリア Debian 勉強会

Debian 勉強会幹事 コバヤシ ノリタダ
2007 年 2 月 17 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？

目的として次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について情報共有する場をつくる
- Debian を利用する際の情報をまとめて、ある程度の塊として整理するための場をつくる

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。

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。

目次

1	Introduction	1
2	事前課題	3
2.1	3
2.2	上川	3
3	Debian Weekly News trivia quiz	4
3.1	2007 年 XX 号	4
4	最近の Debian 関連のミーティング報告	5
4.1	東京エリア Debian 勉強会 23 回目報告	5
5	Debian 勉強会 2007 年度計画検討結果	6
6	dpatch updates 2007	7
6.1	インストールのしかた	7
6.2	作業のしかた	7
6.3	各種ツールを利用してみる	7
6.4	debian/ディレクトリのみを展開したパッケージのメンテナンスをする	7
6.5	今後の開発指針	8
6.6	参考文献	8
7	dbfs	9
7.1	dbfs とはなにか	9
7.2	インストールのしかた	9
7.3	使いかた	9
7.4	まとめ	12
8	次回	13

2 事前課題

上川純一

今回の事前課題は「XXXX」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。

2.1

2.2 上川

3 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 界隈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。

3.1 2007 年 XX 号

<http://www.debian.org/News/weekly/2007/XX/> にある X 月 X 日版です。

問題 1.

- A
- B
- C

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 23 回目報告

5 Debian 勉強会 2007 年度 計画検討結果

上川純一

2007 年度の計画を検討した結果次のような内容が出てきました。

Debian が現在市場に提供している付加価値としては

デプロイしやすいしくみであること、
ライセンスが明確であること
たくさんソフトウェアがあること
アップグレードが平穩であること
くだらないパッケージもはいつていること
ソースコードが簡単にとってこれること
ユーザが多いこと
学ぶことがおおいこと

というものがある。

また外部的な要因として

Windows VISTA の公開 テスト開始

Mac OSX iPhone, Leopard

他のディストリビューション FC, SUSE, RHEL の新リリースが出る、Ubuntu も出るが、宇宙に再度いってしまうのではないかな？

ハードウェア Quad core, Wii/PS3/.. などが出る

ユーザの期待として 次はさすがに GUI で D-i できるだろう

というのがあある。

ユーザの拡大方向としては、

- 主婦
- 学生 → 教育に Debian を利用すべきだ

また、今年の活動は、来年を考えて動くべきだろう。

以上をふまえての今年度の計画は

- 2 小林さん幹事
- 3 岩松さん担当で OSC
- 4 えとーさんでエッチインストール大会
- 5 ごとむさん主催で google 開催 (! ?)
- 6 やぶきさんでえじんばら開催
- 7 Debconf 参加報告会
- 8 Debian 14 周年

9以降 後で考える。



6 dpatch updates 2007

上川純一

dpatch を使うときにもしかすると便利かもしれない小ネタを報告します。

6.1 インストールのしかた

```
apt-get install dpatch
```

でインストールできます。

6.2 作業のしかた

debian/rules を適切に変更したあと、dpatch-edit-patch で編集します。^{*1}

パッチは debian/patches 以下で管理されます。debian/patches/00list ファイルに適用するパッチの一覧があり、それを参照して パッチを適用します。

6.3 各種ツールを試してみる

各種ツールがあるのでみてみましょう。

dpatch メインのツールです。表立って直接利用することはありません。debian/rules などから呼び出されます。

dpatch-edit-patch 編集のツールです。パッチを作成、もしくは編集する際に利用します。dpatch-edit-patch パッチ名 適用するパッチ という形で指定すると、指定した適用するパッチまでが適用された状態のソースツリーが一時ディレクトリに展開された状態でシェルが起動します。そこで編集し、シェルを終了 (ctrl-D もしくは exit) するとパッチが作成され、debian/patches 以下にファイルが生成されます。

dpatch-convert-diffgz .diff.gz から dpatch ファイルを生成するためのツールです。

dpatch-get-origtargz dpatch-edit-patch が内部的に利用するツール。

6.4 debian/ディレクトリのみを展開したパッケージのメンテナンスをする

dpatch で管理すると、Debian のソースパッケージのうち、debian/以下のディレクトリに対しての変更しか発生しなくなります。.orig.tar.gz を展開した状態でのパッケージのメンテナンスは実は必要なく、debian/ 以下だけをバージョン管理ツールで管理して、orig.tar.gz は必要に応じてアップストリームからダウンロードしてくるというスタイルで開発をすることができます。

dpatch-get-origtargz はそのためのツールで、現在の debian/ ディレクトリに対応した .orig.tar.gz を環境変数 DPGO_ORIGTARDIR で指定したパス上に存在する .orig.tar.gz から探してきます。ローカルで見付からなければ uscan や apt-get source などを利用して、ネットワークからとってきてくれたりもします。

^{*1} /usr/share/doc/dpatch/examples/rules/rules.dh.gz 参照

svn-buildpackage などは、この方法を利用して dpatch と連携することが可能です。

6.5 今後の開発指針

6.6 参考文献

- 2005 年 7 月 Debian 勉強会資料



7 dbs

岩松

7.1 dbs とはなにか

dbs は Debian Build System の略です。dpatch や quilt はパッチを管理する方法に重点を置いているのですが、dbs は名前のとおり、ビルドまでの面倒を見るためのツールです。使いかたとしては、dbs も dpatch もあまり変わりません。debian/rules で専用のライブラリを include して使うだけです。ただ、作法がありところどころ違うところもあります。今回は dpatch と比べてどこが違うのかを比べてみようと思います。

7.2 インストールのしかた

```
# apt-get install dbs
```

7.3 使いかた

dbs の使うためのサンプルとして hello-dbs^{*2} というパッケージが存在します。これを使ってどのように dbs を使うのか見ていこうと思います。

7.3.1 hello-dbs を取得

hello-dbs のソースパッケージを取得します。

```
# apt-get source hello-dbs
```

7.3.2 展開されたソースパッケージ

展開されたソースパッケージ内をみると、以下のような構成になっています。

```
iwamatsu@chimagu:~/dev/debian/dbs/hello-dbs-1.3$ ls
debian hello-1.3.tar.gz
```

ここでわかる dbs を使ったパッケージの特徴として、ソースパッケージは tar.gz 形式で提供されている点です。dh_make を使って生成されたパッケージではこのような構成にはなっていません。upstream のソースパッケージは Debain 標準の形ではないということです。

しかし、debian ディレクトリは、他のパッケージの構成とあまり変わりません。

```
iwamatsu@chimagu:~/dev/debian/dbs/hello-dbs-1.3$ ls debian/
README.Debian README.build changelog compat control copyright dirs hello.1 patches rules
```

7.3.3 debian/rules ファイル

dbs では debian/rules に設定項目を書くことによって、細かい設定を行うことができます。hello-dbs では以下の設定を行っています。

^{*2} <http://packages.debian.org/unstable/devel/hello-dbs>

```
# DBS options

package      := hello-dbs
PWD          := $(shell pwd)
CFLAGS       := -O2 -Wall
INSTALL = install
INSTALL_DATA := $(INSTALL) -m644
INSTALL_DIR  := $(INSTALL) -p -d -o root -g root -m 755
INSTALL_FILE := $(INSTALL) -p      -o root -g root -m 644
INSTALL_PROGRAM := $(INSTALL) -m755
INSTALL_SCRIPT := $(INSTALL) -p      -o root -g root -m 755
SCRIPT_DIR   = /usr/share/dbs

# the dbs rules
TAR_DIR := hello-1.3.orig
include $(SCRIPT_DIR)/dbs-build.mk
```

以下に各設定項目の内容を示します。

- package
パッケージ名
- PWD
パッケージカレントディレクトリ
- CFLAGS
C コンパイラに設定するオプション
- INSTALL_DATA
インストールするデータのパーミッション
- INSTALL_DIR
インストール先ディレクトリのパーミッション
- INSTALL_FILE
インストールするファイルのパーミッション
- INSTALL_PROGRAM
インストールするプログラムのパーミッション
- INSTALL_SCRIPT
インストールするスクリプトのパーミッション
- SCRIPT_DIR
dbs スクリプトディレクトリパス
- TAR_DIR
tar 解凍後ディレクトリ名

これら以外に

`include $(SCRIPT_DIR)/dbs-build.mk` で `include` することにより、`dbs` を使うことができるようになります。

7.3.4 パッチ

`dbs` では `dpatch` と同様、`debian/patches` ディレクトリにパッチを格納する必要があります。パッチはユニファイド diff 形式 で書かれている必要があり、ファイル名の先頭には数字を付けます。数字とファイル名の間はアンダースコアで区切ります。特に拡張子等は必要ありません。`dbs` はこの数字が割り当てられている順にパッチをソースコードに適用していきます。

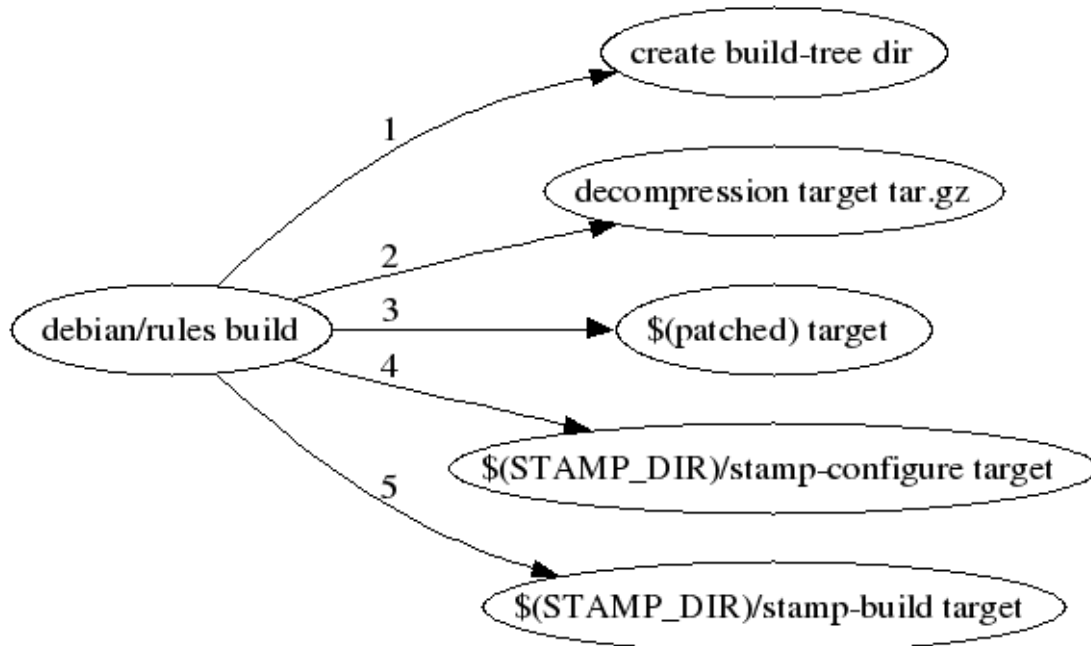
`dpatch` では `00list` というファイルにパッチ名を書き、書かれた順にパッチを適用していきます。パッチの順番が変わったときは、`00list` ファイルを修正すればいいだけですが、`dbs` の場合はパッチの順が変わったときはファイル名を変更しないといけません。

また、`hello-dbs` パッケージでは `00_maillocation` パッチがあります。

```
% ls -l debian/patches
00_maillocation
```

7.3.5 パッケージのビルド

パッケージのビルドが行われるとき、以下の図の順でターゲットが呼ばれます。



1. create build-tree ディレクトリ

`$BUILD_TREE` で指定されたディレクトリを作成します。デフォルトでは `build-tree` になっています。

2. decompression tar.gz

ソースコードが圧縮されている `tar.gz` を `build-tree` ディレクトリに解凍します。

3. \$(patched) target

`debian/patches` ディレクトリにあるパッチを適用します。

4. \$(STAMP_DIR)stamp-configure target

`stamp-configure` ターゲットを実行します。hello-dbs では `$BUILD_TREE` ディレクトリに移動し、`configure` を実行します。

5. \$(STAMP_DIR)/stamp-build target

`stamp-build` ターゲットを実行します。hello-dbs では `$BUILD_TREE` ディレクトリに移動し、`make` を実行します。

build 時の `dpatch` との違いは、

- パッチを当てるためのターゲット名が異なる `dbs` は `$(patched)`、`dpatch` は `patch-stamp`。
- マーク用のファイル名が異なる `stamp-configure` や `stamp-build` というマーク用のファイル名が逆だったりします。(通常は `configure-stamp` / `build-stamp`)

7.3.6 パッケージの clean

`dbs` のソースの `clean` ターゲットはいたってシンプルです。`dpatch` は既に展開されているソースにパッチを適用するため、`clean` ターゲット時には適用されたパッチを外す処理が必要になりますが、`dbs` では、build 時に生成されたマークファイル用ディレクトリやソース格納先ディレクトリ (`$BUILD_TREE`) をばっさり削除します。これには、適用されたパッチの管理等を行わずに済むというメリットがあります。

しかし、dbs は パッケージビルド毎に

1. ディレクトリを削除
2. tar.gz を解凍
3. パッチ適用

とするので、サイズの大きい tar.gz をパッケージ化するときは時間がかかるというデメリットもあります。

7.4 まとめ

今回、dbs を触ってみたのですが、

- ソース見えないソースが tar で固まっているため、見るができない。見るには コマンドを使って解凍する必要がある。
-

8 次回

未定です。内容は本日決定予定です。
参加者募集はまた後程。



会 勉 強 会 Debian イト



Debian 勉強会資料

2007 年 2 月 17 日 初版第 1 刷発行
東京エリア Debian 勉強会（編集・印刷・発行）
