

東京エリア Debian 勉強会

第83回 2011年12月度

野島 貴英 nozzy@debian.or.jp

IRC nick: nojima

twitter: nozzy123nozzy

2011年12月17日

設営準備にご協力ください。

会場設営などよろしくおねがいします。

- 注意事項
 - 飲酒禁止
 - 宗教禁止
 - 営利活動禁止
- 最近あった Debian 関連のイベント報告
 - 第 81 回 東京エリア Debian 勉強会 (筑波)
 - 第 82 回 東京エリア Debian 勉強会 (OSC)
- Debian Trivia Quiz
- 事前課題紹介
- 2011 年の振り返り
- quilt で porting してみた
- 月刊 Debhelper 第 2 回



イベント報告



東京エリア
Debian 勉強
会(筑波)

東京エリア Debian 勉強会 in 筑波

- 開催場所は 筑波大学
- 10/22 (土)13 時に現地集合
- 学生さんも参加されました。
- Debian ユーザ寄りの発表が多め?でした。



東京エリア
Debian 勉強
会(OSC)

- 開催場所は 明星大学
- 11/19(土) 11:00-11:45 で最近の Debian についてのセッションが行われました。
- ブースを開きました(2日間) LiveCD も配りました。



DWN quiz




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




`debian-devel-announce@lists.debian.org`,
`debian-devel@lists.debian.org` に投稿された内容と
Debian Project News などからです。

問題1. 11月終わり頃にルートファイルシステムの構造について議論を呼んでます。内容は?

-  A /user を作る
-  B /bin, /sbin, /lib の実体を /usr 以下に移動して、代わりにシンボリックリンクにする
-  C /etc の実体を /usr 以下に移動して、代わりにシンボリックリンクにする

問題1. 11月終わり頃にルートファイルシステムの構造について議論を呼んでます。内容は?

答えは:

-  A /user を作る
-  B /bin, /sbin 以下に移動
ポリックリ
-  C /etc の移動して、代
リンクにす



B




問題1. 11月終わり頃にルートファイルシステムの構造について議論を呼んでます。内容は?

解説: 他主要ディストリビューションが採用検討中...

問題2. sun-java6 が Debian パッケージとして配布できなくなりました。代わりに Debian で推奨される Java は?

-  A openjdk
-  B gcj-jdk
-  C coco-java

問題2. sun-java6 が Debian パッケージとして配布できなくなりました。代わりに Debian で推奨される Java は?

-  A openjdk
-  B gcj-jdk
-  C coco-jav

答えは:






A

問題2. sun-java6 が Debian パッケージとして配布できなくなりました。代わりに Debian で推奨される Java は?




解説: 残念だ > `oracle`

問題3. 11/19に長らく活動を停止していたパッケージチームが復活宣言をしました。どれでしょう?

-  A CORBA packaging team
-  B Ham-radio packaging team
-  C SDL packaging team

問題3. 11/19に長らく活動を停止していたパッケージチームが復活宣言をしました。どれでしょう?

答えは:


-  A CORBA packaging team
-  B Ham-ra
-  C SDL packaging team



C

問題3. 11/19に長らく活動を停止していたパッケージチームが復活宣言をしました。どれでしょう？


解説: これからも頑張ってくださいね



問題4. 10/28 ~ 30 で MiniDebconf2011 が開かれました。どこの国でしょう？

-  A ニカラグア
-  B インド
-  C フランス

問題4. 10/28 ~ 30でMiniDebconf2011が開かれました。どこの国でしょう？

-  A ニカラグア
-  B インド
-  C フランス


答えは:



B

問題4. 10/28 ~ 30 で MiniDebconf2011 が開かれました。どこの国でしょう？

解説: 来年は日本がいいなあ



問題5. Wheezy フリーズの為のBSP が各国で開かれました。ドイツとどこ？

-  A フランス
-  B ニカラグア
-  C ポーランド

問題5. Wheezy フリーズの為のBSP が各国で開かれました。ドイツとどこ？

-  A フランス
-  B ニカラ
-  C ポーラ

答えは:



C

問題5. Wheezy フリーズの為のBSP が各国で開かれました。ドイツとどこ？

解説: Wheezy のフリーズは2012/6 なので、開発作業はお早めに



事前課題

Twitter クライアント Polly を Ubuntu PPA が微妙だったのでパッケージ化してみました .

アイコンが Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported だったので削除しました . 自前で描くべきか微妙です。コードの多くが GPL-3+ で、一部アイコンセットが GPL-2 となっていて矛盾しているのでそのアイコンセットを削除しました。Python で書かれており、Python ライブラリを Build-Depends-*Indep* と Depends に手書きで追加しました。Depends は自動生成できたらうれしいかもしれません。

パッケージ化したものはないので、使い込んだ Debian パッケージだけです。

- kfreebsd-image-8-amd64
FreeBSD 本家ではビルドされるのに、Debian パッケージでは付属していないものが多く、その違いに悩んだ。
- gftp
ssh コマンドみたくデフォルトの ssh 鍵認証、パスワード認証でログイン処理が進んでくれず、ssh 鍵認証でログインに失敗すると画面が進んでくれない。アップストリームの問題かは不明。
- icewm

- 今年パッケージ化した Debian パッケージ:
bluez-tools, fonts-ipamj-mincho, mozc, mtdev, xf86-input-mtrack,
xf86-input-multitouch
- 今年使い込んだ Debian パッケージ:
libpng, opencv, bluez, ruby, mozc, buldd, pbuilder
- 出会った課題: release チームとのやりとり

なかおけいすけ

- パッケージ化しているパッケージ
stm32flash
- 課題
man がないとか、ライセンスファイルの書き方がおかしいとか、パッケージングポリシーに従うこと、意外に大変なこと。

Debian 構成や固有部分が絡んでるもので挙げてみました:

- busybox
OK:ビルド設定が不足して job control できない
- initramfs-tools
??:rootwait があるから rootdelay は不要と思ったら MD+USB で起動障害
 - Linux 的には上は正だが、udev も見てタイミング調整するというクイックハック状態なのだった
- grub-pc
OK:grub2 でモニタとシリアルの同時有効化機能が一時落ちて、制御不能になった

やまだ continued

- extlinux
OK:設定バラバラ化のみならず、標準ではシリアル有効化の設定を仕込む箇所が存在せず、制御不能に
- dropbear
?:initramfs 内で dropbear を起動する荒業により、固定 IP の DNS サーバが DHCP アドレスになり NW 死亡
- debhelper
?:CFLAGS(や相当設定) の指定方法がわからず、 gcc -m32 や -f... の渡し方に悩む
- kexec
OK:デフォで再起動が単なる kexec 実行に切り替わり、 BIOS が取れず悩んだ
- apt
?:実は security.d.o が 120ms-300ms の彼方で遠い・・・

emacs

常にカスタマイズを続けています。これをどれだけ使いこなすかが、課題です。

postfix のソースコードをいじってパッケージ化させてインストールしなおしました。

CN の回線上で SMTP サーバを構築する際は OP25B の関係上 OCN の SMTP サーバを通さなければならないのですが、その SMTP サーバが特殊な仕様のように postfix の設定を変えるだけではサーバに弾かれてしまいます。そこで、CN のサーバ向けにソースコードをいじりました。

- Unbound:
仕事で評価用 DNS を建てるのに使ったり、dnstudy 等で発表の危険な LT ネタとして使用。BIND に比べて良くできてるので特に課題等はなし
- zabbix:
基本 stable を使っていること、かつ発展途上のソフトであることから豊富なあまりポイントが有った。が基本的にぐぐって解決したり、表示上の問題なので無視したりしてます。

今年はあまりパッケージをいじくってないです...パッケージ化したものといえばIRCクライアントの loqui とかぐらいでしょうか。それも upstream の方が、ほとんど雛形作っていただいていたので細かい所を直しただけでした。

DEP5 への対応や他の DEP への対応などが今後パッケージクオリティを挙げる上で課題でしょうか(主に面倒くさい意味で)

今年パッケージ化した Debian パッケージ: 無しうーむ、自分専用のパッチをあてたカーネルパッケージぐらいすね。貢献はできてないな。

- 今年使い込んだ Debian パッケージ: devscripts pbuilder
devscripts でポチポチと sbuild 用にパッケージを作り、pbuilder で再ビルドしてました。今年は FTBFS を直してもらおう BTS をたまにするのと、自分専用リポジトリを最新に保つだけで、いっぱいいっぱいでした。
- 来年使い込む予定の Debian パッケージ: sbuild buildd
さて、弾ができたし、突撃じゃー。
- 出会った課題:
base.tgz とかを最初に作る時、arch=allなのはオフィシャルから、アーキテクチャ依存なパッケージはオレオレリポジトリから、という使い方ができなかった。ヘタレなおいらは、build-essential なパッケージに依存する allなのは、全部オレオレリポジトリにつっこんで解決。

まえだこうへい

12/8 時点の状況

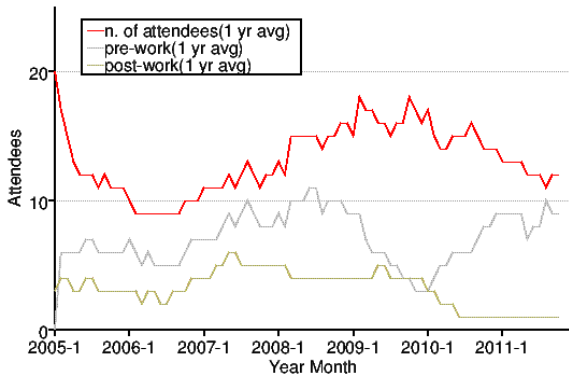
- パッケージング済み
 - python-funcparserlib : テストでコケる問題
 - python-webcolors
- パッケージング途中
 - python-ordereddict : Python2.6 のみ
 - python-blockdiag : Python 2.6 は python-ordereddict を、2.7 は組み込みの ordereddict を使うようにパッケージングするには debian/control どう書けばいいんだ?
 - python-{sec,act,nw}diag および python-sphinxcontrib.{block,seq,act,nw}diag : python-blockdiag 待ち
 - python-tomahawk : nodetests でコケる問題

- パッケージング途中
 - xmris
score の others に rw 付与を止めさせたい。古いゲームは特に。
 - tracef
dynamic loading を hook して、symbol 再ロードして欲しい。
- 使い込んだパッケージ
 - totem,gstreamer-tools
いくつか機能の動作が不完全な気が。例: 字幕ファイル利用とか。
 - gnome-shell
いろいろドキュメントなさすぎ。
 - po-mode.el
改行全部に \n を入れられてしまうので、po4a 製 po ファイルの翻訳作業がきつー。



2011年の振り返り

参加者数、事前・事後課題 年移動平均推移



- 参加者数は下降傾向
- 事前課題は横ばい傾向
事前課題をちゃんと出す=常連参加者に収斂?
- 事後課題(ブログ)の率はさらに低下

2011年参加者数実績

	参加人数	場所	内容
1月	12	荻窪	Kinect, アンケートシステム, CACert サイン会
2月	13	北新宿生涯学習館	HDFS, Debian Game Team
3月	?	OSC	CACert ATE Tokyo
4月	12	IIJ	backports, initramfs, 月刊 PPC64
5月	15	戸山生涯学習館	Apache2 モジュール, Debian on ニフクラ, Debian/m68k
6月	17	オリンピックセンター	ドキュメント処理系, 2011 再計画
7月	3	ポスニア	DebConf11
8月	12	荻窪	パッケージング関連, Debconf11 報告
9月	9	山喜旅館	Debian 温泉 2011
10月	22	筑波大学	Haskell, LaTeX, レポート自動生成, 月刊 Debhelper 開始
11月	?	OSC	CACert
12月	9	SQUARE ENIX	quilt で porting, 月刊 Debhelper

- 昨年同様、人数的には大きく増減はない
- OSC にならび、大学での開催時は参加者数および初参加者が増加
- 毎月数名は初参加者いる。うち一部は 2 回目以降も参加ちゃんと事前課題を提出する人が常連になる傾向?
- あんさんぶる荻窪以外の公民館などの利用が増加
- 三回目の Debian 温泉。伊東の山喜温泉で開催
- Debian Hack Cafe も月 1 ペースで再開

事前課題の提出率は維持しつつ、参加者数は増やしていく為の対策は必要

2011年の成果

	2010	2011
DD になった人	やまねひでき	
NM	kiwamu, 上野だいき	
DM	kurashiki, kiwamu, uwabami...	
Debian JP 加入	1 人	4 人 (東京 2 人)

タイムライン 書き直そう!


2008	2009	2010	2011	2012
<p>python3.0 ruby1.9 wine1.0, wine64 RoR2.0 で普及に 4 コア 64bitCPU, Core2Quad 普及 ニコニコ動画 1000 万 ユーザ初音ミクブーム 地デジ関連の PC 製品 普及 勉強会普及 公衆無線 LAN 携帯電話の売上落ち る iPhone, Android 登場 emobile 100 円 PC 抱きあわせ Zaurus 販 売終了 サーバの仮想化 ESXi, シンクライアント MacBook Air 発売 世界経済の崩壊 (IT 投 資緊縮財政、失職者増 加) FreeBSD7 Debian 次世代育成計 画始動 Debian Maintainer 制度始動 セキュリティ ー関連 (OpenSSL 事件、 DNS 事件) クラウド関連が流行?</p>	<p>政権交代, スパコン事業仕分け, 円高 Windows7, Snow Leopard Netwalker, Eye-Fi, Kindle2, DS LL, PSP-GO, POKEN MacBook の IEEE1394 終了, Cell 終了 ラブプラス, OSS を使ったエロ ゲー登場 (OpenCV), AR, セカ イカメラ, マジコン販売取締り, デジタルサイネージ JLS で Linux 来て大騒ぎ DD, 2 世誕生, Debian 結婚 ブーム, Lenny リリース Google Voice, Wave, Chrome, Chrome OS, Go, 日 本語入力, 徒歩ナビ Twitter, * なうブーム tile window manager, CouchDB デスクトップ 4 コア 8GB, ノ ート PC 2 コア 4GB, メモリ DDR3 に移行中, メモリ高騰 Linux が標準インストールの PC(Dell) SSD の値段と容量がこなれる (まあまあ) HDD がなくなる? 高くなる?(ならず) SSD 特化した FS が出てきた IPv6 使えるようになってる (来 年) DL 禁止法? torrent に逆風?</p>	<p>Toystory 3 デスクトップ PC:QuadCore 普通 DDR3 安い 8GB 8000 円 ノートパソコン: NetBook かとおも いきやタブレット PC が普及 iPhone4, iPad Willcom 再生 (Soft- bank に) Sun 終了 自炊ブーム, 電子書 籍, kindle とか kFreeBSD リリース されそう zfs がデフ ォルトで選べる Btrfs こなかった, ext4 きた ISO1 祭り, Android 祭り ARM 全盛 クラウド流行 KVS 大流行</p>	<p>デスクトップ終 了? 高速化しな い ノートパソコン: ChromeOS? Android? webOS 終了の お知らせ?(終了 &復活?) Adobe Flash 復活のお知ら せ (キタ), Silverlight 終 了のお知らせ (台湾を除 く)(続いている?) Squeeze リ リース (おめで とう) IPv4 割り当て の終了のお知らせ (キタ) 地上波デジタル 移行延長 (無 い?) Btrfs まだ頑張 る (Fedora 乙) Java 終了 (Sub Java 終了) Open Office が Oracle Office に (ナイ)</p>	<p>デスクトップ:終了し ている。 サーバ: 光インター コネクト?VPS 以外 はない? ノートパソコン:In- tel じゃないもの (MIPS/ARM) が主 流に。 携帯電話: ガラバゴ スの終焉。LTE が主 流になっていない。 Softbank の二年契約 が終了、SIM Free があたりまえに? 日立と東芝ハードデ ィスク事業を売却統合? 液晶が絶滅。 Oracle が Btrfs を終 了させる MySQL が OraSQL に MacBook が新しい 基盤に。</p>

2012,2013 年を予想してみよう

2012	2013
<p>ARM サーバでたけど微妙 デスクトップがノートの 1/10 ノートがタブレットに抜かれる ノートの CPU が ARM, MIPS, OS が iOS, Android OS X が iOS になる タブレット向け Debian インストールプロジェクト 立ち上げる (やまだ) カーネルはタブレットので、ユーザランドは Debian Wheezy フリーズ予定どおりされる? Debian MiniConf Japan を開催する(一同全員) node.js のパッケージを Debian パッケージにする プロジェクトを立ち上げる (上川) ARM64 コケる 企業が IPv4 目当てで企業、大学買収 3 人くらい DD になる USB3 普及, USB SAN を作る (山本) Cat5 ケーブル販売中止 有線<無線 家庭内 LAN</p>	<p>Wheezy リリースされる? E-ink のカラー版に最適 化された GUI を作る (岩松) IPv6 元年 Debian 勉強会 100 回記念までに、DD を 5 人く らい増やす (前田) 有線インターネット死亡 無線 LAN 規格変わる IEEE802.11bg 死亡 Debian GNU/iOS Debian GNU/*BSD を作る (杉本) libc また変わる Debhelper の仕組みを変える (山田) thunderbolt 死亡 aptitude の反乱 i386 ビルドもうイラネ Debian GNU/hurd はまだはやい LLVM に切り変わる</p>



quilt で port-
ing して
みた



月刊 deb-
helper 第
2回

今回発表の流れ

- ちょっとおさらい
- debhelper 共通の事柄
- 今月のコマンド: dh
- 今月のコマンド: dh_testroot
- 次回発表者

ちょっとおさらい

例: xmrisc パッケージの場合

ソース展開後のパッケージディレクトリ:

```
~/prog/xmrisc/xmrisc-4.0.5/ ls
```

```
CHANGES      PixmapList.h  defmred.h      player.c
CHANGES.4.01 README        defmris.c      puzzle.gdn
... 中略...
Icon.h        debian/       menubar.c      xmrisc.h
Imakefile     defcom.c     monster.c     xmrisc.man
Makefile.std  defcom.h     move.c
PixmapList.c  defmred.c    patchlevel.h
```

ちょっとおさらい

debian/ディレクトリ以下

```
~/prog/xmris/xmris-4.0.5/debian/ ls
README.Debian  control  patches/  xmris.manpages
README.source  copyright rules*     xmris.menu
changelog      dirs     watch     xmris.postinst
compat        docs    xmris.links xmris.postrm
```

ちょっとおさらい

debian/rule 中身

```
~/prog/xmris/xmris-4.0.5/debian/ cat rules
#!/usr/bin/make -f
%:
    dh $@ --with quilt

override_dh_auto_configure:
    xmkmf -a
```

ちょっとおさらい

debian/rules build 実行

```
~/prog/xmris/xmris-4.0.5/ debian/rules build
dh build --with quilt
  dh_testdir
  dh_quilt_patch
パッチ debian_system.patch を適用しています
patching file Imakefile
... 中略(ビルド続く)...
```

今日の暴言:

/usr/bin/dh_XXX の形のコマンドであれば
何でも debhelper コマンド
(man debhelper より)

debhelper コマンド群

```
~/ apt-file search /dh_ | fgrep /usr/bin
autotools-dev: /usr/bin/dh_autotools-dev_restoreconfig
autotools-dev: /usr/bin/dh_autotools-dev_updateconfig
bash-completion: /usr/bin/dh_bash-completion
cli-common-dev: /usr/bin/dh_auto_build_nant
cli-common-dev: /usr/bin/dh_auto_clean_nant
cli-common-dev: /usr/bin/dh_clideps
.. 中略...
~/ apt-file search /dh_ | fgrep /usr/bin | wc -l
139
```

!139 個もあるぞ。(1人2個発表で、6年間の長期連載企画!)

debhelper 共通コマンドラインオプション

全 debhelper コマンドに共通:

- -v, -verbose:
何実行してるかも見せる
- -no-act:
実行しようとする debhelper コマンドみせるだけ。実行しない。
- -ppackage, -package=package :
処理するパッケージを指定
- -a, -arch
- -i, -indep
- -s, -same-arch

説明は `man debhelper`

debhelper 共通コマンドラインオプション

全 debhelper コマンドに共通:

- -Npackage, -no-package=package:
指定パッケージを処理しない
- -remaining-packages:
すでに処理しているものは処理しない。
- -ignore=file:
debian/ の特定の file を無視する。
- -Ptmpdir, -tmpdir=tmpdir:
tmpdir をパッケージ構築ディレクトリとして利用
- -mainpackage=package
- -O=option

説明は `man debhelper`

debhelper 環境変数

指定されていると debhelper の動作変わります。

- `DH_VERBOSE`
1 にすると `-v` つけたのと同じ。
- `DH_COMPAT`
互換性度合い (COMPATABILITY LEVEL) を強制的に指定 (`debian/compat` より優先)
- `DH_NO_ACT`
1 だと `-no-act` オプションをつけたのと同じ。
- `DH_OPTIONS`
debhelper コマンドの末尾に `$DH_OPTIONS` の中身を指定したのと同じ
- `DH_ALWAYS_EXCLUDE`
指定されているファイル, ディレクトリパターンを除外。 `-X` オプションと同等。 指定例:
`DH_ALWAYS_EXCLUDE=CVS:svn` など。

互換性度合い (COMPATABILITY LEVEL)

- 目的:
過去のバージョンの debhelper との動作互換を図る為に用意。
- 指定箇所:
debian/compat ファイルに数字で指定。一時的なら DH_COMPAT に指定でも OK。
- 廃止: 1 ~ 4 (v1 ~ v4 の意味)
- 現在推奨値: 8 (v8 の意味)
- 絶賛開発中: 9 (v9 の意味)

今月のコマンド: dh

- debhelper コマンドを自動的に呼び出すだけ。
“dh シーケンス名” とすると、シーケンス名に紐づいた一連の debhelper コマンドが次々と呼び出される。

dh 起動してみる

dh clean を起動してみる。

```
~/dh clean
  dh_testdir
  dh_auto_clean
Checking a few things
Warning:Makefile is older than Imakefile
Geronimo!
rm -f foo
rm -f bar
... 中略...
```

利用可能なシーケンス名

- binary
構築からパッケージ作成まで実行するシーケンスです。
- binary-arch
arch 依存のパッケージの構築からパッケージ作成まで実行するシーケンスです。
- binary-indep
arch 非依存のパッケージの構築からパッケージ作成まで実行するシーケンスです。
- build
構築からテストまで実行するシーケンスです。
- build-arch
arch 依存のパッケージの構築からパッケージ作成まで実行するシーケンスです。

利用可能なシーケンス名(続き)

- build-indep
arch 非依存のパッケージの構築からパッケージ作成まで実行するシーケンスです。
- clean
一度パッケージを構築したディレクトリから、パッケージ構築時に生成したものを取り除き、構築ディレクトリを綺麗にします。
- install
構築から、パッケージ生成直前までの処理を行うシーケンスです。

利用可能なシーケンス名(続き)

- install-arch
arch 依存のパッケージについて、構築から、パッケージ生成直前までの処理を行うシーケンスです。
- install-indep
arch 非依存のパッケージについて、構築から、パッケージ生成直前までの処理を行うシーケンスです。

なお、`-with foo` を指定すると、`dh` に指定可能なシーケンスが増える場合があります (例: `-with quilt` の `patch` シーケンス等。)

dh のコマンドラインオプション

- `-with addon[,addon ...]`
適切な場所で一連のコマンドを実行するような付加機能 (addon) を指定します。
- `-without addon`
`-with` とは逆の働きをします。指定された付加機能を使わないようにします。
- `-list, -l`
利用可能な付加機能一覧。

dh のコマンドラインオプション

- -no-act
指定された一連の処理の内容を表示するだけコマンドとなります。表示だけして実際にはコマンドを実行しません。
- その他
dh に、先に記載した以外の何かオプションを渡すとそれはのちに実行する全コマンドへ引き渡されます。-v、-X、-N や、他の特別なオプションを指定するのに使われます。

廃止されたコマンドラインオプション

-until, -before, -after, -remaining がありましたが、これらは全部 dh が解釈する “override_DH コマンド名ターゲット” による動作に置き換えられた為、廃止となりました。
なので、昔の debian/rule にあるような、以下の用な書き方は 廃止です。

廃止された書き方

```
#!/usr/bin/make -f
```

```
%:
```

```
dh $@
```

```
build: build-stamp
```

```
build-stamp:
```

```
dh build --before configure
```

```
dh_auto_configure -- --with-gnu-ld --disable-nls
```

```
dh build --after configure
```

```
touch build-stamp
```

override_debhelper コマンド名

dh コマンドは”dh シーケンス名”により、そのシーケンスに必要な一連の debhelper コマンドを呼び出す機能があります。
(どんな debhelper コマンドが呼び出されるかは、`-no-act` をオプションにつけて、`dh -no-act build` とか、`dh -no-act install` とかしてください)
この呼び出されるコマンドを一部変更したい場合は以下のように書きます。

今時の書き方:

```
#!/usr/bin/make -f
```

```
%:
```

```
dh $@
```

```
override_dh_autoconfigure:
```

```
dh_auto_configure -- --with-gnu-ld --disable-nls
```

override_debhelper コマンド名

この"override_debhelper コマンド名" ターゲットは、コマンドを実行したくない場合にも利用可能です。
("override_debhelper コマンド名" のアクションを空にする事がミソです。)

```
dh_auto_test,dh_compress,dh_fixperms を実行したく無い場合:  
#!/usr/bin/make -f
```

```
%:
```

```
dh $@
```

```
override_dh_auto_test override_dh_compress override_dh_fixperms
```

dh コマンドのオプション `--with addon` にて addon が提供するパッケージの作成方法を組み込む事ができます。お使いのシステムで現在どんな addon が使えるかは `dh -list` を実行すると一覧が出てきます。

```
$dh --list
bash-completion
dkms
python-central
python-support
python2
quilt
tex
$
```

(... お使いのシステムによって表示される量が変わります...)

```
$ apt-file search Debhelper/Sequence
autotools-dev: /usr/share/perl5/Debian/Debhelper/Sequence/autotools-dev
bash-completion: /usr/share/perl5/Debian/Debhelper/Sequence/bash-completion
... 中略...
sphinx-common: /usr/share/perl5/Debian/Debhelper/Sequence/sphinx-common
tex-common: /usr/share/perl5/Debian/Debhelper/Sequence/tex-common
xserver-xorg-dev: /usr/share/perl5/Debian/Debhelper/Sequence/xserver-xorg-dev
xulrunner-dev: /usr/share/perl5/Debian/Debhelper/Sequence/xulrunner-dev
$ apt-file search Debhelper/Sequence | wc -l
43
$
```

全部で43個もありますね。(debian sid で実行)

addon 複数

```
quilt 用の addon と、 autotools_dev 用の addon を併用したい時:  
#!/usr/bin/make -f  
%:  
    dh $@ --with quilt --with autotools_dev  
#    dh $@ --with quilt,autotools_dev も OK
```

例えば、`-with quilt` の場合、

- ① `dh clean` にて、`dh_clean` を呼び出す前に、`quilt` パッケージと一緒に提供している `dh_quilt_unpatch` コマンドを呼び出すようになります。
- ② `dh build` では、`dh_auto_configure` の前に `dh_quilt_patch` を呼び出すようになります。
- ③ `dh` にシーケンス名 `patch` が追加され、`dh patch` が使えるようになります。

quilt.pm の中身

```
quilt 用の addon の中身:  
#!/usr/bin/perl  
use warnings;  
use strict;  
use Debian::Debhelper::Dh_Lib;  
insert_before("dh_auto_configure", "dh_quilt_patch");  
insert_before("dh_clean", "dh_quilt_unpatch");  
# Eval to avoid problem with debhelper < 7.3.12  
eval { add_command("dh_quilt_patch", "patch");};  
1;
```

addon の API

- `insert_before($existing,$new)`
\$existing で指定される debhelper コマンドを実行する直前に \$new を実行します。
- `insert_after($existing,$new)`
\$existing で指定される debhelper コマンドを実行した直後に \$new を実行します。
- `remove_command($command)`
\$command を dh が実行しないようにします。
- `add_command($command,$sequence)`
\$sequence で示されるシーケンスで実行されるコマンド群の最後に \$command を付け加えます。また、本 API を使ってシーケンスを新たに作成することができます。

addon のAPI(続き)

- `add_command_options($command,@options)`
\$command に、配列@options で示される一連のオプションを付け加えて実行するようにします。
- `remove_command_options ($command,@options)`
\$command から配列@options で示される一連のオプションを取り除く。@options をまったく指定せずに `remove_command_options($command)` と呼び出すと、\$command についてのオプション全部を取り除きます。

月刊 Deb 専 2011 年 12 月号 pp.21 あたりの図 2 参照



“debian/パッケージ名.debhelper.log” ファイルについて

最近の dh コマンドを使う debian/rules には、ファイルの依存関係についての記載がありません。この為、パッケージビルド中で処理が中断した場合、どこから再開すれば良いかを debian/rules で make が判定する事はできません。そのため、代わりに“debian/パッケージ名.debhelper.log” に記録を残しリジュームします。

```
debian/パッケージ名.debhelper.log の中身:  
dh_auto_test  
dh_prep  
dh_installdirs  
... 中略...  
dh_buiddeb
```

“debian/パッケージ名.debhelper.log” ファイルについて

注意: なお、処理再開の場所は、このログファイルのみ参照して決める為、処理を中断した後に、パッケージのソースファイルを変更して再開させるような使い方はできません。例えば、ソースファイル中のあるファイルを変更した為、特定のパッケージのシーケンスについては再会時に全部やり直しが必要だったとしても、これを自動で検知することはできません。

そりゃそうだ。

dpkg-buildflag との統合 (v9)

互換性度合い (COMPATABILITY LEVEL) に v9 指定すると、dh は内部で dpkg-buildflags 相当の処理を呼び出してコンパイル時の環境変数を指定できるようになります。なので、呼び出される debhelper 時に設定される環境変数は、

- ① /etc/dpkg/buildflags.conf の中身
- ② XDG_CONFIG_HOME/dpkg/buildflags.conf (XDG_CONFIG_HOME は環境変数です) の中身
- ③ HOME/.config/dpkg/buildflags.conf (HOME は環境変数です) の中身
- ④ DEB_flag_MAINT_SET, DEB_flag_MAINT_STRIP, DEB_flag_MAINT_APPEND, DEB_flag_MAINT_PREPEND, DEB_BUILD_MAINT_OPTS (全部環境変数です) の値

により様々に変化します。

dpkg-buildflag との統合 (v9)

```
$ dpkg-buildflags
CFLAGS=-g -O2 -fstack-protector --param=ssp-buffer-size=4
-Wformat -Wformat-security -Werror=format-security
CPPFLAGS=-D_FORTIFY_SOURCE=2
CXXFLAGS=-g -O2 -fstack-protector --param=ssp-buffer-size=4
-Wformat -Wformat-security -Werror=format-security
FFLAGS=-g -O2
LDFLAGS=-Wl,-z,relro
$ env DEB_CFLAGS_MAINT_SET='-O3 -Wall -pedantic' dpkg-buildflags
CFLAGS=-O3 -Wall -pedantic
CPPFLAGS=-D_FORTIFY_SOURCE=2
CXXFLAGS=-g -O2 -fstack-protector --param=ssp-buffer-size=4
-Wformat -Wformat-security -Werro
r=format-security
FFLAGS=-g -O2
LDFLAGS=-Wl,-z,relro
$
```

CFLAGS が変化している事に注目

今月のコマンド: `dh_testroot`

- 動作概要

現在の実行ユーザが `root` であるかどうかを確認するコマンドです。 `root` ユーザでは無い場合、エラーメッセージを出力して処理を中断します。

- コマンドラインオプション

コマンドラインオプションは特にありません。何か指定しても無視されます。

dh_testroot を実行してみる

```
$ sudo dh_testroot
$ echo $?
0
$ dh_testroot
You must run this as root (or use fakeroot).
$ echo $?
255
$ fakeroot dh_testroot
$ echo $?
0
```

このように root 権限で実行するか、fakeroot 経由で実行した時のみ 0 を返却します。

man 文章査読のお願い


debhelper の man 文章の po4a 日本語訳しました。

(中間報告として、1回100%まで訳したものを
debian-doc@d.o.j に流した)

まだ粗い訳なので、修正したらも一度ながすので、だれか査読
手伝ってーっ


次回発表者

次回の幸運な発表者は、



ばばーん

次回発表者



山田さんですー



今後のイベント

今後のイベント

- 12 月度 HackCaffe
12/26(月)19:00-21:00
<http://atnd.org/events/22786>
https://twitter.com/debian_hackcafe
- 1 月 第 84 回 東京エリア Debian 勉強会

ステーキハウス TEXAS

<http://r.gnavi.co.jp/g083522/>