

.Deb

銀河系唯一のDebian専門誌

2014年07月19日

特集： Jenkins での Debian パッケージ自動化



Debian 勉強会

	2	Debian Trivia Quiz	3
目次			
1	事前課題	2	
1.1	koedoyoshida	2	
1.2	dictoss(杉本 典充)	2	
1.3	工藤寿晴	2	
1.4	吉野 (yy-y-ja-jp)	2	
1.5	zinrai	2	
1.6	wbchsyn	2	
1.7	鈴木崇文	2	
1.8	野島 貴英	2	
1.9	岩松 信洋	2	
1.10	まえだこうへい	2	
		3	最近の Debian 関連のミーティング報告
		3.1	東京エリア Debian 勉強会 114 回目報告
		4	Jenkins での Debian パッケージ自動化
		4.1	背景
		4.2	自動化するプロセス
		4.3	Jenkins 及び reprepro 環境の構築
		4.4	まとめ
		5	索引

1 事前課題

野島 貴英



今回の事前課題は以下です:

1. 本日、何の作業をやるかを宣言ください。

この課題に対して提出いただいた内容は以下です。

1.1 koedoyoshida

未定

1.2 dictoss(杉本 典充)

パッケージ作成に関する作業は、<http://wnpp.debian.net/> に報告が出ているパッケージ化、またはパッチづくりをやると思っています。具体的には munin、pygtkplot、pylzma、wx3.0-doc、hex2bin が候補です。

1.3 工藤寿晴

自分が Vine で管理しているパッケージから、何か Debian 用のパッケージを作ります。(フォントか CUI のソフトウェアを検討中)

1.4 吉野 (yy-y-ja-jp)

manpages-ja

1.5 zinrai

パッケージ作成
pulledpork
<https://code.google.com/p/pulledpork/>

1.6 wbcchsyn

GitHub で公開中の自作プログラムをパッケージする予定 (先月の勉強会で発表した物です。)

<https://github.com/wbcchsyn/dVault>

時間があまったら、プログラムの機能実装を行います。

1.7 鈴木崇文

Cocot のパッケージングをやります。

1.8 野島 貴英

パッケージング道場参加します。

- xmris のパッケージング
- kanatest のバグつぶし
- Softether のパッケージ化ヒントが得られるまでとなっています。

1.9 岩松 信洋

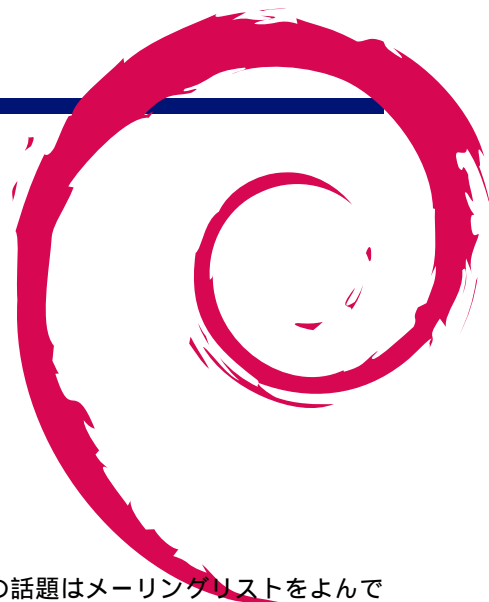
パッケージアップロードします。

1.10 まえだこうへい

- ToDo 潰し
<http://d.palmtb.net/pages/about.html#todo-from-maintainer-dashboard>
- pending のままの Serf 関連のパッケージ化を行う予定。

2 Debian Trivia Quiz

野島 貴英



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` や `debian-devel@lists.debian.org` に投稿された内容などからです。

問題 1. `libc` が変更になりました。変更前と変更後の組み合わせで正しいのはどれ？

- A `eglibc` から `glibc` へ変更
- B `glibc` から `eglibc` へ変更
- C `uClibc` から `glibc` へ変更

問題 2. ネットワークは `Tor` を使い、ファイル/メール/メッセージャソフトは暗号化対応という特徴を持つ `debian` ベースの Live OS の名前は何でしょう？

- A `Plan9`
- B `Tails`
- C `GNU/Hurd`

問題 3. `Debian 6 LTS` 開発の為に寄付を受け付ける会社名は？

- A `Canonical`
- B `SquareEnix`
- C `Freexian`

問題 4. `jessie` のデフォルトの `gcc` のバージョンは何になった？

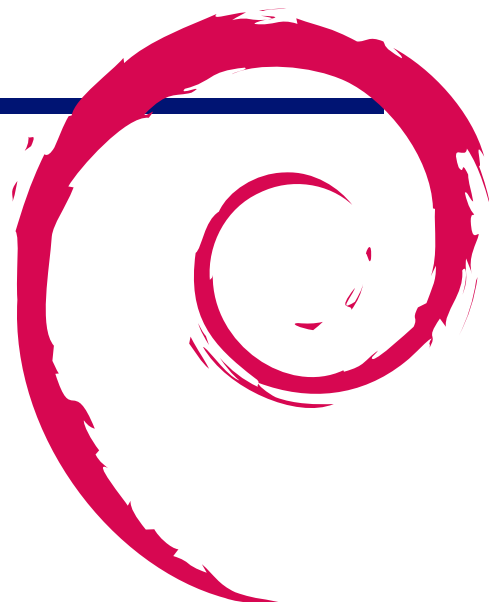
- A 4.9
- B 4.6
- C 2.6

問題 5. `jessie` の freeze は 11 月 5 日です（注：リリースじゃないよ。）ここで、スケジュールで新規の transition が出来なくなる日はいつでしょう？

- A 10 月 5 日
- B 9 月 5 日
- C 8 月 5 日

3 最近の Debian 関連のミーティング報告

野島 貴英

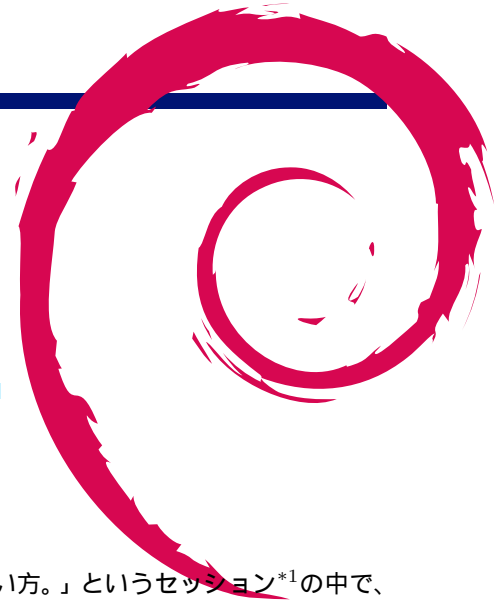


3.1 東京エリア Debian 勉強会 114 回目報告

東京エリア Debian 勉強会 114 回目は (株) スクウェア・エニックスさんで開催されました。

9 人の参加者がありました。また、wbccsyn さんにて、GPG の秘密鍵を堅牢にかつセキュアに管理する事を検討した成果についての発表がありました。今回は、その成果の 1 つとして、堅牢化についてフォーカスし、RAID の考え方をういて複数の記憶メディアにデータを分散配置して復元するソフトウェアを python で実装した事についての発表でした。また、こちらについて様々に意見交換が参加者の間で行われました。

発表者自身のアイデアに基づく Debian に関係した成果物の発表は勉強会としても大変価値があると考えております。引き続き、発表者の方々によるユニークなアイデアと成果の発表が期待されます。



4 Jenkins での Debian パッケージ自動化

前田耕平

昨年の大統一 Debian 勉強会 2013 で、「とある企業での Debian システムの使い方。」というセッション^{*1}の中で、社内向けの Debian パッケージの作成と、reprepro を使ったパッケージの配布方法についての話をしました。今回はその続編です。

4.1 背景

管理するパッケージが少ない場合は、手作業でのビルドでも間に合っていました。管理するパッケージ数が増え、対象ディストリビューションが増え (Precise, Wheezy, Trusty)、一方メンテナンスする人を増やすのは時間が掛かる、という状況でした。

そこで、別件^{*2}で Jenkins を用意したこともあったので、ソースパッケージを作成するプロセス以外は、全て Jenkins で実行して自動化することにしました。

4.2 自動化するプロセス

パッケージのビルドからローカルアーカイブへの登録までに行うプロセスは図 1 のようになります。1a は公式パッケージのソースパッケージを dget など取得する場合、1b は独自にソースパッケージを作成する場合です。ビルドする対象のソースパッケージによって実際的手段自体は異なってきますが、実行するプロセス自体は図の通り変わりません。

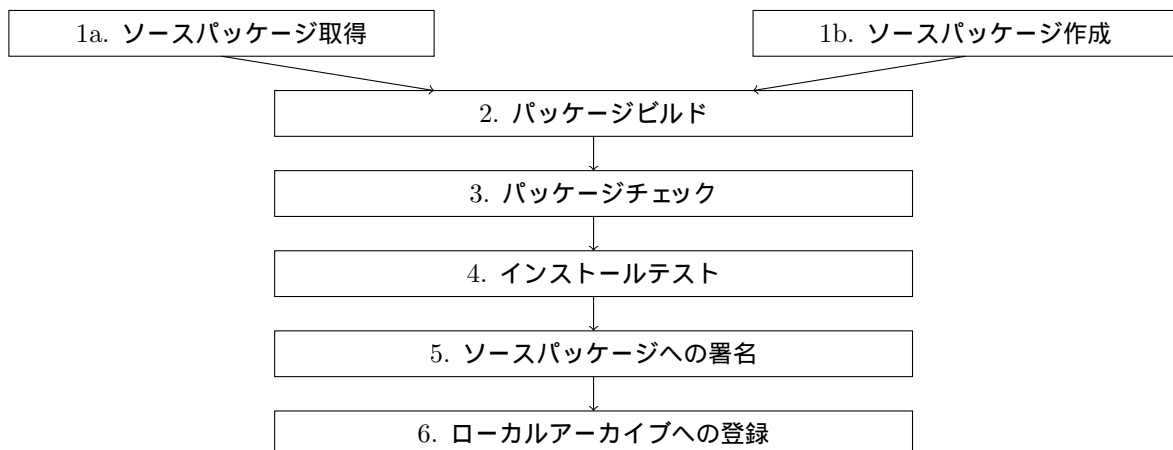


図 1 パッケージビルド～ローカルアーカイブへの登録までのプロセス

^{*1} <http://gum.debian.or.jp/2013/session/437.html>

^{*2} Java や Python のプロジェクトのビルドやテストの実行目的。

各プロセスの実行内容は下記ようになります。

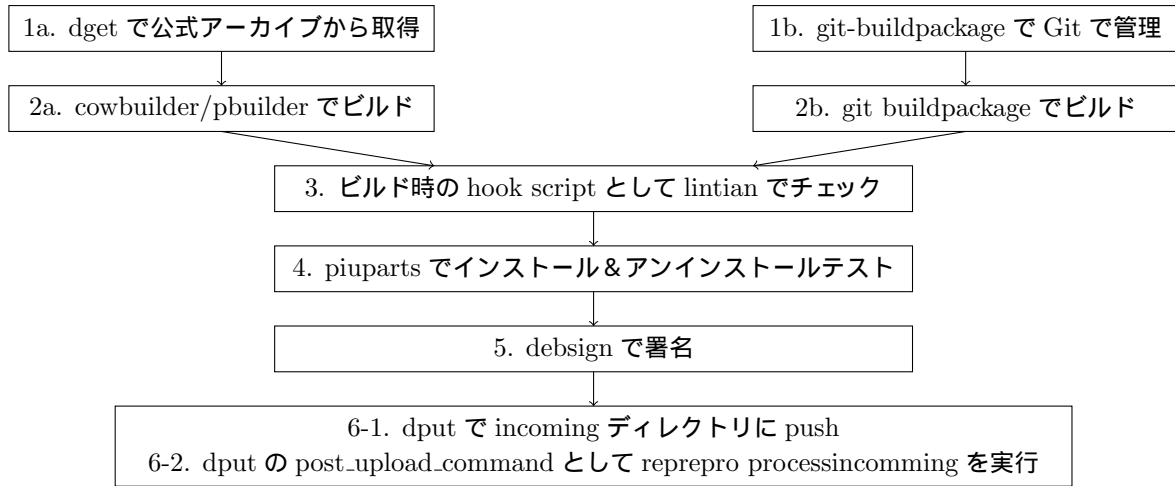


図 2 各プロセスの実行内容

これらを Jenkins でジョブとして実行するためには、対話形式の処理をなくし、リモートホスト上の処理も実行可能な状態で、一連の処理を Jenkins のジョブスクリプトを作成する必要があります。

4.2.1 Jenkins で実行する上での課題

Jenkins で実行する上で問題となるのは、GPG と SSH 公開鍵のパスフレーズの入力と、reprepro でのアーカイブの登録です。

reprepro でのアーカイブ登録は `reprepro processincoming` コマンドで行います。Jenkins と別ホストで `reprepro` を実行する場合、`dput.cf` の `post_upload_command` を使えば解決できます。例えば、Ubuntu Trusty 用のローカルアーカイブに `dput` する場合は、下記のような設定を `dput.cf` にしておきます。

```
[trusty]
fqdn      = repo.example.org
method    = scp
incoming  = /var/lib/debpkg-custom/ubuntu/incoming/
login     = repoadm
post_upload_command = ssh repoadm@repo.example.org /usr/bin/reprepro -b /var/lib/debpkg-custom/ubuntu \
--keepunreferencedfiles processincoming trusty #実際は一行
```

公開鍵の入力は、次の 3 箇所が必要になります。^{*3}

- GPG
 - 5. `debsign` での署名
 - 6-2. `dput` の `post_upload_command` として `reprepro processincoming` を実行
- SSH
 - 6-1. `dput` で `incoming` ディレクトリに `push`

一方、GPG については既存環境で使っている鍵をそのまま利用する関係上、避けて通れません。Jenkins のジョブスクリプトで

```
$ echo -e 'passphrase\npassphrase\n' | debsign some.changes
$ echo -e 'passphrase\npassphrase\n' | dput some.changes
```

と実行しても標準入力からはパスフレーズは渡せません。しかし、`gpg` コマンド自体には `--no-tty --batch` オプションを使って、対話形式でなくてもパスフレーズを入力することができます。そのため、方法としては、

^{*3} Jenkins と `reprepro` を同じホストで実行する場合 (つまり一台で完結させる場合) には SSH 公開鍵のパスフレーズ入力は特に考慮することはありません。

- debsign, reprepro processing を TTY なしで対話形式でなくても GPG パスフレーズを入力できるようにする
- expect を使ってプロンプトで判断してパスフレーズを入力させる

の 2 つの方法が考えられます。で、先に着手した debsign については、expect で処理するのは負けだと思ったので、別途 pydebsign という Python ライブラリを実装しました*4。しかし、reprepro については時間の都合上、結局 Python 版の expect の pexpect を使いました。下記のようになります。*5

```
def upload_files(codename, changes_path, reprepro_passphrase):
    """ uploading to local archive incoming directory of reprepro """
    os.environ['LANG'] = 'C'
    command = "dput %s %s" % (codename, changes_path)
    exp = pexpect.spawn(command, maxread=4000)
    exp.logfile_read = sys.stdout
    exp.expect('Please enter passphrase:')
    exp.sendline(reprepro_passphrase)
    exp.expect('Please enter passphrase:')
    exp.sendline(reprepro_passphrase)
    exp.expect(pexpect.EOF)
    exp.close()
    return exp.status
```

4.2.2 Jenkins で必要なプラグイン

パスフレーズは渡せるようになりましたが、Jenkins のジョブを実行するとパスフレーズが丸見えです。そのため、"Mask Passwords Plugin"*6 というプラグインを使います。他に必要なプラグインは下記の通りです。また、Jenkins のジョブスクリプト管理用と、git-buildpackage 用と複数の Git リポジトリを扱えるようにするため、"Git Plugin"*7 と "Multiple SCMs Plugin"*8 も必要です。これらのプラグインは後述の Jenkins 環境構築後に、Jenkins の管理画面からインストールします。

4.2.3 その他の考慮点

- Jenkins で自動ビルド & ローカルアーカイブへ登録したパッケージに依存するパッケージを作成することもあります。そのため、後述の pbuilder と cowbuilder のイメージの APT line には、ローカルアーカイブの URL を登録し、ローカルアーカイブの公開鍵を apt-key で登録しておく必要があります。
- バックポートする場合、orig.tar.gz,xz が changes ファイルのリストにはデフォルトでは登録されず、reprepro で登録する際にエラーになります。そのため、debbuildopts='-sa' を指定する必要もあります。
- pbuilder や cowbuilder でパッケージをビルドすると、デフォルトでは /var/cache/pbuilder/results ディレクトリ以下に作成されます。このままでは、Jenkins のジョブごとに処理するには不都合です。pbuilderrc で BUILDRESULT に、Jenkins の workspace 以下のパスを指定するようにします。

以上の点を考慮し、Python で Jenkins ジョブ用のスクリプトを作成しました。*9 都合により、Golang 版も作成しました。*10 これらは、前述の hook script で行う lintian でのテストを、dput の -l オプションで実行できるようにしています。*11

4.3 Jenkins 及び reprepro 環境の構築

Jenkins で自動実行するための環境として、下記の準備が必要になります。

*4 Git: <https://github.com/mkouhei/pydebsign>, PyPI: <https://pypi.python.org/pypi/pydebsign>

*5 debsign のこれで良かったですね。。。

*6 <https://wiki.jenkins-ci.org/display/JENKINS/Mask+Passwords+Plugin>

*7 <https://wiki.jenkins-ci.org/display/JENKINS/Git+Plugin>

*8 <https://wiki.jenkins-ci.org/display/JENKINS/Multiple+SCMs+Plugin>

*9 デモのみ。

*10 <https://github.com/mkouhei/godebbuild>

*11 backport などの場合、lintian の Warning, Error を修正するためのパッケージの変更せず、許容してしまいたいこともあるため。

- Jenkins のインストール&設定
- Jenkins 用ユーザへの sudo 設定
 - /usr/sbin/pbuilder
 - /usr/sbin/cowbuilder
 - /usr/bin/git-buildpackage
 - /usr/bin/piuparts
 - /bin/chown
- pbuilder and cowbuilder イメージの作成
 - ビルド対象のディストリビューションの数分だけ作成
- debsign および reprepro 用の GPG 鍵の作成
- reprepro への upload 用の SSH 公開鍵の作成^{*12}
- dput.cf の設定
- rerepro のインストール&設定
- reprepro 管理 pool の公開設定
- ローカルアーカイブ用の GPG 公開鍵の export

このうち、pbuilder と cowbuilder イメージの作成の一部と、GPG 鍵の作成、SSH 公開鍵の作成については、手動で行う必要がありますが、他は Ansible で自動化しました。^{*13}

4.4 まとめ

ソースパッケージの作成と Jenkins のジョブの作成以外は自動化できました。これにより、Debian パッケージの作成をやったことが無い人でもバックポートなどはできるようになりました。しかし、実際にはジョブを実行するとパッケージの依存関係や、lintian、piuparts などのエラーが発生します。そのため、Debian パッケージ作成について知らなければ、それらのエラーに対応できないことが予想されます。

また、ソースパッケージの作成は相変わらず、属人化しています。^{*14}なので、今回の Debian パッケージング道場でパッケージ作成ができる人が増えるとイイですね。^{*15}

^{*12} Jenkins と reprepro が別ホストの場合

^{*13} デモ参照。Playbook は、<https://github.com/mkouhei/playbook-tokyodebian115> に用意してあります。

^{*14} つまり、職場では自分しか作成できない。

^{*15} 原稿作成時点では職場の同僚の参加者は未定。

5 索引

building-packages-automatically-on-jenkins, 5





Debian 勉強会資料

2014年07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）
