

# Grand Unified Debian

💉 銀河系唯一のDebian専門誌





あんどきゅめんてっど でびあん 2016 年冬号 2016 年 12 月 29 日 初版発行



# 目次

1	Introduction	2
2	Debian で lxc をセットアップしてみよう	3
3	preseed で Debian を自動インストールをしてみよう	9
4	Let's Encrypt のススメ	14
5	OpenFOAM で数値流体解析	17
6	Debian パッケージの" 説明文 " について	23
7	go-apt-cacher/mirror	31
8	sbuild と debci を触ってみた	33
9	debexpo(mentors.d.n) をハックするには	37
10	dh_strip_nondeterminism について	55
11	DEP5/Machine-readable debian/copyright 再考	62
12	Debian Trivia Quiz	68
13	Debian Trivia Quiz 問題回答	71

## 1 Introduction

DebianJP

## 1.1 東京エリア Debian 勉強会

Debian 勉強会へようこそ。これから Debian の世界にあしを踏み入れるという方も、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?

Debian 勉強会の目的は下記です。

- Debian Developer (開発者)の育成。
- 日本語での"開発に関する情報"を整理してまとめ、アップデートする。
- ●場の提供。
  - 普段ばらばらな場所にいる人々が face-to-face で出会える場を提供する。
  - Debian のためになることを語る場を提供する。
  - Debian について語る場を提供する。

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。情報の共有・活用を通して Debian の今後の能動的な展開への土台として、"場"としての 空間を提供するのが目的です。

## 1.2 関西 Debian 勉強会

関西 Debian 勉強会は Debian GNU/Linux のさまざまなトピック (新しいパッケージ、Debian 特有の機能の仕組、Debian 界隈で起こった出来事、などなど)について話し合う会です。

目的として次の三つを考えています。

- ML や掲示板ではなく、直接顔を合わせる事での情報交換の促進
- 定期的に集まれる場所
- 資料の作成

それでは、楽しい一時をお楽しみ下さい。



杉本 典充

## 2.1 はじめに

コンピュータの仮想化技術にコンテナ型仮想化という仕組みがあります。本発表では、コンテナ型仮想化技術の1 つである lxc について説明します。

## 2.2 仮想化技術について

#### 2.2.1 仮想化技術の分類

コンピュータにおける仮想化技術にはいくつかの種類があります。

- コンテナ型仮想化
  - ゲスト環境はホスト環境のあるディレクトリに配置した実行ファイルおよびライブラリの集合であり、実 行中のゲスト環境はホスト環境から見ると単なるプロセス群である。実装例は OpenVZ、LXC。
- 準仮想化型
  - ホスト環境とやりとりする API を利用できるように修正が入った OS をゲスト環境として動作させる方式
     (=既存の OS そのままでは動かない)。実装例は Xen。
- 完全仮想化型(エミュレーション型)
  - 既存の OS を無修正のままゲスト環境として動作させる。ホスト環境上で動作する仮想化アプリケーショ
  - ンがゲスト環境をエミューションする。実装例は QEMU、VirtualBox。
- 完全仮想化型 (ハイパーバイザ型)
  - 既存の OS を無修正のままゲスト環境として動作させる。CPU の仮想化機能を使うことでホスト環境に おけるオーバーヘッドを極力減らし、エミュレーション型より高いパフォーマンスを出せる。実装例は KVM。

#### 2.2.2 仮想化技術のメリットとデメリット

準仮想化型および完全仮想化型におけるゲスト環境は特定のハードウェアをもつ物理マシンをエミュレートするいわゆる「仮想マシン」を実行の単位とします。そのため仮想マシン上でもカーネルを実行させる必要があり CPU、メモリ、ディスクを多く使用します。その代わり、物理マシンへ普通に OS をインストールして実行していたプログラムをそのまま実行できる場合が非常に多いことが優位です。

コンテナ型仮想化におけるゲスト環境は、ホスト環境のカーネルで動作するプロセス群であるため、余分なカーネ ルや管理デーモンを動作させる必要がなく省リソースで動作できる点が優位です。その代わり、ホスト環境の OS や 設定によってゲスト環境に動作制約がつくことがあります\*1。

#### 2.2.3 コンテナ型仮想化技術の基本 chroot

コンテナ型仮想化の基礎技術に chroot があります<sup>\*2</sup>。あるディレクトリ配下に実行ファイル、ライブラリ、設定 ファイルを適切に配置した rootfs(=コンテナ環境) に対して chroot(2) または chroot(3) を実行することで、コンテ ナ環境内のライブラリを使ってコマンドを実行できます。

#### 2.3 Ixc 解説

2.3.1 lxc とは

lxc とは<sup>\*3</sup>、コンテナ型仮想化として動作する単位であるゲスト環境 (=rootfs) の配置、起動、終了、コンソール入 出力、ネットワーク等を管理する仕組みです。lxc を使うことによって、コンテナ環境を起動し、IP アドレスを割り 当て、あたかもそこに1台の仮想マシンが起動したかのように振る舞うことができます。

#### 2.3.2 lxc のインストール

今回は Debian GNU/Linux 8 Jessie の amd64 上で lxc をインストールして動作させてみます。Debian Project では以下にドキュメントがまとまっています。

- https://wiki.debian.org/LXC
- https://wiki.debian.org/LXC/SimpleBridge

#### まず、lxc パッケージと仮想ネットワークを構築するパッケージをインストールします

```
# apt-get install lxc bridge-utils libvirt-bin
```

#### lxc をインストールすると、lxc-\*コマンドが使えるようになります。

<pre>\$ ls /usr/bin/lxc*</pre>		
/usr/bin/lxc-attach	/usr/bin/lxc-start	/usr/bin/lxc-test-list
/usr/bin/lxc-autostart	/usr/bin/lxc-start-ephemeral	/usr/bin/lxc-test-locktests
/usr/bin/lxc-cgroup	/usr/bin/lxc-stop	/usr/bin/lxc-test-lxcpath
/usr/bin/lxc-checkconfig	/usr/bin/lxc-test-apparmor	/usr/bin/lxc-test-may-control
/usr/bin/lxc-clone	/usr/bin/lxc-test-attach	/usr/bin/lxc-test-reboot
/usr/bin/lxc-config	/usr/bin/lxc-test-autostart	/usr/bin/lxc-test-saveconfig
/usr/bin/lxc-console	/usr/bin/lxc-test-cgpath	/usr/bin/lxc-test-shutdowntest
/usr/bin/lxc-create	/usr/bin/lxc-test-clonetest	/usr/bin/lxc-test-snapshot
/usr/bin/lxc-destroy	/usr/bin/lxc-test-concurrent	/usr/bin/lxc-test-startone
/usr/bin/lxc-device	/usr/bin/lxc-test-console	/usr/bin/lxc-test-symlink
/usr/bin/lxc-execute	/usr/bin/lxc-test-containertests	/usr/bin/lxc-unfreeze
/usr/bin/lxc-freeze	/usr/bin/lxc-test-createtest	/usr/bin/lxc-unshare
/usr/bin/lxc-info	/usr/bin/lxc-test-destroytest	/usr/bin/lxc-usernsexec
/usr/bin/lxc-ls	/usr/bin/lxc-test-device-add-remove	/usr/bin/lxc-wait
/usr/bin/lxc-monitor	/usr/bin/lxc-test-get_item	
/usr/bin/lxc-snapshot	/usr/bin/lxc-test-getkeys	

libvirt-bin をインストールして libvirtd を起動すると、デフォルトで 192.168.122.0/24 の NAT ネットワー クが構成されます。今回は作成したコンテナ環境をこの NAT ネットワークに接続し、ホスト環境とつながる 192.168.122.1(=virbr0) をデフォルトゲートウェイとして通信できるようにします。<sup>\*4</sup>

次に、lxc で動作するコンテナ環境にリソース制約を行う仕組みである cgroups を設定します。

<sup>\*1</sup> raw ソケットが利用できない、共有メモリが利用できない、loopback が利用できない、同一ホスト環境で動作している他のコンテナと通 信できてしまう、などの制約がある場合があります。

<sup>\*2</sup> chroot システムコールは 1982 年にウィリアム・ネルソン・ジョイ (ビル・ジョイの名で知られている) が開発したことを起源とされていま す。ビル・ジョイはプログラムをクリーンビルドできる環境がほしかったため通常利用の環境と分離する手段として開発したと言われてい ます。

<sup>\*&</sup>lt;sup>3</sup> LinuX Containers を省略して lxc と読んでいます。

<sup>\*4</sup> 仮想ネットワークはノートパソコンや VPS で構築する場合は NAT の方がよいと思いますが、自宅や社内のネットワークで構築する場合 はブリッジ接続 (=ホスト環境とコンテナ環境が同一ネットワークに属する形態) を利用するほうがよい場合もあります。

```
# vi /etc/fstab
cgroup /sys/fs/cgroup cgroup defaults 0 0
# mount /sys/fs/cgroup
# mount | grep cgroups
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,
release_agent=/lib/systemd/systemd-cgroups-agent,name=systemd)
```

この状態で、lxc を使える環境であるか確認するコマンド"lxc-checkconfig"を実行してみます。「enabled」と書か れているところは、実際のターミナル上では緑色で表示されます。disableの機能はありませんので、動作できる環境 であることが確認できました。

# lxc-checkconfig
Kernel configuration not found at /proc/config.gz; searching
Kernel configuration found at /boot/config-3.16.0-4-amd64
Namespaces
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled
Control groups
Cgroup: enabled
Cgroup clone_children flag: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: enabled
Cgroup cpuset: enabled
Misc
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
File capabilities: enabled
Note : Before booting a new kernel, you can check its configuration
usage : CONFIG=/path/to/config /usr/bin/lxc-checkconfig

#### 2.3.3 コンテナの作成:lxc-create

ゲスト環境となるコンテナ環境 (=rootfs) は lxc-create コマンドで作成することができます。今回の lxc のコンテ ナ環境は、ホスト環境と同じ Debian GNU/Linux 8 Jessie amd64 を作成します<sup>\*5</sup>。今回は、"debstudy1" という名 前でコンテナを作成します。

```
# LANG=C SUITE=jessie MIRROR=http://ftp.jp.debian.org/debian lxc-create -n debstudy1 -t debian
debootstrap is /usr/sbin/debootstrap
Checking cache download in /var/cache/lxc/debian/rootfs-jessie-amd64
Copying rootfs to /var/lib/lxc/debstudy1/rootfs...Generating locales (this might take a while)...
Generation complete.
insserv: warning: current start runlevel(s) (empty) of script 'checkroot.sh' overrides LSB defaults (S).
insserv: warning: current stop runlevel(s) (S) of script 'checkroot.sh' overrides LSB defaults (empty).
insserv: warning: current stop runlevel(s) (S) of script 'checkroot.sh' overrides LSB defaults (empty).
insserv: warning: current start runlevel(s) (empty) of script 'checkroot.sh' overrides LSB defaults (S)
update-rc.d: error: umountfs Default-Start contains no runlevels, aborting.
insserv: warning: current start runlevel(s) (empty) of script 'hwclock.sh' overrides LSB defaults (S).
insserv: warning: current stop runlevel(s) (0 6 S) of script 'hwclock.sh' overrides LSB defaults (0 6).
update-rc.d: error: cannot find a LSB script for hwclockfirst.sh
Creating SSH2 RSA key; this may take some time ...
2048 df :99:56:34:c7:66:41:02:24:c2:b4:62:f5: /etc/ssb/ssb host rsa key pub (BSA)
                                                                                                                                       overrides LSB defaults (S).
2048 df:99:56:34:c7:6d:d1:0a:2d:e2:b4:6a:fd:a0:62:f5 /etc/ssh/ssh_host_rsa_key.pub (RSA)
Creating SSH2 DSA key; this may take some time
1024 90:42:45:1d:fd:03:92:04:6c:e0:fb:e6:06:cc:07:06 /etc/ssh/ssh_host_dsa_key.pub (DSA)
Creating SSH2 ECDSA key; this may take some time ...
256 6a:4a:1a:6f:27:59:33:6c:58:5c:58:27:03:08:3b:ea /etc/ssh/ssh_host_ecdsa_key.pub (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 36:d6:9b:d3:9d:96:a4:af:af:8c:75:11:90:76:56:75 /etc/ssh/ssh_host_ed25519_key.pub (ED25519)
Failed to read /proc/cmdline. Ignoring: No such file or directory invoke-rc.d: policy-rc.d denied execution of start.
Current default time zone: 'Asia/Tokyo'
                                          Sun Jul 10 13:26:07 JST 2016.
Local time is now:
Universal Time is now: Sun Jul 10 04:26:07 UTC 2016.
Root password is 'Won4EiUa', please change !
```

lxc-create を実行すると、"/var/lib/lxc/debstudy1" というディレクトリが作成され、その配下に rootfs と lxc の ゲスト環境用設定ファイルが生成されます。なお、一度コンテナを作成するとダウンロードした rootfs ファイルは

<sup>\*&</sup>lt;sup>5</sup> ゲスト環境は、ホスト環境の linux カーネルで動作するバイナリであればホスト環境と異なるディストリビューションでも動作可能です。

キャッシュされます。(2回目以降の作成でDebian パッケージのダウンロードが不要になり早く処理が終わります)

# ls -l /var/lib/lxc/debstudy1
合計 8
-rw-r--r-- 1 root root 479 7月 10 13:26 config
-rw-r--r-- 1 root root 0 7月 10 13:26 fstab
drwxr-xr-x 22 root root 4096 7月 10 13:26 rootfs
# ls /var/lib/lxc/debstudy1/rootfs
bin boot dev etc home lib lib64 media mnt opt proc root run sbin selinux srv sys tmp usr var

lxc が参照する設定ファイルの初期状態は以下になります。

```
# cat /var/lib/lxc/debstudy1/config
# Template used to create this container: /usr/share/lxc/templates/lxc-debian
# Parameters passed to the template:
# For additional config options, please look at lxc.container.conf(5)
lxc.network.type = empty
lxc.rootfs = /var/lib/lxc/debstudy1/rootfs
# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf
# Container specific configuration
lxc.mount = /var/lib/lxc/debstudy1/fstab
lxc.utsname = debstudy1
lxc.arch = amd64
lxc.autodev = 1
lxc.kmsg = 0
```

この config ファイルにコンテナ環境が利用するネットワーク設定を行います。

```
# vi /var/lib/lxc/debstudy1/config
(末尾に以下を追加)
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = virbr0
lxc.network.name = eth0
lxc.network.ipv4 = 192.168.122.203/24
lxc.network.ipv4.gateway = 192.168.122.1
```

## 2.3.4 コンテナの一覧:lxc-ls

lxc-ls コマンドで作成したコンテナの一覧を表示できます。

```
# lxc-ls
debstudy1
```

#### 2.3.5 コンテナの削除:lxc-destroy

lxc-destroy コマンドでコンテナを削除できます。

# lxc-destroy -n <lxc-name>

#### 2.3.6 コンテナの起動:lxc-start

lxc-start コマンドでコンテナを起動します。コンテナを起動すると、コンテナ環境内にある init プログラムが実行 され常駐します。

-d オプションはコンテナ環境をバックグラウンドで起動するオプションです。-d オプションを付けない場合、 lxc-startを実行したターミナルはコンテナ環境のコンソールに接続されます。

```
# lxc-start -n debstudy1
または
# lxc-start -n debstudy1 -d
```

#### 2.3.7 コンテナの終了:lxc-stop

lxc-stop コマンドでコンテナを終了します。コンテナを終了すると、init プログラムが起動中のデーモンを終了させ、init プログラム自身も終了します。

# lxc-stop -n debstudy1

#### 2.3.8 コンテナのコンソールへ接続する:lxc-console

lxc-console コマンドでコンテナ環境のコンソールへ接続します。 "Ctrl+a q" の順にキー入力すると、コンソールから抜けることができます。

```
# lxc-console -n debstudy1
Connected to tty 1
```

```
Connected to tty 1
Type <Ctrl+a q> to exit the console, <Ctrl+a Ctrl+a> to enter Ctrl+a itself
Debian GNU/Linux 8 debstudy1 tty1
debstudy1 login:
```

## 2.4 lxc を実用する

#### 2.4.1 lxc のゲスト環境をを利用可能な状態までセットアップする流れ

lxcのゲスト環境へのログインはsshでログインしたいと思うことが多いと考えます。そのため、lxcのゲスト環境 を作成して動作させるまでのセットアップの流れの一例をまとめてみます。

なお、debian の場合は lxc-create したときの rootfs 内に openssh-server パッケージもインストールされます。

- lxc-create コマンドでゲスト環境を作成する
- config ファイルを修正し、IP アドレス付与及びネットワーク設定を行う
- chroot コマンドで rootfs へ直接入る (または、lxc-start でコンテナ環境を起動してコンソールへ接続する)
  - passwd コマンドで root パスワードを書き換える
  - adduser コマンドでユーザを作成する
  - apt-get install sudo vim-tiny
  - visudo
  - dpkg-reconfigure locales (初期値は lxc-create したときに LANG 値となります)
- lxc-start -n {lxc-name} -d を実行し、バックグラウンドでゲスト環境を起動
- ホスト環境から ssh コマンドでゲスト環境の IP アドレスとユーザを指定してログインする

ssh ログインができ、sudo が利用できるようになれば後はお好みで設定ができると思います。

#### 2.4.2 何に lxc を使うか

lxc をどのような場面で利用するかはその人次第です。自分は以下の用途で利用しています。

- 一時的な検証で、ホスト環境にいろいろインストールしたくない場合(例:デーモンの再起動が必要になる)
- python2 系と python3 系が混在した複数の web アプリケーションを 1 つのホストで動かしたい場合 (debian の場合、libapache2-mod-wsgi と libapache2-mod-wsgi-py3 は共存できない)
- ホスト環境は systemd のままで、ゲスト環境は sysvinit で動作させたい場合 (systemd に対応しないプログラムを利用する苦肉の策)
- 開発したアプリケーションをクリーンな環境でビルドやインストールできるかテストを行う場合\*6\*7
- amd64 上で i386 環境がほしい、または異なる CPU アーキテクチャのコンテナ環境が動作するクロス環境が ほしい場合\*<sup>8</sup>

## 2.5 **おわりに**

Debian GNU/Linux 上で lxc を試してみました。lxc は、複数ホストの lxc 環境を制御する LXD や docker の基礎 技術であるため、皆さま試してみてください。

## 2.6 参考文献

- <sup>r</sup>LXC J https://linuxcontainers.org/
- <sup>r</sup>LXC Debian Wiki J https://wiki.debian.org/LXC
- 杉本 典充 (2013)「debootstrap を有効活用してみよう」http://tokyodebian.alioth.debian.org/pdf/ debianmeetingresume201304.pdf

<sup>\*6</sup> きちんと debian パッケージにしてインストールするようにすれば、この用途で利用することはないはずです。

<sup>\*&</sup>lt;sup>7</sup> debian にある pbuilder や cowbuilder はクリーンな rootfs へ chroot して debian パッケージのビルドができるかをテストする便利コマ ンドです。

<sup>&</sup>lt;sup>\*8</sup> cross debootstrap といい、QEMU と組み合わせます。



杉本典充

## 3.1 はじめに

仮想化技術が進み、OS をインストールする機会が減った人、増えた人とそれぞれ皆様の事情があると思います。今回は Debian をインストールする機会が多い方に向けてインストール作業を自動で行う preseed 機能をご紹介します。

## 3.2 debian をインストールする方法

Debian を PC やサーバへインストールする場合は、Debian Installer を使うことが多いです。\*<sup>9</sup> Debian Installer は以下の形式で提供されており、状況によって使い分けることを想定しています。

- すべての Debian パッケージを収録したイメージ。CD, DVD, Blu-ray の ISO 形式で提供。
- ネットワークインストール用のブートイメージ。通称 netinst イメージ。CD の ISO 形式、USB メモリ形式で 提供。
- vmlinuz、initrd.gz ファイルを単体で提供。仮想化環境やボードデバイス向け。

日常環境で利用する場合は最新バージョンを利用すると思いますので、ネットワークインストール用イメージを使うことをお勧めします(ネットワークから最新のパッケージをダウンロードしてインストールするためです)。

#### 3.3 preseed

#### 3.3.1 preseed とは

preseed とは、Debian Installer の「インストールの実行中に手動で回答を入力せずに、インストールプロセス中の質問の答を設定する方法を提供」する機能をいいます。<sup>\*10</sup>

preseed 機能では、インストール処理のパラメータを定義した入力ファイルが必要になります。そのファイルのデフォルトのファイル名は preseed.cfg となっています。

#### 3.3.2 preseed の種類

preseed 機能は3種類に分かれており、preseed.cfgを読み込む方法やタイミングが異なります。

- initrd preseed
  - Debian Installer の initrd.gz の中に/preseed.cfg を追加で配置する

<sup>\*&</sup>lt;sup>9</sup> ボードデバイスのように Debian Installer を使わないでインストールしている猛者もいます。

 $<sup>^{*10}\ \</sup>rm https://www.debian.org/releases/jessie/amd64/apb.html.ja$ 

- インストーラ起動直後の選択肢から/preseed.cfgの定義に従い自動インストールする
- file preseed
  - Debian Installer の ISO ファイルの中に/preseed.cfg を追加で配置する(=要リマスタリング)
  - Debian Installer の USB メモリの中に/preseed.cfg を追加で配置する
  - Debian Installer の kernel ブートパラメータに preseed/file、 preseed/file/checksum を指定する必要がある
- $\bullet\,$  network preseed
  - IP アドレスの取得または設定後に preseed.cfg を wget でダウンロードして読み込む
  - Debian Installer の kernel ブートパラメータに preseed/url、preseed/url/checksum を指定する必要が ある
  - Debian Installer では IP アドレスを設定する前にも選択肢が出てくるため、完全な自動インストールをするには追加の kernel ブートパラメータが必要

#### 3.3.3 preseed ファイルの仕様

preseed ファイルで指定できるパラメータの解説は以下に詳しく書かれています。

- ●「B.4. 事前設定ファイルの内容 (jessie 用)」
  - https://www.debian.org/releases/stable/amd64/apbs04.html.ja
- ●「B.2.4. preseed で利用できるエイリアス」
  - https://www.debian.org/releases/stable/amd64/apbs02.html.ja#preseed-aliases

Debian-8(jessie) 用のサンプル preseed ファイルは以下で提供しています。

• https://www.debian.org/releases/jessie/example-preseed.txt

### 3.3.4 インストール中のカスタムコマンド実行

preseed で自動インストールをしているときに任意な処理(=カスタムコマンド)を実行する機能があります。カ スタムコマンドの実行タイミングは以下の3パターンを指定できます。

preseed/early\_command

```
全体のインストール処理が始まる前にコマンドを実行したい場合は、"preseed/early_command"で指定できます。
```

d-i preseed/early\_command string anna-install some-udeb

#### 処理名/early\_command

ある処理の実行前にコマンドを実行したい場合は、"処理名/early\_command"で指定できます。

d-i partman/early\_command string debconf-set partman-auto/disk ''\$(list-devices disk | head -n1)''

#### preseed/late\_command

インストールが完了して再起動する前にコマンドを実行したい場合は、"preseed/late\_command"で指定できます。 なお、この状況ではインストール先ディスクの/パーティションが/target ヘマウントした状態になっています。

d-i preseed/late\_command string apt-install zsh; in-target chsh -s /bin/zsh

## 3.4 preseed を使って debian をインストールする

## 3.4.1 インストールする環境

インストールするホスト構成とネットワーク構成は以下の環境とします。

- Debian GNU/Linux 8のKVM ホスト環境を準備
- インストールするホストは、KVM の仮想マシンとする
  - ネットワークカードは1つ
  - ディスクの接続バスは sata<sup>\*11</sup>
  - KVM ホストから KVM ゲストへの接続はシリアルコンソール接続とする

## 3.4.2 今回作成した preseed.cfg ファイル

以下の preseed.cfg ファイルの設定で自動インストールします。

<sup>\*&</sup>lt;sup>11</sup> KVM を使う場合はディスクの接続バスを virtio にする方が性能はよいです。virtio を使う場合は preseed の定義にある/dev/sda の部分 を/dev/vda に変更することを忘れないでください。

d-i debian-installer/language string C	
d-i debian-installer/country string JP	
d-i debian-installer/locale string ja_JP.UTF-8	
d-i keyboard-configuration/xkb-keymap select jp	
d-i netcfg/enable boolean true	
d-i netcfg/choose_interface select auto	
d-i netcfg/get_hostname string deb-preseed	
d-i netcfg/get_domain string localdomain	
d-i netcfg/hostname string dev-preseed	
d-i netcfg/wireless_wep string	
d-i hw-detect/load_firmware boolean false	
d-i mirror/country string manual	
d-i mirror/http/hostname string ftp.jp.debian.org	
d-i mirror/http/directory string /debian	
d-i mirror/http/proxy string	
d-i mirror/suite string jessie	
d-i passwd/root-login boolean true	
d-i passwd/make-user boolean true	
d-i passwd/root-password password rootpass	
d-i passwd/root-password-again password rootpass	
d-i passwd/user-fullname string Test User	
d-i passwd/username string testuser	
d-i passwd/user-password password testpass	
d-i passwd/user-password-again password testpass	
d-i passwd/user-default-groups string audio cdrom video sudo	
d-i clock-setup/utc boolean true	
d-i time/zone string Asia/Tokyo	
d-i clock-setup/ntp boolean true	
d-i clock-setup/ntp-server string ntp.nict.jp	
d-i partman-auto/init_automatically_partition select biggest_free	
d-i partman-auto/disk string /dev/sda	
d-i partman-auto/method string regular	
d-i partman-lvm/device_remove_lvm boolean true	
d-i partman-md/device_remove_md boolean true	
d-i partman-lvm/confirm boolean true	
d-i partman-lvm/confirm_nooverwrite boolean true	
d-1 partman-auto/choose_recipe select atomic	
d-1 partman-partitioning/confirm_write_new_label boolean true	
d-1 partman/choose_partition select finish	
d-1 partman/confirm boolean true	
d-1 partman/confirm_nooverwrite boolean true	
d-1 partman-md/confirm boolean true	
d-1 partman-partitioning/confirm_write_new_laber boolean true	
d-1 partman/cnoose_partition select linish	
d-i partman/confirm poorean true	
d i partman/confirm_nooverwrite boorean true	
d i partman/mount_style select uuru	
d i base installer/install recommends boorean laise	
d-i ant-setun/non-free boolean false	
d-i ant-setup/non liee boolean taise	
d-i ant-setun/use mirror boolean true	
tasksel tasksel/first multiselect ssh-server	
d-i pkgsel/include string ntp ntpdate sudo curl	
d-i pkgsel/upgrade select none	
popularity-contest popularity-contest/participate boolean false	
d-i grub-installer/skip boolean false	
d-i grub-installer/only_debian boolean true	
d-i grub-installer/with_other_os boolean true	
d-i grub-installer/bootdev string /dev/sda	
d-i debian-installer/add-kernel-opts string console=ttyS0,115200n8	
d-i finish-install/reboot_in_progress note	
d-i cdrom-detect/eject boolean true	
d-i preseed/late_command string \	
in-target /usr/bin/curl http://192.168.22.102/preseed/done.cgi	

#### 3.4.3 netinst イメージを使った自動インストール

netinst 用 ISO ファイルからブートしてインストールする場合は、Debian Installer の kernel ブートパラメータに 以下のようなコマンドを追加で指定します。

auto=true priority=critical url=http://192.168.22.41/preseed.cfg preseed/url/checksum=8e85ff2ddb966321b91f13f9aba9dc9f

preseed/url/checksum は md5sum なのですが、指定が必須となっているため注意しながら入力します。もし checksum が合わない場合は preseed.cfg ファイルは無効と判定される仕様になっています<sup>\*12</sup>。

#### 3.4.4 virt-install を使った自動インストール

KVM の仮想マシンをインストールするコマンドに"virt-install"があります。この virt-install を使って KVM ゲ ストマシンのインストールに preseed を使う場合は、"-initrd-inject"オプションを指定します。

<sup>\*12</sup> その場合は、インストールがストップしたり、手動でインストール場合と同じく質問に答える画面で入力待ち状態になります。

"-initrd-inject"オプションは initrd.gz の中に指定したファイルを忍び込ませることができます。なお、このインストール方法の場合は Debian Installer のブート時に/preseed.cfg ファイルが存在することから initrd preseed の動作

## でインストール処理が進みます。

```
$ sudo virt-install \
    --name deb-preseed-1 \
    --disk path=/var/lib/libvirt/images/deb-preseed-1.img,format=qcow2,bus=sata \
    --vcpus 1 --ram 1024 \
    --network bridge=br0,model=e1000 \
    --graphics none \
    -os-type linux --os-variant generic \
    --console pty,target_type=serial \
    --location 'http://ftp.jp.debian.org/debian/dists/jessie/main/installer-amd64/' \
    --initrd-inject=/var/lib/libvirt/images/preseed.cfg \
    --extra-args 'console=ttyS0,115200n8 serial'
```

## 3.5 まとめ

Debian にある preseed 機能を使って自動インストールを試してみました。KVM などの仮想環境を扱っていて日常 的に Debian をインストールしている場合は効率的に作業ができると思いますので活用してみてください。

## 3.6 参考文献

- $\bullet \ \texttt{`} DebianInstallerPreseed \ \texttt{`} https://wiki.debian.org/DebianInstaller/Preseed \ \texttt{`} https://wiki.debian.org/D$
- •「付録 B preseed を利用したインストールの自動化」https://www.debian.org/releases/jessie/amd64/ apb.html.ja

# 4 Let's Encrypt のススメ

かわだ てつたろう

## 4.1 はじめに

無料で SSL/TLS 証明書を取得できるサービスとして Start SSL や Wo Sign、CAcert などいくつかありますが、 今回は Debian パッケージを使用して簡単に導入できる Let's Encrypt を紹介します。

## 4.2 Let's Encrypt について

Let's Encrypt<sup>\*13</sup>は 2016 年 4 月 12 日にサービスを開始した認証局 (CA) です。日本語の情報は「Let's Encrypt 総合ポータル<sup>\*14</sup>」にまとまっていますので参照してください。

その特徴として

- フリーで自動化されたオープンな認証局
- 非営利団体 ISRG(Internet Security Research Group) が運営
- ACME(Automated Certificate Management Environment)  $\mathcal{I}\Box \vdash \exists \mathcal{V}$

があげられます。

#### 4.2.1 証明書

発行される証明書は、ドメイン認証 (DV:Domain Validation) 証明書です。企業認証 (OV:Organization Validation) 証明書や EV(Extended Validation) 証明書は発行されません。ワイルドカード証明書には対応していません が、複数のサブドメイン名の証明書が取得できますので問題にはならないでしょう。

ルート証明書は、IdenTrust 社の証明書 (DST Root CA X3) でクロス署名されておりほとんどの環境に対応して います<sup>\*15</sup>。Debian では Debian 6 squeeze から使用できます。

証明書の期限は 90 日となっており、60 日ごとの更新がすすめられています<sup>\*16</sup>。

## 4.2.2 ACME

ドメイン認証証明書発行にはドメイン所有者の確認が必要です。多くの場合その確認に CA から送られてきたテキ ストを

対象ドメインの Web サーバで公開

<sup>\*13</sup> https://letsencrypt.org/

<sup>\*14</sup> https://letsencrypt.jp/

<sup>\*&</sup>lt;sup>15</sup> https://community.letsencrypt.org/t/which-browsers-and-operating-systems-support-lets-encrypt/4394

 $<sup>^{*16} \; \</sup>texttt{https://letsencrypt.org/2015/11/09/why-90-days.html}$ 

- 対象ドメインの DNS の TXT レコードに追加
- 対象ドメインのメールアドレスで受けとり CA の Web ページに入力

するなどドメイン所有者にしかできないいずれかの方法が取られます。

このようなドメイン所有者の確認を含めた CSR の作成から証明書発行までの手順を自動化し標準化した仕様が ACME<sup>\*17</sup> になります。仕様はサーバ、クライアント両方について規定されており、Let's Encrypt はサーバ側のリ ファレンス実装といえます。

ACME では次のいずれかの方法でドメイン所有者の確認が行なわれます。

- 対象ドメインのサーバで TLS を有効
- 対象ドメインの DNS の TXT レコードに指定したテキストを追加
- 対象ドメインの Web サーバで指定したテキストを公開

## 4.3 導入

それでは、実際に Let's Encrypt の証明書を取得します。環境は、DNS の設定は完了している Debian 8.6 jessie で行ないます。また、事前に利用規約\*<sup>18</sup>を読んで同意しておいてください。利用規約に同意したものとして手順を紹介します。

#### 4.3.1 certbot

ACME のクライアントアプリが Debian では certbot パッケージとして提供されています。jessie では jessiebackports に含まれていますので backports を有効にしてインストールします。

\$ sudo -c 'echo deb http://ftp.jp.debian.org/debian jessie-backports main >> /etc/apt/source.list'
\$ sudo apt update
\$ sudo apt install certbot -t jessie-backports

#### 4.3.2 証明書取得

証明書を取得するには次のコマンドを実行します。Web サーバ (ポート 80 と 443 を使用しているプロセス) を止め ておいてください。

<pre>\$ sudo certbot certonlystandaloneagree-tos -m postmaster@example.org -d example.org IMPORTANT NOTES:</pre>	
- Congratulations! Your certificate and chain have been saved at	
on 2016-12-23. To obtain a new or tweaked version of this	
certificate in the future, simply run certbot again. To	
renew"	
- If you like Certbot, please consider supporting our work by:	
Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate Donating to EFF: https://eff.org/donate-le	

これで example.org ドメインの証明書が取得でき、/etc/letsencrypt/live/example.org 以下に cert.pem, chain.pem, fullchain.pem, privkey.pem ができあがります。これらファイルを Web サーバなどに設定して SSL/TLS を有効にします。ファイルは/etc/letsencrypt/archive/example.org 以下の世代ファイルへのシンボリックリンクと なっており、証明書を更新しても参照側の設定変更の必要をなくしています。

#### 4.3.3 webroot の使用

多くの環境では Web サーバが稼動しているはずです。証明書取得の度に Web サーバを止めるわけにはいきません ので、稼動している Web サーバを利用してドメイン所有の確認を行なう webroot オプションを使うことになります。

<sup>\*17</sup> https://github.com/ietf-wg-acme/acme/

<sup>\*18</sup> https://letsencrypt.org/repository/

複数のドメイン、サブドメインを使用する場合はドメインごとに webroot の設定を行なうことになりますが、これ

を一つのディレクトリにまとめる方法が Arch Wiki に記載されています\*19。

これを利用して nginx 環境で行なうと次のようになります。

```
$ cat /etc/nginx/snippets/letsencrypt.conf
location ~~ /.well-known/acme-challenge {
    alias /var/lib/letsencrypt/.well-known/acme-challenge;
    default_type "text/plain";
    try_files $uri =404;
}
$ cat /etc/nginx/sites-enabled/default
server {
    ...
    include /etc/nginx/snippets/letsencrypt.conf;
}
$ sudo certbot certonly --webroot --agree-tos -m postmaster@example.org -d example.org --hsts -w /var/lib/letsencrypt/
```

## 4.3.4 証明書の更新

証明書の更新は renew で行ないます。

\$ sudo certbot renew

更新は、証明書取得時に生成される/etc/letsencrypt/renewalの設定ファイルに従って行なわれます。

Let's Encrypt では 1 日 2 回証明書の更新を行なうことが推奨されています。Debian パッケージに は/etc/cron.d/certbot が含まれておりインストールした時点で行なうようになっています。

証明書は期限が指定期日 (デフォルト 30 日) 未満になると更新されますが、更新された場合 Web サーバ など使用しているサービスの再起動などが必要です。そのためのオプションとして-post-hock がありますの で/etc/cron.d/certbot に指定しておくとよいでしょう。

#### 4.3.5 その他

certbot のデフォルトサブコマンド run では証明書の取得から Web サーバへのインストールまでを行います。筆者 は Web サーバの設定ファイルまで書き換えるのは望まないので使用しませんでした。nginx には対応途中のようで すが Apache をお使いの方は試されてみるとよいでしょう。

その他に証明書の失効などさまざまなことも certbot コマンドで行なえますので詳しくはドキュメント\*20してください。

## 4.4 まとめ

Let's Encrypt の紹介と certbot を使った証明書の取得、更新の仕方について説明しました。手軽に証明書の取得 が行なえますのでぜひ導入して暗号化しましょう。ただし、期限の短い証明書になりますので、自動更新設定をきっ ちり行なって期限切れにならないよう気をつけてください。

 $<sup>^{*19} \; \</sup>texttt{https://wiki.archlinux.org/index.php?title=Let\%E2\%80\%99s\_Encrypt}$ 

<sup>\*20</sup> https://certbot.eff.org/docs/

# 5 OpenFOAM で数値流体解析

Yosuke Otsuki

## 5.1 背景:仕事で数値計算

飛行機の翼が発生する揚力、建物の耐震性、新薬の成分などは先人たちの発見してくれた数学的な法則によって、 ある程度は計算によって導くことができます。今日では、これらの方程式を人間が手で解くことは少なくなりつつあ ります。数値計算と呼ばれる、人間が方程式を計算機で計算できる形に変形し、実装して、計算機に解かせる方法に が広がってきたためです。流体や構造と行った方程式をすでにプログラムしておいて、設計者が数値を入力するだけ で使えるようになっているソフトウェアを使用し工学に応用することを CAE (Computer Aided Engineering) とい います。

CAE は主に 3 つの分野に分類できます。解析する形状を作成する (CAD)、解析を行う (ソルバー)、そして、解 析の結果を確認する (可視化) の 3 分野です。本日紹介する OpenFOAM は、流体分野に特化した解析を行うソフト です。

## 5.2 概要: なぜ Open Source の解析ソフト?

数値計算は多くの計算資源を必要とします。ハードウェアが安価になったことから、複数ノードによる分散処理 によって大規模な問題を解く方法が一般的に広まってきました。しかし、大抵の場合、商用の CAE ソフトウェアは ノード単位でライセンスを購入する必要があり、たとえ計算資源が十分でも金銭的な余裕がなければ、並列計算を利 用した解析を実行できないことがあります。このような問題から、Opensource の解析ソフトが注目されるようにな りました。実際に製造業の分野での使用も広がっています。また、商用のソフトとの違いは、拡張性に富んでいるこ とです。メッシュ関連のライブラリ、乱流モデルのライブラリ、移流行離散化方法のライブラリなどが用意されてお り、既存のアプリケーションもそれらのライブラリを呼び出した数百行程度のソースコードで書かれています。

## 5.3 Install

#### 5.3.1 Debian

debian では freefoam というパッケージ名でビルド済みの OpenFOAM を使用することができます。freefoam と OpenFOAM との違いは、freefoam では、kiva\*<sup>21</sup>のピストンモデルなどがなく、商用可視化ソフトへの出力形式な ども削られているようです。OpenFOAM の開発は続けられていますが、freefoam の開発は数年前から止まったまま のようです。

<sup>\*&</sup>lt;sup>21</sup> 内燃機関に特化した商用解析ソフト

#### 5.3.2 Build: ビルドの準備

freefoam は debian 系のパッケージならば、お手軽に OpenFOAM を体験できる点で有利ですが、最新の OpenFOAM には新機能や不具合の修正なども反映されているため、最新版を使用することをおすすめします。 最新の安定版は、http://www.openfoam.com からダウンロードできます。開発版は github のレポジトリを参照し てください。https://github.com/OpenFOAM ダウンロードするべきアーカイブは 2 つあります。

```
OpenFOAM-x.y.z.tgz
ThirdParty-x.y.z.tgz
```

OpenFOAM 自体のソースコードと OpenFOAM が必要とするライブラリをまとめたものです。このライブラリ 群は debian のものを使用することもできます。ただし、OpenFOAM に同梱されたものを使用したほうが、バー ジョンの互換性が取れるため安全です。OpenFOAM が必要とするライブラリのうち ThirdParty にソースが同伴さ れているものは以下です。

```
ThirdParty-x.y.z

|-- openmpi

|-- scotch

|-- pscotch

|-- metis (option)

|-- libcgal
```

openmpi は Message Passing Interface のライブラリです。scotch は領域分割ツール、ptscotch は scotch の並列 計算版です。metis も領域分割ツールです。libcgal は計算機科学ライブラリです。ドロネー分割機能が依存していま す。この機能が必要なければ、インストールしなくても問題ありません。boost がバージョン 3.0 から必要になりま した。ビルドを開始する前に必要なパッケージをインストールしておきます。

```
sudo apt-get install build-essential flex bison \
    cmake zliblg-dev libboost-system-dev \
    libboost-thread-dev libopenmpi-dev openmpi-bin \
    gnuplot libreadline-dev libncurses-dev libxt-dev
```

以上で、ビルド環境が整いました。さらに詳しいシステム要件、どのパッケージをビルドするのに何が依存しているか、redhat 系での環境構築方法などは、本家のサイトを参照してください。http://www.openfoam.com/ documentation/system-requirements.php

#### 5.3.3 Build: ビルド作業

インストール用のディレクトリを作成してください。ビルド作業もこのディレクトリの中で行われます。このディ レクトリの場所は必須ではありませんが、これ以外のディレクトリにする場合、設定ファイルの編集が必要になり ます。

```
<sup>$ mkdir $HOME/OpenFOAM</sup>
上記のディレクトリの中で、ソースコードを展開します。以下、バージョンを 3.0.x とします。
```

```
$ tar xzf OpenFOAM-3.0.x.tgz
$ tar xzf ThirdParty-3.0.x.tgz
```

設定ファイルは以下です。\$HOME/OpenFOAM をインストール先にしていない場合は、これらのファイルの編集をしてください。また、時間短縮のため、平行コンパイルをおすすめします。使用する設定ファイルをテキストエディタで開いて、WM\_NCOMPPROCS=4 (set WM\_NCOMPPROCS=4)と記述しておくと4並行でコンパイルしてくれます。

```
$HOME/OpenFOAM/OpenFOAM-3.0.x/etc/bashrc
$HOME/OpenFOAM/OpenFOAM-3.0.x/etc/cshrc
```

設定が完了したら、ビルドを開始します。まず、設定した変数を反映し、各種ツールにパスが通ったか確認します。

```
$ cd $HOME/OpenFOAM/OpenFOAM-3.0.x
$ . etc/bashrc
$ cd $WM_PROJECT_DIR | pwd
/home/yosuke/OpenFOAM/OpenFOAM-3.0.x
```

ビルドしようとしている OpenFOAM のディレクトリが表示されるはずです。問題なければ、ビルドを始めます。 相当高性能なマシンでも 30-40 分ぐらいかかります。

wmake >& build.log

ログにエラーが出ていなければ、ビルド完了です。ついでに、解析結果を可視化するためのツールもインストール しておきましょう。paraview という kitware が作成している並列可視化ツールをインストールします。debian なら ば、下記コマンドで簡単にインストールできます。

# apt-get install paraview

#### 5.4 解析してみる

チュートリアルを利用して解析を行ってみましょう。今回は、バイクの周りの空気の流れの解析を行ってみましょう。

cd OpenFOAM-3.0.x/tutorials/incompressible/pisoFoam/les/motoBike/motorBike 1s 0 物理量の初期状態を格納、形状が時間変化する場合は形状情報も格納される constant system 各種設定を格納:空間離散化の方法、時間積分方法、線形ソルパ、領域分割設定、結果出力の時間間隔、時間増分など

実際には、このチュートリアルでは、これ以外のファイルがあるはずです。しかし、これらは簡易実行用のスクリ プトやチュートリアル用の初期データなので、解析には必須ではありません。必須なのは上記で挙げたディレクトリ のみです。

#### 5.4.1 メッシュの作成

解析を行うには、1) メッシュを作成、2) 入力値を設定する、3) 並列計算する場合は、領域を分割する、4) 解析を実行する。以上のようなステップを踏みます。まずはメッシュから作成しましょう。OpenFOAM では blockMesh と単純形状のみ作成できるメッシャーと snappyHexMesh という複雑な形状に対応したメッシャーが利用できます。他にも、一般的なメッシュ作成ツールから形状を取り込むこともできます。snappyHexMesh は blockMesh で作成した計算領域を少しずつ形状を変化させ最終形上に近づけてゆきます。そのため、最初に blockMesh の実行が必要です。



一度 paraview を起動して現在どのようなメッシュが作成されているか観察してみることにしましょう。まず、 paraview を起動します。

```
$ paraview &
```

[File] / [Open] から Open File: ダイアログを開きます。 [Files of type] から [All files (\*)] を選択すると、 Filename に controlDict というファイルが現れるはずです。そのファイルを選択し [OK] ボタンを押し、[Open With] ダイアログのファイル形式のリストから OpenFOAM を選択してください。[OK] ボタンを押下すると、メイ ンの画面に戻ります。その後 [Apply] ボタンを押すと四角いブロックが表示されるはずです。

今度は、snappyHexMesh を使用して実形状のメッシュを作ります。このメッシャーは、並列実行することが可 能です。numberOfSubdomains を編集して2にし、そして、simpleCoeffsのn(211)としてあります。こ れは、先ほどの四角いブロックを2分割して並列でメッシュ作成を行います。nの成分をすべて乗算したとき、 numberOfSubdomains と同じになるようにしてください。

poseParDict
*\ ield   OpenFOAM: The Open Source CFD Toolbox   peration   Version: 3.0.x   id   Web: www.OpenFOAM.org   anipulation
*/
<pre>2.0; ascii; dictionary; "system"; decomposeParDict; * * * * * * * * * * * * * * * * * * *</pre>
ins 2;
ptscotch;
(2 1 1); 0.001;
ffs
(2 1 1); 0.001; xyz;
"cellDecomposition":
1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1

お使いの計算機のコア数と同じにすると、最も効率よく実行できます。並列で snappyHexMesh を使用するには下 記のように実行します。

mpirun -np 2 snappyHexMesh -parallel >& snappyHexMesh.log

計算にはかなり時間がかかります。先ほどと同じように、paraview で作成したメッシュを見てみましょう。バイクを運転しているひとのメッシュが作成されているはずです。



## 5.5 解析結果を実行する

解析を実行してみましょう。今回のチュートリアルは、非圧縮性の粘性流体を PISO 法で解きます。このアルゴリズムは、pisoFoam というアプリケーションで利用できます。移流項の空間離散は 1 次風上差分、拡散項は 2 次精度の中心差分を利用しています。これらは、system/fvSchemes で設定されています。線形ソルバは AMG 前処理ありの CG 法を使用します。こちらは、system/fvSolutions で設定されています。

## 5.6 解析結果を見てみる

解析は終了しましたが、解析結果はデータファイルとして出力されただけです。数値ファイルから、数百万点の要素を持つメッシュから空気の流れを読み取ることなどはほぼ不可能です。そのため、解析結果のデータファイルを人間が目で見て理解できるように表示してくれる可視化ソフトを使用します。すでにメッシュ作成のところで紹介済みですが、ここでも paraview を使用します。



## 5.7 まとめ

#### 5.7.1 良い点

OpenFOAM は、商用ソフトと比べて数学的な法則に乗っとてくれているので嬉しいです。商用のソフトは、数値 解析の安定性を確保するために、数学的なトリックを多用していることが多いです。1000 並列ぐらいまでならば、問 題なくスケーリングすることが知られています。ただし、理論性能比は察してください。そして、ソースコードさえ 読めれば、どんな機能でも自分でカスタマイズなどができることです。

#### 5.7.2 悪い点

数値解析の安定性はあまり良くないです。かなりの数値流体の知識が要求されます。Unix/Linux に詳しくない技術者にはとっつきにくいかもしれない。



## 6.1 自己紹介

吉野 与志仁 (よしの よしひと,@yy\_y\_ja\_jp)です。Debian 公式開発者ではないです。manpages-ja パッケージの メンテナです。Debian JP Project メンバーです。

## 6.2 Agenda

Debian パッケージの "説明文 "について。

- 1. Debian とは?
- 2. パッケージ
- 3. DDTP / DDTSS
- 4. まとめ

## 6.3 Debian とは?

https://www.debian.org/



Debian – The Universal Operating System

- The 一つしかない
- Universal 普遍的な
- Operating System オペレーティングシステム (OS)

## 6.4 Operating System

Operating System – オペレーティングシステム (OS)

- "コンピュータを動作させるために必要な基本プログラムとユーティリティの集合体"
- "Debian は、純粋な OS 以上の機能を提供します。あなたのコンピュータに手軽にインストールできるよう 43000 を越すコンパイル済ソフトウェアが パッケージとして付属しています。"

ざっくり言うと

- コンピュータを動かすもの
- あなたの使いたいソフトを手軽にインストールできるもの

#### 6.5 Universal Operating System

- Universal 普遍的な
- Operating System オペレーティングシステム (OS)

#### Debian lt

- 様々なコンピュータ(ハードウェア)を動かせます
   アーキテクチャ・移植版 (amd64, arm64, ...)
- 様々なソフトウェアを手軽にインストールできます
   43000 を越すパッケージ

#### Debian を様々な人々が使えます

- 人種・民族・言語 翻訳
   Debian のインストーラ (d-i) は 75 (22) 言語に対応
- 年齡 Debian Jr., Debian Edu, ...
- 障碍
   d-i は点字ディスプレイに対応
   Debian Accessibility, ...

## 6.6 パッケージ

- ソフトウェアを手軽にインストールできるようにしたもの
- 各パッケージには基本的に必ず責任者(メンテナ)が存在
   Debian はコミュニティベースなので、基本ボランティア
   そのソフトウェアを多くの人に使ってもらいたいという思いがある人など(そのソフトウェアの業界の人や、
   そのソフトウェアの作者自身であることも)
- 各パッケージには" control "ファイルが含まれ、それに基づいて管理されている

## 6.7 control ファイルの例 (jessie の nginx パッケージ)

Package: nginx Version: 1.6.2-5+deb8u2 Architecture: all Maintainer: Kartik Mistry <kartik@debian.org> Installed-Size: 99 Depends: nginx-full (>= 1.6.2-5+deb8u2) | nginx-light (>= 1.6.2-5+deb8u2) | nginx-extras (>= 1.6.2-5+deb8u2), \\ nginx-full (<< 1.6.2-5+deb8u2.1^{-}) | nginx-light (<< 1.6.2-5+deb8u2.1^{-}) | nginx-extras (<< 1.6.2-5+deb8u2.1^{-}) Section: httpd Priority: optional Homepage: http://nginx.net Description: small, powerful, scalable web/proxy server Nginx ("engine X") is a high-performance web and reverse proxy server created by Igor Sysoev. It can be used both as a standalone web server and as a proxy to reduce the load on back-end HTTP or mail servers. This is a dependency package to install either nginx-full (by default) or nginx-light.

## 6.8 パッケージ

たくさんあるパッケージからどうやって選びますか?

パターン 1

- 1. **ぐぐる**
- 2. 誰かのブログ記事を読む「~という名前のパッケージをインストールしましょう」
- 3. それに従うか決める

パターン 2

- 1. **ぐぐる**
- 2. Debian のパッケージページに行き着く
  - 内容は control ファイルの中身とほぼ同じ
- 3. 読んでインストールするか決める



## 6.9 control ファイルの中身

Package - パッケージの名前 (例: Package: nginx) Maintainer - パッケージのメンテナ例: Maintainer: Kartik Mistry <kartik@debian.org> Section - パッケージの分類 (例: Section: httpd)

Description – パッケージの説明文

- Debian Policy 3.4
  - 初めて見た人がインストールしたいか判断できるように書くべしとされている
- 重要なことは説明文の出だしに書くことになっている
- 1 行目 short description (短い説明文)
   例:

Description: small, powerful, scalable web/proxy server

2行目以降 – long description (長い説明文)

# Nginx ("engine X") is a high-performance web and reverse proxy server created by Igor Sysoev. It can be used both as a standalone web server and as a proxy to reduce the load on back-end HTTP or mail servers.

This is a dependency package to install either nginx-full (by default) or nginx-light.

## 6.10 Description - パッケージの説明文

control ファイルの Description は英語

Debian はコミュニティベースなので、誰かボランティアが翻訳する

- パッケージになっているソフトウェアを知らないと正しく翻訳できない
- Universal なのでソフトウェアの分野(業界)が幅広く、少人数でやるのは困難

自分の普段使っているパッケージ・自分の専門分野のパッケージなら翻訳できるはず!

## 6.11 DDTP – Debian Description Translation Project / DDTSS

http://ddtp.debian.net/

- パッケージの Description を翻訳するプロジェクト
- Debian 公式開発者でなくても作業できます
- 日本語チームの方針:翻訳の品質を向上させること。量ではない(でも原文の伸びはすごい...)



## 6.12 DDTP – Debian Description Translation Project

## 言語別統計

例:



## 6.13 DDTP – Debian Description Translation Project

#### 言語別・Priority 別統計

## DDTP stats

Here are some stats generated from the DDTP server with regards to translations.

These stats were current as of 2016-06-16 04:42:16 UTC (+0000).

The distribution measured is **sid/main** 

The architecture used is amd64

Note this uses the actual priorities out of the Packages files. The DDTS uses a slightly different measurement, which is why they don't use exactly the same list.

The numbers are counts. If it says 35, that means that 35 of the 43 (where 43 is the total number listed at the bottom) packages of that priority have been translated. If you hold your mouse over the number, it gives examples of what is missing.

Popcon500 simply counts the number of translated packages in the top 500 ranked in the <u>PopCon-rank-by-vote</u>. This is to get some idea of the translations with respect to the packages people actually use.

Popconrank counts the number of votes for translated packages. That is, roughly, the percentage indicates the probability that a vote for a package in popcon was a vote for a translated package.

Number of translated package descriptions per category from the Package file:

						Co	icegoi)	/							
	Lang	Ree	quired	Imp	ortant	Sta	andard	0pti	onal	Ex	tra	Pop	con500	Popcor	nRank
it	(Italian)	61	(100%)	61	(100%)	84	(100%)	29861	(80%)	8713	(62%)	500	(100%)	62.45M	(99%)
da	(Danish)	61	(100%)	61	(100%)	83	(98%)	24616	(66%)	8457	(60%)	499	(99%)	61.25M	(97%)
fr	(French)	61	(100%)	61	(100%)	84	(100%)	9333	(25%)	2734	(19%)	500	(100%)	54.10M	(86%)
de	(German)	61	(100%)	60	(98%)	82	(97%)	10440	(28%)	2160	(15%)	498	(99%)	60.39M	(96%)
sk	(Slovak)	61	(100%)	61	(100%)	84	(100%)	6044	(16%)	1065	(7%)	500	(100%)	61.04M	(97%)
ja	(Japanese)	61	(100%)	60	(98%)	83	(98%)	6499	(17%)	981	(6%)	497	(99%)	52.87M	(84%)
pt_	BR (Brazilian Portuguese)	61	(100%)	58	(95%)	81	(96%)	4792	(12%)	1015	(7%)	479	(95%)	51.30M	(81%)
ru	(Russian)	60	(98%)	60	(98%)	79	(94%)	2475	(6%)	394	(2%)	438	(87%)	44.38M	(70%)
ko	(Korean)	61	(100%)	60	(98%)	71	(84%)	2233	(6%)	235	(1%)	381	(76%)	40.68M	(64%)
es	(Spanish)	61	(100%)	61	(100%)	60	(71%)	1878	(5%)	197	(1%)	231	(46%)	27.13M	(43%)
								-							

## 6.14 DDTSS

Web で Description の翻訳・レビュー・修正ができるサイト 日本語への翻訳は http://ddtp.debian.net/ddtss/index.cgi/ja

## DDTSS for ja

#### Project-Messages (0):

None

#### Please login... Only with a login, you can use all features! Go to Login

The DDTSS is a queue-based system. New translations are automatically fetched only when there are only a few pending translation. Translations will only be sent off after they have received a certain amount of review. New translations will be fetched in priority order, so the goal is to get your priority numbers as low as possible!

The rate of getting new descriptions to translate is not automatic. If you find you keep running low on things to translate, ask me to increase it. Common translation rules

## ?ja-Team-Messages (0):

None

Note: this language requires a login to do edits. Please sign up to the language mailing list to learn the procedures before doing anything.

#### Pending translation

The number of translations pending review has gotten quite large. Consider reviewing some translations first.

<u>ambdec</u> (priority 50)
 <u>gringo</u> (priority 50)

## Infos about :

- Account: 203.212.
- not logged in (<u>Login</u>)
  Translations:
- Translation
   Reviews:

## 6.15 DDTSS

Debian 公式開発者でなくても作業できます

誰でもアカウントを作成して翻訳・レビュー・修正ができます

詳しくは 第 53 回東京エリア Debian 勉強会 (2009 年 6 月勉強会)の資料など

https://tokyodebian.alioth.debian.org/2009-06.html

## DDTSS for ja

#### ?Project-Messages (0):

Nor

The DDTSS is a queue-based system. New translations are automatically fetched only when there are only a few pending translation. Translations will only be sent off after they have received a certain amount of review. New translations will be fetched in priority order, so the goal is to get your priority numbers as low as possible! The rate of getting new descriptions to translate is not automatic. If you find you keep running low on things to translate, ask me to increase it.

Common translation rules

#### ? ja-Team-Messages (0):

#### None

Note: this language requires a login to do edits. Please sign up to the language mailing list to learn the procedures before doing anything.

#### Pending translation

The number of translations pending review has gotten quite large. Consider reviewing some translations first.

- ambdec (priority 50)
- <u>ambdec</u> (priority 50)
   <u>gringo</u> (priority 50)
   <u>python-empy</u> (priority 50)
   <u>tk2</u> (priority 50)
- <u>tk2</u> (priority 50)
   <u>tumgreyspf</u> (priority 50)
   <u>uchardet</u> (priority 50)
   <u>xmlbeans</u> (priority 50)
   <u>zita-rev1</u> (priority 50)
   <u>augeas-doc</u> (priority 47)

- <u>auqeas-doc</u> (priority 47)
   <u>auqeas-lenses</u> (priority 47)
   <u>lua-auqeas</u> (priority 47)
   <u>python-auqeas</u> (priority 47)
   <u>tomcate</u> (priority 47)
   <u>tomcate</u> (priority 47)
   <u>tomcate-docs</u> (priority 47)
   <u>libonding-auqeas-extra</u> (priority 47)
   <u>libonding-auqeas-extra</u> (priority 45)
   <u>libindirect-perl</u> (priority 45)
   <u>libindirect-perl</u> (priority 45)
   <u>libindirect-backend-auqeas-perl</u> (priority 45)

- 20. <u>libconfig-model-backend-augeas-perl</u> (priority 44)
- augeas-dbg (priority 43) libmono-debugger-soft4.0a-cil (priority 43) 21. 22.
- <u>libzvbi-common</u> (priority 42)
   <u>libaugeas0</u> (priority 41)
   <u>libaugeas-dev</u> (priority 34)

Fetch specific description

#### Force fetching even if not untranslated

#### Pending review (510)

- <u>libyte-2.91-dev</u> (needs initial review)
   <u>libyte-dev</u> (needs initial review)
   <u>libyte-dev</u> (needs initial review)
   <u>libyte-dev</u> (needs initial review)
   <u>liboittio-dev</u> (needs initial review)
   <u>liboittio-dev</u> (needs initial review)
- libocas0 (needs initial review)
- <u>libocas-dev</u> (needs initial review)
   <u>tcl8.5</u> (needs initial review)

#### Infos about YOSHINO Yoshihito:

- Account: yy\_y\_ja\_jp
  logged\_in (<u>Logout</u>)
  Translations: 5372
  Reviews: 2231

#### Messages for you (1)

• (<u>ban\_nobuhiro</u>): Sat Jul 9 12:59:14 2011 <u>DEL</u> 夜中は重いかもしれません。特に02-03時台

Reviewed by you (0)

#### None

Recently translated

- 1. <u>xul-ext-greasemonkey</u> (ok) Tue May 31 13:42:12 2016
  2. <u>wordnet-dev</u> (ok) Tue May 31 13:42:12 2016
  3. <u>python-unshare</u> (ok) Tue May 31 13:42:11 2016
  4. <u>debian-refcard</u> (ok) Tue May 31 13:42:07 2016
  5. <u>debian-kernel-handbook</u> (ok) Tue May 31 13:42:07
  2016 2016 . <u>debian-handbook</u> (ok) Tue May 31 13:42:06 2016 . <u>debian-faq-zh-cn</u> (ok) Tue May 31 13:42:05 2016 8. <u>debian-faq-ru</u> (ok) Tue May 31 13:42:05 2016 10. <u>debian-faq-fr</u> (ok) Tue May 31 13:42:04 2016 11. <u>debian-faq-fr</u> (ok) Tue May 31 13:42:04 2016 12. <u>debian-faq-fr</u> (ok) Tue May 31 13:42:03 2016 13. <u>xul-ext-stylish</u> (ok) Thu May 19 12:22:23 2016 14. <u>wildmidi</u> (ok) Thu May 19 12:22:28 2016 15. <u>unbound-nost</u> (ok) Thu May 19 12:22:28 2016 16. <u>unbound-anchor</u> (ok) Thu May 19 12:22:27 2016 15. <u>tinproyx</u> (ok) Thu May 19 12:22:27 2016 19. <u>stretchplayer</u> (ok) Thu May 19 12:22:26 2016 20. <u>gviaggiatreno</u> (ok) Thu May 19 12:22:26 2016

## 3人がレビューすると翻訳済みになり、その説明文が Debian のレポジトリに反映されます



## 6.16 まとめ

- Debian は Universal な OS なので
  - 幅広い分野のソフトウェアのパッケージが多数ある

#### - その中からインストールしたいものを選ぶきっかけとして、パッケージ名、説明文などがある

- 様々な母国語・言語のユーザーが使う 翻訳を提供

- Debian はコミュニティベースなので基本ボランティア
   パッケージのメンテナンス、翻訳、...
- 日本語チームとしては、質の高い日本語でパッケージ説明文を日本語ユーザーに提供したい
- DDTSS で誰でも説明文の翻訳ができます
- パッケージのソフトウェアやその分野(業界)を知らないと翻訳ができない。質の高い翻訳もできない ユーザーのみなさん、お使いのパッケージの翻訳・レビュー・修正にご協力ください!
   みなさん、あなたの専門分野のパッケージの翻訳・レビュー・修正にご協力ください!

## 6.17 補足1

質問:DDTSS にアクセスしてみたいのですが、つながりません。

はい、5月からつながりません。今のところ /etc/hosts に次の行を追加してください。 117.121.245.169 ddtp.debian.net

2016 年 8 月末から、 https://ddtp2.debian.net でアクセス出来るようになりました。\*<sup>22</sup>

## 6.18 補足2

• 小学生でも英語が必修なのに、翻訳する意味はあるのですか?

パッケージのメンテナは英語が苦手な人にも使ってもらいたいと思っているでしょう。

• 私の専門分野のパッケージはありますが、日本語はオワコンです。翻訳する意味はあるのですか?

英語が苦手な人でもパッケージを使ってくれるようになれば、その人があなたの専門分野に興味を持ってく れる可能性が高まると思います。

• 私の専門分野にある最近の用語には日本語訳がありません。

その用語の部分はカタカナが業界的に通用しているならカタカナで、そうでないなら無理に訳さず英語のま までいいと思います。

 $<sup>^{\</sup>ast 22}$  https://lists.debian.org/debian-i18n/2016/08/msg00011.html

# 7 go-apt-cacher/mirror

湯谷啓明

サイボウズ株式会社 湯谷啓明です。作成、公開した go-apt-cacher/go-apt-mirror について説明します。

## 7.1 go-apt-cacher/go-apt-mirror とは

- Apt レポジトリに特化したキャッシュプロキシ/ミラーリングツール
- ハッシュの一致をチェックするので壊れない!
- Go 製

GitHub で公開してます。https://github.com/cybozu-go/aptutil

## 7.2 Cybozu & Debian

- サーバは Ubuntu
- サービスに必要なコンポーネントは deb パッケージ化して Apt レポジトリ (JFrog Artifactory) で集中管理
- 参考: 社内利用のための deb パッケージング入門 http://blog.cybozu.io/entry/2016/05/16/111500

## 7.3 JFrog Artifactory

- アーティファクト管理ツール
- OSS 版は Maven レポジトリ機能のみ
- 商用版は apt, yum, npm, PyPI, ... と多種
- REST API で CI/CD ツールと連携できる
- リモートレポジトリのミラーはできない
- リバースプロキシ兼キャッシュはできる

## 7.4 キャッシュ/ミラーが必要な理由

- Artifactory の負荷を減らしたい
- パッケージダウンロードを速くしたい
- 更新頻度が高く、ファイルが少ない
   必要なものだけキャッシュ
- 更新頻度が低く、ファイルが多い (レポジトリの一部を)まるごとミラー

## 7.5 go-apt-cacher の特徴

- Release や Packages からチェックサム情報を抽出
  - ダウンロードしたファイルのチェックサムが合わなければ破棄
  - Release は定期的にチェックし、チェックサムが更新されたファイルのキャッシュは自動破棄
- HTTP ヘッダは無視
  - キャッシュした Release を定期的に自動更新
  - 他のファイルは上記チェックサム変更の自動破棄で対応
- Go なので速い
  - 同時 1,000 クライアントも余裕
  - キャッシュ済みファイルは 1ms 以下で処理

#### 7.6 go-apt-mirror の特徴

- 部分的ミラー
  - Suite, Section, Architecture, Source を指定
  - 必要なものだけミラー
- rsync より高速な更新
  - インデックスを先に処理してファイルのチェックサム情報を入手
  - 以前と変化がなければローカルファイルシステム上で再利用
- 不完全なミラーは決して作らない
  - インデックスおよびファイルのチェックサムはすべてチェック
  - 正しいセットが作れない場合、ロールバック
- ミラーの更新がアトミック
   更新中の状態は決して見せない

# 7.7 実装で大変だったところ

Debian Control File のパース

- 存在しないファイルが Release/Packages に記載されている
- ファイルはひとつのインデックスだけに書かれているとは限らない
- ファイルが存在しないと違うフォーマットのファイルを返してくる
- ...

## 7.8 Feedbacks are welcome!!

リリースしたばかりなのでバグもあると思いますが、よければ使ってみてください 弊社での必要性に応じて作られたツールで、既存のツールを置き換えるものではありません

## 8 sbuild と debci を触ってみた

## 8.1 はじめに

Debian パッケージを作成する際に「ビルド時の依存関係を満たしているのか」だけではなく「そのパッケージの 更新による他のパッケージへの影響/不具合はどうなのか」という懸念があります。今回の発表では、pkg-ruby-extra team で導入している sbuild + debci という環境を例に、それらの懸念をどの様に解消しているのかについて簡単 にまとめてみました

佐々木洋平

### sbuild とは?

https://wiki.debian.org/sbuild より抜粋:

**sbuild** is used on the official buildd network to build binary packages for all supported architectures. It can also be used by individuals to test that their package builds in a minimal installation of Debian Unstable. In particular, this helps ensure that you haven't missed any build dependencies.

The main alternative to sbuild is pbuilder combined with cowbuilder.

というわけで、sbuild は「公式の buildd network で使用されている」クリーンルームビルド環境です。同様の パッケージ作成環境には (上記 Wiki で触れられている通り) pbulider や cowbuilder があります\*<sup>23</sup>。

## debci とは?

debci は Debian パッケージに関する CI – Continuous Integration つまり「ビルドやテストを継続的に実行する ことで品質改善を行なう」ためのフレームワークです。Debian パッケージの「テスト」に関する枠組みとしては既に

- piuparts による「パッケージとしての品質」のテスト
- autopkgtest による「ソフトウェアとしての」テスト

があります。これらに加えて debci では

パッケージを更新した際に、依存する他のパッケージが正しく動作するか

<sup>\*&</sup>lt;sup>23</sup> pbulider や cowbuilder に対して、sbuild は何が嬉しいのでしょうか? 調べてみましたが、どちらかに絶対的な優位性がある、といっ た話でも無さそうです。慣れている方を使ったら良いのではないか、という無難な結論になりそうですね。強いて言えば、既に触れた通り 「公式配布パッケージ群のビルド環境である」という点が優位な所でしょうか?

を「テスト」しています (パッケージが正しく動作するか、は autopkgtest を用いています)。実際に debci が動作 している状況は https://ci.debian.net/ で見ることができます。

		Deb	an Contir	uous li	ntegrat	ion - Fir	efox									
O Debian Continuo	ue Integration	Home	Statue	Docu	mentatir	n D	ata									
( bobian committee	do intogration		010100													
News	2															
		Pac	kages													
upyfai 0.12.0+dfsg-5 unstable/amd64	PASSED on	3	4 a	b	с	d	e 1	9	ł	n i	i	k	1	lib3		
2016-10-21 23:12:11 UT	C   02 hour(s) ago	liba	libb	libc	libd	libe	libt	H	bg	libh	libi	libj		libk		
₩postgresql-9.6 9.6.0	0-1 FAILED on	libl	libm	libn	libo	libp	lib	1	ibr	libs	libt	lib	J	libv		
unstable/amd64 (prev 2016-10-21 22-14-4210	iously passed)	libw	libx	liby	libz	m	n	0	р	q	r	s	t	u		
	o   oo noar (o) ago	v	w ×	v	7											
illava-server 2016.9-1	PASSED on															
2016-10-21 21:57:40 UT	C   03 hour(s) ago															
Infonionaireuite 0.4.2	EALED on															
unstable/amd64 (prev	iously passed)															
2016-10-21 21:55:07 UT	03 hour(s) ago															
<b>₩</b> ruby2.3 2.3.1-5 FAI	ED on															14
unstable/amd64 (prev	iously passed)															

⊠ 1 debci Ø screenshot: https://ci.debian.net/

## 8.2 sbuild, debci *Φ* setup

pkg-ruby-extras team の setup スクリプトは以下の通りです:

```
#!/bin/sh
set -exu
sudo apt-get install -qy \
   autopkgtest \
  build-essential \
  gem2deb \
  git \
git-buildpackage \
  myrepos \
  quilt \
  sbuild \
  lxc \
debci \
sudo mkdir -p /root/.gnupg # To work around #792100
sudo sbuild-update --keygen
sudo sbuild-adduser $USER
chrootname=unstable-$(dpkg --print-architecture)-sbuild
chroot=/srv/chroots/$chrootname
if schroot --list --all-source-chroots | grep unstable-amd64-sbuild; then
else
sudo sbuild-createchroot unstable $chroot http://httpredir.debian.org/debian
fi
fi
done
if ! grep -q /var/cache/apt/archives /etc/schroot/sbuild/fstab; then
    sudo sh -c 'echo /var/cache/apt/archives /var/cache/apt/archives none rw,bind 0 0 >>/etc/schroot/sbuild/fstab'
fi
# configure lxc networking if needed
if grep -q '^lxc.network.type\s*=\s*empty' /etc/lxc/default.conf; then
sudo apt-get install -qy libvirt-clients libvirt-daemon-system iptables ebtables dnsmasq-base
if ! (sudo virsh net-list | grep -q default); then
sudo virsh net-start default
ende virsh net-start default
     sudo virsh net-autostart default
  fi
  sudo sed -i -e '/lxc.network.type/d' /etc/lxc/default.conf
   sudo tee --append /etc/lxc/default.conf <<EOF</pre>
lxc.network.link = virbr0
lxc.network.flags = up
EOF
sudo tee /etc/sudoers.d/lxc <<EOF
%sudo ALL = NOPASSWD:SETENV: /usr/bin/lxc-*, /usr/bin/timeout</pre>
EOF
fi
sudo debci setup
```
#### 8.3 sbuild: 1st step

前半部分を参考に、先ずは sbuild を設定してみます。何も考えなければ、

```
sudo mkdir -p /root/.gnupg # To work around #792100
sudo sbuild-update --keygen
sudo sbuild-adduser $USER
chrootname=unstable=$(dpkg --print-architecture)-sbuild
chroot=/srv/chroots/$chrootname
sudo sbuild-createchroot unstable $chroot http://ftp.jp.debian.org/debian
for conf in $(grep -1 ^union-type=' /etc/schroot.d/*-sbuild*); do
if ! grep -q "^union-overlay-directory=" "$conf" ; then
echo 'union-overlay-directory=/dev/shm' | sudo tee --append "$conf"
fi
done
if ! grep -q /var/cache/apt/archives /etc/schroot/sbuild/fstab; then
sudo sh -c 'echo /var/cache/apt/archives /var/cache/apt/archives none rw,bind 0 0 >>/etc/schroot/sbuild/fstab'
fi
```

でしょうか?<sup>\*24</sup> 上記によって、

- union-fs による差分管理
- /var/cache/apt/archives を親と共有 (bind mount)

した unstable-\$arch の sbuild 環境ができあがります。eatmydata や ccache を加えておきたい場合には、

```
sudo sbuild-createchroot \
    --include=eatmydata,ccache,gnupg \
    unstable /srv/chroot/unstable-amd64 http://ftp.jp.debian.org/debian
```

```
の様に --include で含めておくパッケージを指定しておきます。
```

sbuild の簡単な使い方を簡単にまとめると以下の通りです。

● 環境設定: /.sbuildrc に以下を書いておきます<sup>\*25</sup>

```
# -*- mode: cperl-mode
## basic
# architecture: all も build する
$build arch all = 1:
  デフォルトで source packag も build する
$build_source = 1;
## debsign
# GPG key id
$key_id = 'Your GPG Key ID';
# distribution の指定
$distribution =
                 'unstable';
## lintinan
# build 後に lintian を走らせる
$run_lintian = 1;
# lintian の option. お好みで。
$lintian_opts = ['-i', '-I'];
## piuparts
# build 後に piuparts を走らせる
$run_piuparts = 1;
# piuparts の option。ここでは schroot で走らせる
$piuparts_opts = ['--schroot', 'unstable-amd64-sbuild'];
```

環境更新

% sudo sbuild-update -udcar unstable-\$arch

-udcar を指定することで"apt-get update; dist-upgrade; clean; autoclean; autoremove"が実行されます。

• パッケージのビルド: dsc がある場合

% sbuild \*.dsc

パッケージのビルド:パッケージのソースディレクトリで

 $<sup>^{*24}</sup>$  GPG の鍵生成に時間がかかる場合には havedged 等で entropy を増やすと良いでしょう

 $<sup>^{*25}</sup>$  build 後に autopkgtest を走らせる設定もあるんですが、これは後程。

% sbuild

• 偶に schroot の session が残るので、綺麗にしましょう:

% sudo schroot --end-session --all-sessions

ざっくりとまとめると、こんな感じです。pbuilderの様に「tar.gzの展開/chroot」という段階が無いため、体感時間は短いです。また、結果の出力が fancy ですね。

8.4 debci: 1st step

次に debci を設定してみます。スクリプトの該当部分は以下:

```
# configure lxc networking if needed
if grep -q 'lxc.network.type\s*=\s*empty' /etc/lxc/default.conf; then
sudo apt-get install -qy libvirt-clients libvirt-daemon-system iptables ebtables dnsmasq-base
if ! (sudo virsh net-list | grep -q default); then
sudo virsh net-start default
fi
sudo virsh net-autostart default
fi
sudo sed -i -e '/lxc.network.type/d' /etc/lxc/default.conf
sudo tee --append /etc/lxc/default.conf <<EOF
lxc.network.type = veth
lxc.network.flags = up
EOF
sudo tee /etc/sudoers.d/lxc <<EOF
%sudo ALL = NOPASSWD:SETENV: /usr/bin/lxc-*, /usr/bin/timeout
EOF
fi
sudo debci setup
```

これは

- lxcのnetworkはdnsmasqで良きにはからう
- debci の lxc backend の設定を行なう

といった塩梅です。このうち lxc 自体の設定は debci setup の実態は/usr/share/debci/backends/lxc/create-testbed です。この中で lxc の設定が行なわれています。

… といった所で、資料執筆の時間が過ぎてしまいました。実際に debci を触ってアレコレするのは当日の実演&次の機会にまとめたいと思います。

# 9 debexpo(mentors.d.n)をハックする には

林健太郎 (kenhys)

## 9.1 はじめに

そのへんに生えている Debian 使いが Debian 公式パッケージを更新するにあたっては、更新したパッケージを Debian 開発者にスポンサーしてもらう必要があります。

その際のパッケージのアップロード先には、mentors.debian.net という Web サイトがよく使われています。 今回は mentors.debian.net をハックする機会があったので、その内容を紹介します。\*<sup>26</sup>

## 9.2 debexpoとは?

http://mentors.debian.net(以下 mentors.d.n) を支えるウェブアプリケーションのことです。リポジトリ名が debexpo になっています。



debexpo の公式サイトは https://alioth.debian.org/projects/debexpo/ です。以前開発の中心となって いたサイト https://workaround.org/project/debexpo には debexpo の名前の由来が次のように記載されてい ます。

The new project was called "debexpo" because it was supposed to become an exposition for Debian packages.

パッケージが一同に会する展覧会とでもいったところでしょうか。

<sup>\*&</sup>lt;sup>26</sup> 2016 年 8 月現在の情報で、現在では若干記述が古くなっている箇所があります。該当箇所には注記を入れてあります。

### 9.2.1 debexpo 概要

debexpoの位置付けがどういうものかなんとなくわかったところで、もうすこし具体的な紹介をしましょう。 debexpoの特徴をいくつか紹介すると以下があげられます。

- Python 製
- Pylons<sup>\*27</sup>フレームワーク採用
- テンプレートエンジンは Mako\*28

この Pylons というフレームワーク、あまり知らない人がいるかも知れませんので補足しておくと、

- Railsっぽいフレームワーク
- 2011 年にメンテナンスモード入りした枯れたフレームワークです
- 後継として Pyramid というフレームワークが開発されています

という特徴があります。



## The sp

debexpoには、実はゆるキャラが設定されています。Web サイトの左上隅の赤枠で囲っているところに何かいますよね、それです。ただし、詳細は不明です。SUUMO<sup>a</sup>のスーモ<sup>b</sup>には片思いのスモミとの想像上の子供スモルがいるという設定<sup>c</sup>があるくらいなので、このキャラにも何かありそうな気もしますが。。。

<sup>a</sup>不動産情報サイト。http://suumo.jp/

<sup>b</sup>マスコットキャラクター。スーモの部屋 http://suumo.jp/edit/suumo-heya/ という特設ページがある。

 ${}^c \; {\tt http://suumo.jp/edit/suumo-heya/character/suumo_character.pdf}$ 

## 9.3 なぜ debexpo をハックする必要が?

パッケージをスポンサーしてもらうときに使う mentors.d.n にはいくつかお薦めな点があります。

- パッケージのチェックもしてくれる
- RFS のテンプレートも生成してくれる

 $<sup>^{*27}</sup>$  http://www.pylonsproject.org/

<sup>\*28</sup> 

Package ve	ersions	
Version 6.0.4-	1	
Information		
Version:	6.0.4–1 (View RFS template)	
Uploaded:	2016-06-07 15:07	
Source package:	https://mentors.debian.net/debian/pool/main/g/groonga/groonga_6.0.4	-1.dsc
Section:	database	
Priority:	optional	
QA information Toggle [ <u>All[Info</u> ] + Buildsystem: F + Homepage cor - Package has li	Package uses debhelper ntrol field present intian warnings	

スクリーンショットからもわかるように、QA information の欄に Lintian の警告などが表示されるようになってい





また、スクリーンショットの「View RFS template」のリンクをたどると、RFS のテンプレートも生成してくれるというのも嬉しいポイントです。

なので、あとはこの  $\mathrm{RFS}$  テンプレートをもとにして、メールするだけになっています。 $^{*29}$ 

実際の RFS template を見てみましょう。

<sup>\*29</sup> とはいうものの、嘘ではないが、正確でもない。



## [fill in] の文字がちらほらあるのがわかりますね。

It builds those binary packages:	es:	package	binary	those	builds	It
----------------------------------	-----	---------	--------	-------	--------	----

groonga-normalizer-mysql - MySQL derived normalizer for Groonga

To access further information about this package, please visit the following URL:

https://mentors.debian.net/package/groonga-normalizer-mysql

Alternatively, one can download the package with dget using this command:

dget -x https://mentors.debian.net/debian/pool/main/g/groonga-normalizer-mysql/groo

More information about hello can be obtained from  ${\tt https://www.example.com}.$ 

Changes since the last upload:

[your most recent changelog entry]

## これだけではありません。ほかにも穴埋めが必要です。



Subject:に種別を書かないといけません。



Severity:を書かないといけません。

Dear mentors,		
I am looking fo	or a sponsor for my package "groonga-normalizer-mysql"	
* Package name Version Upstream Autho * URL * License Section	: groonga-normalizer-mysql : 1.1.2 or : [fill in name and email of upstream] : [fill in URL of upstreams web site] : [fill in] : 1105	

Upstream, URL, License:を書かないといけません。

dget -x https://mentors.debian.net/debian/pool/main/g/groonga-normalizer-mysq

More information about hello can be obtained from https://www.example.com.

Changes since the last upload:

[your most recent changelog entry]

Changelog を書かないといけません。 これで終わりでしょうか。いいえ違います。



さりげなく埋めこまれた hello と example.com が残っています。

ようやくあれこれ直し終えました。これでメールが出せると、思うかも知れません。事実私も最初はそう思いました。

ここで問題になるのが、先頭に埋めこまれたスペース2つです。

Package: sponsorship-requests Severity: normal [important for RC bugs, wishlist for new packages]

Dear mentors,

ここで、上記はコマンドメールであることを思いださねばなりません。つまり、そのままメールするともちろんエ ラーになります。

したがって、「なぜ debexpo をハックするのか?」に対する答えは、「RFS テンプレートの残念っぷりをどうにか したい」から、ということになります。

Summary*: RFS template feature requests	
Detailed description	
In the RFS bug template, pleas	n.
* Don't indent the pseudo-hea will not be able to parse them of * Remove the line "More inform * Automatically fill in the Subjee * Same for Severity, if possible. * Parse information from the pa debian/control. Upstream Auth is http://www.debian.org/doc/f * Fill in the "Changes since the Finally, it would be great if a ma <a ;,="" by="" href="mailto:submit@bugs.&lt;br&gt;RFS&lt;/a&gt;&lt;/td&gt;&lt;td&gt;lers by two spaces. When copied and pasted to an email message, they need to be removed manually, as the control server therwise.&lt;br&gt;ation about hello can be obtained from http://www.example.com." it="" or="" replace="" sensible.<br="" something="">t part that currently reads "[put in ITP, ITA, RC, NMU if applicable]" by parsing the (WNPP et. al.) bugs that the package closes. ckage to fill in Upstream Author, URL and License automatically. URL should probably be filled from the Homepage field in or and License could be gathered from the Upstream-Name and Upstream-Contact fields in debian/copyright if the package ackaging-manuals/copyright-format/1.0/ or previously DEP-5 compliant. (License is probably a bit more complex.) ast upload: \n\n [your most recent changelog entry]" automatically by parsing the changelog. iilto: link was provided for opening a new message with the Subject and Body fields filled in automatically, i.e. something like debian.org?subject=RFS%3A%20foo%2F1.5-1%20%5BITP%5D&amp;body=Package%3A%20sponsorship-requests&gt;Send</a>	

Alioth をみてみると、同じ思いの人がいました。それは Alioth の tracker で4年も前に通った道だ、という。

## 9.4 どうやってハックしたのか?

おおむね、以下の流れで debexpo をハックすることになりました。

- ・ upstream 探し
- ドキュメント探し
- まずは動かしてみる
- あたりをつけて修正
- そして PR へ

特別なことは何もなくて、よくあるフリーソフトウェアの修正です。

## ・コラム:debexpoの歴史について -

debexpoは2003年に最初のコードがPerlで書かれました。しかし、機能拡張の要望に応えたりしていくには支 障があったため、Pythonで書き直されたという経緯があります。debexpoの開発が活発になったのは、2008年 のことです。Google SoC に採択されたため、Jonny Lamb 氏らにより一気に開発がすすみました。 2009年から2010年はゆるやかな開発が続きました。http://expo.debian.net/が公開されたのもこのころです。 現在のmentors.d.nへと切り替えられたのは2011年のことです。その後も、2012年にはUIの改善(debexpo v2)や再度 GSoC への採択 (debexpo v3) などと開発が続いています。

このあたりの変遷について、Nicolas Dandrimont 氏による「The State of mentors.debian.net GSOC and Beyond」という発表資料に詳しく書かれているので興味がある人は参照するとよいでしょう。<sup>*a*</sup>

 $^a$  http://fr2012.mini.debconf.org/slides/debexpo.pdf

## 9.4.1 upstream 探し

$\left( \right)$	owncloud-client-110n - ownCloudSync folder synchronization - I
	mentors.debian.net - Hosting and hardware provided by <u>Waveco</u> - <u>Source code and bugs</u> - <u>Sourc</u>

debexpo の場合には、mentors.d.n 下部にリンクがきちんとあるので、Alioth をみればいいとわかりました。



ただ、どうやら最近はコミットがないのが不安になりました。よく使われているならそこそこメンテされているイ メージがあったからです。実際にはそうでもありませんでした。



あとから、GitHubのほうが実は新しい\*30ことがわかりました。

実際の運用としては、Alioth が masterhttps://alioth.debian.org/projects/debexpo/ で、GitHub のを マージという運用になっているようです。

## 9.4.2 ドキュメント探し

リポジトリの docs/\*にドキュメントが整備されていました。インストール手順は docs/installing.rst を参照すれば よいとわかりました。ただ、残念なことにその内容の一部はリンク先が 404 になってしまっていました。

## 9.4.3 まずは動かしてみる

ドキュメントから、セットアップ方法は3種類あることがわかりました。

- 既存システムにインストール
- virtualenv でインストール
- VirtualBox でインストール

まずは VirtualBox で試してみることにしました。環境を分けたいのがその選択理由です。ただ、Vagrantfile がアレな状態であることがわかりました。



Debian 7.6 (2014 年 7 月 12 日)?になっていました。Debian 7.10 がもうすでにでているご時世にも関わらずです。 vagrant up してみるとまた残念な状態でした。

```
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
=>> default: Box 'chef/debian-7.6' could not be found. Attempting to find and install...
default: Box Provider: virtualbox
default: Box Version: >= 0
The box 'chef/debian-7.6' could not be found or
could not be accessed in the remote catalog. If this is a private
box on HashiCorp's Atlas, please verify you're logged in via
'vagrant login'. Also, please double-check the name. The expanded
URL and error message are shown below:
URL: ["https://atlas.hashicorp.com/chef/debian-7.6"]
Error: The requested URL returned error: 404 Not Found
}
```

box が見つからなくてコケていました。そこで、PR#32 で修正しました。失敗していたのは、Bento project に移行していたせいでした。

<sup>\*30</sup> https://github.com/debexpo/debexpo



起動して、ログインするには次のようにします。

```
$ vagrant up --provision
$ vagrant ssh
```

vagrant ssh して paster コマンドを実行してサーバーを起動します。

\$ cd debexpo \$ . venv/bin/activate \$ paster serve development.ini

このようにすると、5000 ポートでサーバーを起動することができます。



Getting your package into Debian

これによりブラウザでアクセス可能になります。

次にユーザーの追加をします。方法は2つあって、ブラウザ経由で追加するのと、JSON をもとに追加するやりかたがあります。

Sign	up	for	an	account
------	----	-----	----	---------

Account details	
Full name:	
E-mail:	
Password:	
Confirm password:	
Account type:	Maintainer OSponsor
Submit	

JSON で追加するなら次のような内容のファイルを用意します。

{	
	"realname":"Hayashi Kentaro",
	"password": "password",
	"email":"hayashi@clear-code.com"
}	

ユーザー追加用のスクリプトが用意されているので、それを利用します。

\$ python ./bin/user\_importer.py -i development.ini -u user.json

次にアカウントの有効化をします。アカウントを有効にするには2つ設定をする必要があります。

- verification (ログインに必要)
- dmup (アップロードに必要)

verification の設定は次のようなクエリを実行することで行います。



verification を空にすることで、メールによる確認プロセスを迂回することができます。

もう一つ、DMUP とはマシン使用ポリシーのことです。開発で使うだけなので、次のようなクエリを実行することで dmup フィールドを更新して、同意したことにします。



ここまでできたら、あとはアップロードするために、.dput.cfの設定をします。

```
[debexpo]
fqdn = localhost:5000
incoming = /upload/kenhys@gmail.com/password
method = http
allow_unsigned_uploads = 0
```

## 用意できたら、実際にアップロードを試してみましょう。

Uploading to debexpo (via http to localhost:5000): Uploading groonga\_6.0.2-1.dsc: Upload failed: 500 Internal Server Error

と思ったら、あっさり 500 Internal Server Error に遭遇しました。



## 残念なことに、あるべきディレクトリがないというオチでした。 そこで、PR#34 で修正してとりこんでもらいました。

Ensur n Merged	re to make incoming directory #34 paulproteus merged 1 commit into debexpormater from kentys:ensur	-to-make-incoming-directory 28 days ago	
Conve	rsation 2 Commits 1 🕑 Files changed 1		
	kenhys commented on 3 May /tmp/debexpo is created by paster setup-app development.ini, but no /tmp/debexpo/pub, so it is needed not to cause 500 internal server en when uploading package.	+ (j) /	

PRを出してみたときに気づいたのですが、最後にテストが通ったの8ヶ月前というオチがついていました。

debexpo / debexpo 🖓 🔤 🖬 🕼			
urrent Branches Build History Pull Requests			More option
X master Merge pull request #37 from kenhys/drop-py26	- <b> #120 failed</b>	() 4 min	37 sec iys ago
master     Merge pull request #34 from kenhys/ensure-to-make-incoming-directory     Asheesh Laroia	- <b> #119 errored</b> 7 18384a8	() 5 min 77 20 da	2 sec iys ago
✓ master Merge pull request #31 from ukleinek/master ⓐ Asheesh Laroia	- <b>O</b> - #113 passed	() 9 min 17 8 mor	48 sec nths ago

これはなぜかというと Travis-CI の環境の変化に誰も気いていない状態だったためです。しばらくコミットがなされていないプロジェクトではありがちです。そこで、PR#38 でテストが通るように修正しました。



## また、Python2.6 で CI はもう不要なので、PR#37 で修正も行いました。

Drop n <sub>Mergeo</sub>	python 2.6 support on Travis-CI #37		
다. Conve	ersation 3 Commits 1		
1	kenhys commented on 14 May	+(ii) d <sup>a</sup> *	
	No description provided.		
	📀 🧑 Drop python 2.6 support	<b>X</b> 10699b2	

ここまで修正して、ようやくパッケージを取り込むところまでたどりつきました。パッケージの取り込みは次のコ マンドを実行します。

\$ ./bin/debexpo\_importer.py \
 -c /tmp/debexpo/growl-for-linux\_0.8.5-1\_source.changes -i development.ini --skip-gpg-check --skip-email

インポートスクリプトを実行したら、あっさり取り込みできずにトレースを吐きました。

Traceback (most recent call last): File "./bin/debexpo\_importer.py", line 60, in i.main() File "/home/vagrant/debexpo/debexpo/importer/importer.py", line 473, in main gpg = get\_gnupg() File "/home/vagrant/debexpo/debexpo/lib/utils.py", line 119, in get\_gnupg return gnupg.GnuPG(config['debexpo.gpg\_path'], File "/home/vagrant/debexpo/venv/local/lib/python2.7/site-packages/paste/registry.py", line 146, in getitem return self.\_current\_obj()[key] KeyError: 'debexpo.gpg\_path'

gpgの検証をスキップするオプションが期待するように動作していなかったので、オプションを正しく解釈するように PR#39 で修正しました。



これでようやく、取り込んだパッケージを Web の画面から確認することができるようになりました。



## 9.4.4 あたりをつけて修正

パッケージをアップロードして、画面から確認できるようになったので、次に本来やりたかった debexpo 自体の改善に取り組みました。まずはディレクトリ構成からあたりをつけることにしました。

config			
controllers			
cronjobs			
importer			
i18n			
lib			
model			
plugins			
public			
templates			
tests			

手がかりとなるのは URL です。

Package versions								
Version 0.8.5-1								
Information Version:	0.8.5 1 (View RFS template)							
Uploaded:	2016 26 19 21:12							
Source package:	http://localhost:5000/debian/pool/n							
Section:	gnome							
Priority:	optional							

知りたいのは View RFS Template のリンク先です。

http://localhost:5000/sponsors/rfs-howto/xxxx

リンク先がわかったので、対応するルーティングを処理するコントローラの実装を探してみたら controllers/sponsor.py を見ればいいことがわかりました。

これに対応する Mako のテンプレートは templates/sponsor/rfs\_howto.mako にあることがわかりました。なんとなく見覚えがありますね。

```
Package: sponsorship-requests
Severity: normal [important for RC bugs, wishlist for new packages]
Dear mentors,
%if c.package:
   I am looking for a sponsor for my package "${ c.package.name }"
%else:
   I am looking for a sponsor for my package "hello":
   %endif
```

やりたいことは、RFS テンプレートから [fill in] を撲滅し、mailto:リンクを生成して RFS を出すときの手間を軽 減することです。当初の目論見では、\$ c.package.name とかあるので、テンプレートを書き換えればいいだけかと 思っていました。しかし、結論からいうとこの案は無理でした。というのも、必要なメタ情報を保持していないこと が明らかになったからです。持ってないものは表示できません。そのため、どうにかして情報をかき集めないといけ ないことになりました。

## 9.4.5 収集するにはどうすればいいか

まずは、インポートの処理の流れを把握する必要があります。そして、どのタイミングで収集すべきかを知らなけ ればなりません。また不足している情報は何かを知る必要もあります。

では、実際のインポート処理はどのようになっているのでしょうか。

インポート処理は、dput で mentors.d.n ヘアップロードされた時点で始まります。そして、前処理が実行され、パッ ケージのインポート処理へと続いていきます。もう少し詳しく説明すると、dput したファイルは/tmp/debexpo/pub へ保存されます。そして、インポート前処理で/tmp/debexpo へ移動されます。このとき、orig.tar.gz がなかったり すると reject メールが送られます。

インポートが完了した時点で、ソースパッケージは/tmp/debexpo/files へと移動されています。このとき、 /tmp/debexpo/files 以下には pool や dist,git ディレクトリが作成されます。また、各種パッケージの情報がインポー ト中にデータベースへと保存されるようになっています。

#### 9.4.6 収集すべきデータを確認する

収集するべきデータを確定するには、メタ情報がどのように保持されているのかを把握する必要があります。そこ で実際のテーブルを覗いてみることにしました。debexpo で使用している主なテーブルは以下の3つです。

- packages
- package\_versions
- package\_info

packages テーブルは、パッケージのマスターテーブルです。名前や説明などのメタ情報を保持しています。

```
sqlite> .schema packages
CREATE TABLE packages (
    id INTEGER NOT NULL,
    name TEXT NOT NULL,
    user_id INTEGER,
    description TEXT,
    watch_counter INTEGER,
    download_counter INTEGER,
    needs_sponsor INTEGER NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY(user_id) REFERENCES users (id)
);
```

package\_versions テーブルはパッケージの版管理のためのテーブルです。何度もアップロードするとレコードが増えていきます。

```
sqlite> .schema package_versions
 CREATE TABLE package_versions (
      id INTEGER NOT NULL,
     package_id INTEGER,
      version TEXT NOT NULL,
     maintainer TEXT NOT NULL,
      section TEXT NOT NULL,
      distribution TEXT NOT NULL,
      qa_status INTEGER NOT NULL,
      component TEXT NOT NULL,
      priority TEXT,
      closes TEXT,
      uploaded DATETIME NOT NULL,
     PRIMARY KEY (id),
     FOREIGN KEY(package_id) REFERENCES packages (id)
 );
```

package\_info テーブルは、プラグインの適用結果を管理します。各種メタ情報を保持しています。

```
sqlite> .schema package_info
CREATE TABLE package_info (
    id INTEGER NOT NULL,
    package_version_id INTEGER,
    from_plugin VARCHAR(200) NOT NULL,
    outcome VARCHAR(200) NOT NULL,
    data TEXT,
    severity INTEGER NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY(package_version_id) REFERENCES package_versions (id)
);
```

たとえば、from\_plugin にはどのプラグインかという情報を保持しています。outcome はエラーメッセージなどの 説明文を保持しています。data は汎用的に使えるように JSON データを保持しています。

実際にどんなデータが格納されているかをみてみましょう。data をうまいこと活用するとよさそうだとわかりま

すね。

```
sqlite> select * from package_info;
1|1|native|Package is not native|{"native": false}]1
2|1|maintaineremail|"Maintainer" email is the same as the uploader|{
"user-email": "hayashi@clear-code.com",
"uploader-emails": [],
"maintainer-email": "hayashi@clear-code.com",
"user-is-maintainer": true
}|1
3|1|debianqa|Package is already in Debian|{
"nmu": false,
"in-debian": true,
"is-debian-maintainer": true
}|1
```

ここまでの結果から、プラグインで追加のメタ情報を収集して、メール用のテンプレート追加し、詳細ページでメ タ情報を表示しつつ mailto:リンク生成する方針としました。

## 9.4.7 プラグインの説明

ここまでの説明で特に断りなくプラグインに言及していました。補足しておくと、debexpoはプラグインで機能拡張するようになっています。パッケージのチェックもプラグインを組み合わせて実現しています。

プラグインの作り方については、docs/writing\_plugins.rst にサンプルのプラグインの実装方法が紹介されていま す。簡単に言うと、BasePlugin クラスを継承した XXXPlugin として実装すれば OK です。

```
class FooPlugin(BasePlugin):
    def test_xxx(self):
        self.passed(outcome, data, severity)
        or
        self.failed(outcome, data, severity)
plugin = FooPlugin
```

そして、debexpo/plugins/foo.py などとして plugins ディレクトリ以下に配置することになっています。 標準で用意されているプラグインには次のようなものがあります。

\$ wo	: -l debexpo/plugins/*.py
99	debexpo/plugins/buildsystem.py
67	debexpo/plugins/changeslist.py
141	debexpo/plugins/closedbugs.py
85	debexpo/plugins/controlfields.py
185	debexpo/plugins/debianqa.py
85	debexpo/plugins/diffclean.py
63	debexpo/plugins/distribution.py
123	debexpo/plugins/getorigtarball.py
116	debexpo/plugins/lintian.py
100	debexpo/plugins/maintaineremail.py
69	debexpo/plugins/native.py
77	debexpo/plugins/notuploader.py
86	debexpo/plugins/removepackage.py
60	debexpo/plugins/ubuntuversion.py
110	debexpo/plugins/watchfile.py

#### 9.4.8 プラグインの適用方法

プラグインを実際に適用するには、設定ファイル (.ini) に記述を追加します。プラグインでは次のタイミングで処理を実行することができます。

- インポート前処理
- QA 処理
- Debian 入りした時
- インポート処理後

インポート前処理で適用するプラグインは、debexpo.plugins.post\_upload に設定します。getorigtarball プラグインがその例です。QA 処理で適用するプラグインは、debexpo.plugins.qa に設定します。lintian プラグインがその例です。

パッケージが Debian 入りしたときに適用するプラグインは、debexpo.plugins.post\_upload\_to\_debian に設定しま す。removepackage プラグインがその例です。

インポート処理後に適用するプラグインは、debexpo.plugins.post\_successful\_upload に設定します。changeslist プラグインがその例です。

## 9.4.9 プラグインの実装

だいたいわかってきたところで、実際にプラグインを実装してみました。debexpo/plugins/rfstemplate.py として、実質 100 行ないくらいで実装できました。やっていることは、debian/changelog や debian/control から必要な 情報を抽出して、package\_info テーブルにメタ情報を保持し、テンプレートを表示するときにデータをバインドして 表示するというものです。

あとは、設定ファイル (development.ini) に実装したプラグインを指定して有効にします。

debexpo.plugins.qa = ... rfstemplate ...

また、忘れずに mailto 用テンプレート (debexpo/templates/sponsor/rfs\_template.mako) も追加しておきます。

%if c.rfstemplate:	):	
Upstream Author	: : \${ c.rfstemplate['upstream-author'] }	
* URL	: \${ c.rfstemplate['upstream-url'] }	
* License	: \${ c.rfstemplate['upstream-license'] }	
%else:		
Upstream Author	: [fill in name and email of upstream]	
* URL	: [fill in URL of upstreams web site]	
* License	: [fill in]	
%endif		

実際に rfstemplate のデータを表示させる部分は package\_info テーブルから JSON データを取得してアサインす るだけ (debexpo/controllers/sponsor.py) なので、簡単です。

ここまでの成果物を PR#35\*31としてだしました。



リンクをクリックすると必要事項が埋められたテンプレートを使ってメーラーが起動するようになっています。

<sup>\*31</sup> https://github.com/debexpo/debexpo/pull/35

#### 9.4.10 PR#35 の経過

さて、PR をだしたあと、その後どうなったかについても紹介しておきます。

- May 4 @olasd さんから好意的な反応
- May 14 どうなった?とつついてみるも反応なし
- May 21 Debian 勉強会でまだマージされてない話をする
- あれやこれやでしばし放置
- June 19 @paulproteus さんをつついてみる
- June 19 20 日にみれるかもと@paulproteus さんから反応あり

paulproteus commented 2 days ago	debexpo member + 💼
Hi <b>@olasd</b> & <b>@kenhys</b> , Thanks for filing it and the initial review. I'll try to take a look on Monday (in two days). If I do	n't, then it will have to
wait until during Debconf.	

- DebConf16 までまってね、ということに
- DebConf16 終了するも、進展なし
- Aug 22 @paulproteus さんにアサインされるも放置プレイを食らう

残念ながらいまだにレビューしてもらえていません。

## 9.5 まとめ

今回は debexpo をハックするに至った経緯と、どんなハックをしたのかを紹介しました。RFS テンプレートが残 念だったのですが、プラグインを作成することで RFS テンプレートを改善することができました。ただし、まだ PR はマージされていないですし、実際にデプロイされるまでの道のりは遠そうです。

最近だと、この問題に関して、mentors.d.n ではなくクライアントツール側で改善しようという動きがあります。 debrequest というコマンドラインツールでいい感じに RFS テンプレートを生成することを目的にしているようで す。実際に試してみたい人は https://lists.debian.org/debian-mentors/2016/10/msg00206.html に開発者 によるアナウンスが投稿されているので、そちらを参照するとよいでしょう。

## 10 dh\_strip\_nondeterminism について

吉野 与志仁

## 10.1 はじめに

2015 年に debhelper の dh(1) シーケンスに組み込まれた dh\_strip\_nondeterminism(1) ですが、これが具体 的に何をやっているのか調べてみました。調べたバージョンは dh-strip-nondeterminism パッケージの 0.028-1 です。

なお、このコマンドの目的は reproducible builds (再現可能なビルド)ですが、詳細は https://wiki.debian. org/ReproducibleBuilds や https://reproducible-builds.org/ などを見ると良いでしょう。

再現可能なビルドを実現するため、このコマンドはビルドツールが生成したデータに埋め込まれた時刻などを指定 した値に書き換えます。実際に見ていきましょう。

## 10.2 具体例

### 10.2.1 ar

\*.a のうち ar archive のもの

\$ file /usr/lib/x86\_64-linux-gnu/libglib-2.0.a
/usr/lib/x86\_64-linux-gnu/libglib-2.0.a: current ar archive

dh\_strip\_nondeterminism(1) は、パッケージの debian/changelog ファイルの中に書いてある時刻を使うよう になっています。この .a ファイルを含むパッケージの debian/changelog ファイルの時刻を見てみます。



この Date: で始まる行(もしくは Timestamp: で始まる行)の値が使われます。

ファイルの中を見てみます。

```
$ env TZ=UTC 7z l /usr/lib/x86_64-linux-gnu/libglib-2.0.a | head -n 20
7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=ja_JP.UTF-8,Utf16=on,HugeFiles=on,64 bits,4 CPUs Intel(R) Core(TM) i5-4250U CPU @ 1.30GHz (40651),ASM,AES-NI)
Scanning the drive for archives:
1 file, 1973536 bytes (1928 KiB)
Listing archive: /usr/lib/x86_64-linux-gnu/libglib-2.0.a
Path = /usr/lib/x86_64-linux-gnu/libglib-2.0.a
Type = Ar
Physical Size = 1973536
SubType = a
                                                            Size Compressed Name
     Date
                       Time Attr
                                               _____
2016-11-07 23:37:05 .....
                                                            84550
                                                                                   84550
                                                                                               1.txt

      2016-11-07
      23:37:05
      ....
      4584
      41bglib_2_0_la-gallocator.o

      2016-11-07
      23:37:05
      ....
      6352
      6352
      libglib_2_0_la-gcache.o

      $ env TZ=UTC ar tv /usr/lib/x86_64-linux-gnu/libglib-2.0.a | head -n 4

% env 12-01c af tv /usr/110/x80_04=11htx/gHt/110g110=2.0.1a + Heat -
rw-r--r-- 0/0 4584 Nov 7 23:37 2016 libglib_2_0_la-gallocator.o
rw-r--r-- 0/0 5960 Nov 7 23:37 2016 libglib_2_0_la-gcompletion.o
rw-r--r-- 0/0 11464 Nov 7 23:37 2016 libglib_2_0_la-grel.o
```

#### 固められた各ファイルの

- タイムスタンプ (mtime) を指定した時刻に書き換えています。
- 所有者 (owner) を 0 に上書きしています。
- グループ (group) を 0 に上書きしています。
- パーミッション (mode) を 755 か 644 のいずれかに揃えています。

#### 10.2.2 zip

\*.zip, \*.pk3, \*.epub, \*.whl, \*.xpi, \*.htb, \*.zhfst のうち Zip archive data または EPUB document のもの

\$ file /usr/share/go-1.7/src/archive/zip/testdata/symlink.zip /usr/share/go-1.7/src/archive/zip/testdata/symlink.zip: Zip archive data, at least v1.0 to extract \$ file /usr/share/debian-reference/debian-reference.ja.epub /usr/share/debian-reference/debian-reference.ja.epub: Zip archive data, at least v1.0 to extract

#### 調べたパッケージの時刻はそれぞれ

\$ dpkg -S /usr/share/go-1.7/src/archive/zip/testdata/symlink.zip golang-1.7-src: /usr/share/go-1.7/src/archive/zip/testdata/symlink.zip \$ dpkg-parsechangelog -1 /usr/share/doc/golang-1.7-src/changelog.Debian.gz | grep ^Date: Date: Thu, 20 Oct 2016 09:10:47 +1300

\$ dpkg -S /usr/share/debian-reference/debian-reference.ja.epub debian-reference-ja: /usr/share/debian-reference/debian-reference.ja.epub \$ dpkg-parsechangelog -1 /usr/share/doc/debian-reference-ja/changelog.gz | grep ^Date: Date: Mon, 17 Oct 2016 22:28:00 +0900

です。

### 中身を見てみます。

<pre>\$ env TZ=UTC 7z l /usr/share    Date Time Attr</pre>	/go-1.7/src/archive/z: Size Compresse	zip/testdata/symlink.zip   tail -n 5 sed Name
2016-10-19 20:10:47	9	9 symlink
2016-10-19 20:10:47	9	9 1 files

\$ zipinfo -v /usr/share/go-1.7/src/archive/zip/testdata/symlink.zip | tail -n 36 Central directory entry #1: symlink offset of local header from start of archive: (0000000000000000h) bytes file system or operating system of origin: version of encoding software: Unix 3.0 minimum file system compatibility required: minimum software version required to extract: MS-DOS, OS/2 or NT FAT 1.0 compression method: none (stored) file security status: extended local header: not encrypted 2016 Oct 19 20:10:46 file last modified on (DOS date/time): file last modified on (UT extra field modtime): 2016 Oct 20 05:10:47 loca file last modified on (UT extra field modtime): 2016 Oct 20 05:10:47 loca 32-bit CRC value (hex): 8e9efad1 2016 Oct 20 05:10:47 local compressed size: 9 bytes uncompressed size: length of filename: 9 bytes 7 characters length of extra field: length of file comment: 24 bytes 0 characters disk number on which file begins: disk 1 apparent file type: Unix file attributes (100755 octal): binary -rwxr-xr-x MS-DOS file attributes (00 hex): none The central-directory extra field contains:
A subfield with ID 0x5455 (universal time) and 5 data bytes. The local extra field has UTC/GMT modification/access times.
A subfield with ID 0x7875 (Unix UID/GID (any size)) and 11 data bytes: 01 04 00 00 00 00 04 00 00 00 00. There is no file comment.

\$ env TZ=JST-9 7z 1 /usr/share/debian-reference/debian-reference.ja.epub | head -n 24 | tail -n 10 Date Time Attr Size Compressed Name 2016-10-17 22:28:00 D.... 0 O META-INF 2016-10-17 22:28:00 ..... 2016-10-17 22:28:00 D.... 255 175 META-INF/container.xml 0 OEBPS 0 2016-10-17 22:28:00 ..... 10451 3596 OEBPS/apa.html 2016-10-17 22:28:00 ..... 2016-10-17 22:28:00 ..... 16458 OEBPS/bk01-toc.html 45365 OEBPS/ch01.html 123568 279742 45365 2016-10-17 22:28:00 ..... 306201 49878 DEBPS/ch02.html 2016-10-17 22:28:00 ..... 86922 15031 DEBPS/ch03.html \$ env TZ=JST-9 7z l /usr/share/debian-reference/debian-reference.ja.epub | tail -n 4 2016-10-17 22:28:00 ..... 2016-10-17 22:28:00 ..... 96004 19619 OEBPS/toc.ncx 20 mimetype 20 2016-10-17 22:28:00 425235 23 files, 2 folders 2593315

#### 固められた各ファイルの

- 並びを名前順に直しています。
- DOS 時刻のフィールドを指定した時刻に書き換えています。しかし、精度が2秒しかないようです。
- 属性を 755 か 644 のいずれかに揃えています。
- central directory header (  $\succeq$  local header )  $\sigma$ 
  - ID 0x5455 (UT universal time) のフィールドに時刻が入っているので、指定した時刻に書き換えています。
  - ID 0x7875 (ux Unix UID/GID) のフィールドにある UID と GID を 0 に上書きしています。

#### 10.2.3 jar

\*.jar, \*.war, \*.hpi, \*.apk のうち Java archive または Zip archive のもの

\$ file ./usr/share/java/commons-lang3.jar ./usr/share/java/commons-lang3.jar: Zip archive data, at least v1.0 to extract

\$ dpkg-parsechangelog -1 ./usr/share/doc/libcommons-lang3-java/changelog.Debian.gz | grep ^Date: Date: Thu, 20 Oct 2016 19:08:15 +0200

## 中を見てみます。

\$ env TZ=UI	C 7z 1 .	/usr/sha	re/java/commons	-lang3.jar	head -n 24   tail -n 10
Date	Time	Attr	Size C	ompressed	Name
2016-10-20	17:08:14	D	0	0	META-INF
2016-10-20	17:08:14		1844	732	META-INF/MANIFEST.MF
2016-10-20	17:08:14		11358	3949	META-INF/LICENSE.txt
2016-10-20	17:08:14		301	187	META-INF/NOTICE.txt
2016-10-20	17:08:14	D	0	0	META-INF/maven
2016-10-20	17:08:14	D	0	0	META-INF/maven/org.apache.commons
2016-10-20	17:08:14	D	0	0	META-INF/maven/org.apache.commons/commons-lang3
2016-10-20	17:08:14		91	83	META-INF/maven/org.apache.commons/commons-lang3/pom.properties
\$ env TZ=U1	C 7z 1 .	/usr/sha	ce/java/commons	-lang3.jar	tail -n 4
2016-10-20	17:08:14		3231	1289	org/apache/commons/lang3/tuple/Pair.class
2016-10-20	17:08:14		3190	1250	org/apache/commons/lang3/tuple/Triple.class
2016-10-20	17:08:14		986869	404745	265 files, 19 folders

<pre>\$ unzip -p ./usr/share/java/commons-lang3-3.5.jar META-INF/MANIFEST.MF</pre>
Manifest-Version: 1.0
Bundle-License: http://www.apache.org/licenses/LICENSE-2.0.txt
Bundle-SymbolicName: org.apache.commons.lang3
Archiver-Version: Plexus Archiver
Implementation-Vendor-Id: org.apache
Specification-Title: Apache Commons Lang
Bundle-DocURL: http://commons.apache.org/proper/commons-lang/
<pre>Include-Resource: META-INF/LICENSE.txt=LICENSE.txt,META-INF/NOTICE.txt =NOTICE.txt</pre>
Require-Capability: osgi.ee;filter:="(&(osgi.ee=JavaSE)(version=1.5))"
Export-Package: org.apache.commons.lang3;version="3.5",org.apache.comm
ons.lang3.builder;version="3.5",org.apache.commons.lang3.concurrent;v ersion="3.5",org.apache.commons.lang3.event;version="3.5",org.apache. commons.lang3.exception;version="3.5",org.apache.commons.lang3.math;v ersion="3.5" org.apache.commons_lang3.muthle:version="3.5" org.apach
e commons langs reflect:version="3 5" org anache commons langs text:v
ersion="3.5" org anache commons lang3 text translate.version="3.5" or
g anache commons langs time version="3.5" org anache commons langs tu
nle:version="3.5"
Bundle-Name: Apache Commons Lang
Implementation-Title: Apache Commons Lang
Bundle-Description: Apache Commons Lang, a package of Java utility cla
sses for the classes that are in java.lang's hierarchy, or are consi dered to be so standard as to justify existence in java.lang.
Implementation-Version: 3.5
Specification-Vendor: The Apache Software Foundation
Bundle-ManifestVersion: 2
Bundle-Vendor: The Apache Software Foundation
Tool: Bnd-2.4.1.201608301338
Implementation-Vendor: The Apache Software Foundation
Bundle-Version: 3.5.0
X-Compile-Target-JDK: 1.5
<pre>Implementation-Build: \${scmBranch}@r\${buildNumber}; 2016-10-20 17:08:1 5+0000</pre>
X-Compile-Source-JDK: 1.5
Created-By: Apache Maven Bundle Plugin
Build-Jdk: 1.8.0_102
Specification-Version: 3.5

jar は zip ファイルなので、\*.zip と同じ変更も加えています。 さらに、

- 各ファイルの並びを名前順に直しています。ただ META-INF/ と META-INF/MANIFEST.MF は先頭にしている ようです。
- META-INF/MANIFEST.MF から
  - Bnd-LastModified: で始まる生成時刻が書かれた行を削除しています。
  - Built-By: で始まるビルドしたユーザ名が書かれた行を削除しています。

コンパイラやビルドツールのバージョン番号は残しているようです。コンパイラのバグ修正等を想定しているのかもしれません。

- \*.properties も変更しています(後述)。
- javadoc な \*.html があったら変更するらしいです(後述)。
- \*.jar があったら再帰的に変更するらしいです。
- META-INF/\*.SF を含んだ jar ファイルは処理の対象外にしています (Bug#807669)。署名付き jar ファイ ルに含まれるようですが、普通はビルド時に自動署名しないだろうという想定なのでしょう。

#### 10.2.4 javadoc

\*.html のうち <!-- Generated by javadoc があるもの

```
$ grep \<html ./usr/share/doc/libcommons-lang3-java/api/org/apache/commons/lang3/StringUtils.html
<html>
$ grep '<!-- Generated by javadoc' ./usr/share/doc/libcommons-lang3-java/api/org/apache/commons/lang3/StringUtils.html
<!-- Generated by javadoc -->
```

ファイル内の

- 生成時の環境の言語に基づいて書かれた html 要素の lang 属性 (<html lang=) を削除しています。
- <!-- Generated by javadoc 行から他の文字列(生成時刻や生成ツールのバージョン番号)をすべて削除しています。ドキュメントの生成ツールは多少バージョンが変わっても生成結果は変わらない想定なのでしょう。</p>

#### 10.2.5 javaproperties

\*.properties のうち Java 系のビルドツールで自動生成されたように見える、#Generated by Apache Maven などを含むもの

\$ unzip -p ./usr/share/java/commons-lang3-3.5.jar META-INF/maven/org.apache.commons/commons-lang3/pom.properties
#Generated by Apache Maven
version=3.5
groupId=org.apache.commons
artifactId=commons-lang3

# で始まる自動生成時刻が書かれた行を削除しています。

#### 10.2.6 png

\*.png のうち PNG image data のもの

\$ file /usr/share/doc/debian-handbook/html/ja-JP/images/developers-map.png
/usr/share/doc/debian-handbook/html/ja-JP/images/developers-map.png: PNG image data, 750 x 450, 8-bit/color RGB, non-interlaced
\$ file /usr/share/emacs/24.5/etc/images/splash.png
/usr/share/emacs/24.5/etc/images/splash.png: PNG image data, 275 x 188, 8-bit/color RGBA, non-interlaced

\$ dpkg -S /usr/share/doc/debian-handbook/html/ja-JP/images/developers-map.png debian-handbook: /usr/share/doc/debian-handbook/html/ja-JP/images/developers-map.png \$ dpkg-parsechangelog -1 /usr/share/doc/debian-handbook/changelog.gz | grep ^Date: Date: Thu, 22 Sep 2016 16:09:44 +0200

\$ dpkg -S /usr/share/emacs/24.5/etc/images/splash.png emacs24-common: /usr/share/emacs/24.5/etc/images/splash.png \$ dpkg-parsechangelog -1 /usr/share/doc/emacs24-common/changelog.Debian.gz | grep ^Date: Date: Mon, 05 Sep 2016 15:05:00 -0500

中を見てみます。

0000070 00 00 74 49 4d 45 07 e0 09 16 0e 09 2c 4b 76  tIME,Kv
# 長さ
#
# 2016 09 22 14 09 44
<pre>\$ strings -a /usr/share/doc/debian-handbook/html/ja-JP/images/developers-map.png   grep -A1 '[tiz]EXtdate:'</pre>
%tEXtdate:create
2016-09-22T14:09:44-00:00
%tEXtdate:modify
2016-09-22T14:09:44-00:00
\$ exiftool /usr/share/doc/debian-handbook/html/ja-JP/images/developers-map.png   tail -n 5
Modify Date : 2016:09:22 14:09:44
Datecreate : 2016-09-22T14:09:44-00:00
Datemodify : 2016-09-22T14:09:44-00:00
Image Size : 750x450
Megapixels : 0.338

変更時刻 (tIME)、date: の値を指定した時刻に書き換えています。

\$ strings -a /usr/share/emacs/24.5/etc/images/splash.png | grep -A1 '[tiz]EXtCreation Time'
'tEXtCreation Time
2016-09-05T20:05:00-00:00
\$ exiftool /usr/share/emacs/24.5/etc/images/splash.png | tail -n 4
Description : GNU Emacs splash image
Creation Time : 2016-09-05T20:05:00-00:00
Image Size : 275x188
Megapixels : 0.052

作成時刻 (Creation Time) を指定した時刻に書き換えています。

自動生成しないアートワーク等もあると思いますが、どちらか区別が付かないから全部書き換えているのでしょうか。

#### 10.2.7 gettext

\*.mo, \*.gmo のうち GNU message catalog のもの

\$ file /usr/share/locale/ja/LC\_MESSAGES/grub.mo
/usr/share/locale/ja/LC\_MESSAGES/grub.mo: GNU message catalog (little endian), revision 0.0, 233 messages
\$ file /usr/share/locale/ja/LC\_MESSAGES/apt.mo
/usr/share/locale/ja/LC\_MESSAGES/apt.mo: GNU message catalog (little endian), revision 0.0, 367 messages

### 各パッケージの時刻は

\$ dpkg -S /usr/share/locale/ja/LC\_MESSAGES/grub.mo grub-common: /usr/share/locale/ja/LC\_MESSAGES/grub.mo \$ dpkg-parsechangelog -l /usr/share/doc/grub-common/changelog.Debian.gz | grep ^Date: Date: Tue, 01 Nov 2016 11:10:52 +0000

\$ dpkg -S /usr/share/locale/ja/LC\_MESSAGES/apt.mo
apt: /usr/share/locale/ja/LC\_MESSAGES/apt.mo
\$ dpkg-parsechangelog -1 /usr/share/doc/apt/changelog.gz | grep ^Date:
Date: Tue, 04 Oct 2016 19:43:35 +0200

## です。

#### 中を見てみると

\$ grep -a POT-Creation-Date: /usr/share/locale/ja/LC\_MESSAGES/grub.mo
POT-Creation-Date: 2016-11-01 11:10+0000

POT-Creation-Date: で始まる行に生成時刻が入っているので、指定した時刻に書き換えています。

## ただ、指定した時刻より新しい場合のみ上書きしています。

\$ grep -a POT-Creation-Date: /usr/share/locale/ja/LC\_MESSAGES/apt.mo
POT-Creation-Date: 2016-08-30 22:20+0200

指定した時刻より古いので書き換えていません。

#### 10.2.8 pearregistry

\*.reg のうち先頭が a: のもの

<pre>\$ file ./</pre>	<pre>6 file ./usr/share/php/.registry/services_weather.reg</pre>																
./usr/sha	re/j	php/	/.re	egis	stry	//se	ervi	ices.	_wea	athe	er.ı	reg	: AS	SCII	[ te	ext,	, with very long lines
\$ hd ./us	r/sl	nare	e/pł	np/	reg	gist	try/	/serv	vice	es_u	veat	the	c.re	∍g l	he	ead	-n 4
00000000	61	3a	32	32	3a	7b	73	3a	37	3a	22	61	74	74	72	69	a:22:{s:7:"attri
0000010	62	73	22	Зb	61	3a	36	3a	7b	73	3a	31	35	3a	22	70	bs";a:6:{s:15:"p
00000020	61	63	6b	61	67	65	72	76	65	72	73	69	6f	6e	22	Зb	ackagerversion";
0000030	73	3a	35	3a	22	31	2e	39	2e	34	22	Зb	73	3a	37	3a	s:5:"1.9.4";s:7:

\$ dpkg-parsechangelog -1 ./usr/share/doc/php-services-weather/changelog.Debian.gz | grep -e ^Date: -e ^Timestamp: Timestamp: 1460011442 Date: Thu, 07 Apr 2016 08:44:02 +0200

\$ hd ./usr/share/php/.registry/services\_weather.reg | tail
00003580 65 22 3b 73 3a 33 3a 22 65 72 75 22 3b 73 3a 34
00003590 3a 22 72 6f 6c 65 22 3b 73 3a 34 3a 22 6c 65 61 -n 8 00003580 00003590 |e";s:3:"eru";s:4|
|:"role";s:4:"lea| 64 22 3b 7d 7d 7d 73 3a 65 72 73 69 6f 6e 22 3b 31 30 3a 22 78 73 64 76 73 3a 33 3a 22 32 2e 30 |d";}}s:10:"xsdv|
|ersion";s:3:"2.0|
|";s:13:"\_lastmod| 000035a0 000035Ъ0 000035c0 22 3b 73 3a 31 33 3a 22 5f 6c 61 73 74 6d 6f 64 00003540 69 66 69 65 64 22 3b 69 3a 31 34 36 30 30 31 31 |ified";i:1460011| 000035e0 34 34 32 3b 7d |442;}| 000035e5

\_lastmodified に生成時刻が書かれているので、指定した時刻に書き換えています。

#### 10.2.9 gzip

\*.gz, \*.dz のうち gzip compressed data のもの

- ファイル名 (FNAME) の削除
- ヘッダ CRC (FHCRC) の削除
- ヘッダ内の時刻 (mtime) が空でなく新しい場合は書き換え

をするらしいんですが、現物を探せませんでした。というのも、Debian パッケージ内によくある\*.gz ファイル(今ま でたくさん見てきた changelog ファイルや man ページなど)は dh\_compress(1) がファイル名や時刻を保存しないオ プションで圧縮しています。しかも dh(1) シーケンスでは dh\_strip\_nondeterminism(1) の後が dh\_compress(1) なので書き換え対象になっていません。

```
$ grep -C2 dh_strip_nondeterminism /usr/bin/dh
    dh_installwm
    dh_installxfonts
    dh_strip_nondeterminism
    dh_compress
    dh_fixperms
```

## 10.3 まとめ

dh\_strip\_nondeterminism(1)は、再現可能なビルドにしていくに当たり、ビルド結果(バイナリなど)を使う際には通常不要な生成時刻などの情報を決まった値に書き換えています。

ビルドツール等を開発している方は、これに頼らず、そもそもビルド時にこういった情報を埋め込まないようにしたほうがよいでしょう。

普通のパッケージメンテナの方は、このコマンドはこのような書き換えをしているので(man ページには good guesses していると書いてありますが)バグっているものがあったら報告しましょう。

reproducible builds に興味がある方は、もっと書き換えたほうがいいかもしれないものがあったらバグ報告したほうがよいかもしれません。

#### 10.4 参考文献

dh-strip-nondeterminism 0.028-1, p7zip-full, unzip, zip, libarchive-zip-perl, libimage-exiftool-perl 各パッケージ

# 11 DEP5/Machine-readable debian/copyright 再考

岩松 信洋

## 11.1 はじめに

Debian ソースパッケージには debian/copyright ファイルがあり、このファイルには対象ソフトウェアのライセン ス、コピーライトが書かれています。2009 年以前は特にフォーマットもなく、ライセンスも包括的な書き方でした。 例えば gcc-defaults ソースパッケージの debian/copyright ファイルは図 2 のようになっています。

gcc-defaults is Copyright (C) 2000, 2001, 2006, 2009 Debian. These scripts are free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version. On Debian GNU/Linux systems, the complete text of the GNU General Public License can be found in '/usr/share/common-licenses/GPL'. The c89 and c99 man pages are taken from netbsd: Copyright (c) 1999 The NetBSD Foundation, Inc. All rights reserved. (省略)

#### 図2 DEP5 非準拠な debian/copyright

2010 年頃 Debian 開発者である Steve Langasek らによって debian/copyright ファイルを機械処理できるフォー マットに切り替え、自動チェックなどができるようにするため、DEP5 / Machine-readable debian/copyright (以 下、DEP5)が策定されました (http://dep.debian.net/deps/dep5/)。策定後 BTS 609160 によって Debian Policy に取り込まれ、Debian Policy の一部 (Debian Policy 12.5、オプショナル扱い) となっています。最新バー ジョンは 1.0 であり、最新版は https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/ から参照できます。

策定から6年近く経ち、多くのパッケージがDEP5準拠の debian/copyright になっています。しかしこのファ イル、Debian Policy でもオプショナル扱いということもあり、一度作ってしまうとあまり更新しないということも あり、内容が変更されずそのまま続けるという問題もあります。今回は DEP5 についてのフォーマットの紹介と、 debian/copyright ファイルの更新方法について紹介します。

## 11.2 DEP5 フォーマットについて

DEP5のフォーマットはヘッダー部とファイル部で分けられます。ヘッダー部にはソフトウェア全体に関わる情報、 例えば頒布元や連絡先、ファイル部にはファイル毎のコピーライトとライセンスを記述します。 ヘッダー部で利用できるフィールドは以下の通りです。

• Format:

フォーマット内容が書かれたファイルの URI を指定します。実際には https://www.debian.org/doc/ packaging-manuals/copyright-format/1.0/ を指定します。昔は packaging-manuals に含まれていな かったため、議論の場であった wiki の URI (http://wiki.debian.org/Proposals/CopyrightFormat) や http:// が指定されているパッケージもあります。

- Upstream-Name: アップストリームのソフトウェアパッケージ名を指定します。Debianの場合は実際のソフトウェア名とDebian ソースパッケージ名が異なる場合があります。このフィールドはオプション扱いです。
- Upstream-Contact:
   アップストリームの連絡先を指定します。このフィールドはオプション扱いです。
- Source:

ソース頒布先を指定します。このフィールドはオプション扱いです。

• Disclaimer:

ソフトウェアの免責事項を記載します。contrib や non-free のパッケージの場合に利用します。このフィールドはオプション扱いです。

• Comment:

コメントを記載します。ソフトウェアのライセンスが複雑な経緯を持っている場合などに利用されるようです。 このフィールドはオプション扱いです。

• License:

ライセンスを記載します。最初の行ではライセンスのショートバージョンを指定し、次の行からはロー カルファイルシステムにあるライセンスファイルへのパス(例:/usr/share/common-licenses/GPL-2)と、 ソフトウェア保証の放棄や問題があった場合の通知方法などを含めた文章を記載します。もしライセン スファイルがローカルファイルシステムにない場合は全文記載する必要があります。ショートバージョ ンのライセンスの記載方法ですが、GNU GPL version2 or later の場合は GPL-2+、Creative Commons Attribution Share Alike license 3.0 の場合は CC-BY-SA-3.0 と指定することができます。詳細は https: //www.debian.org/doc/packaging-manuals/copyright-format/1.0/#license-short-name を参照し てください。このフィールドはオプション扱いです。

 Copyright: コピーライトホルダーを記載します。このフィールドはオプション扱いです。

上記のフォーマットだけでは、ファイル毎にライセンスが異なる場合、記載することが難しくなります。なので、 本フォーマットでは、上記に加え、ファイル毎のライセンスとコピーライトホルダーを記載するファイル部のフォー マットがあります。ファイル部で利用できるフィールドは以下の通りです。

• Files:

ファイルを記載します。同じライセンス、コピーライトホルダーをもつファイルをまとめて記載することができます。

- Copyright:
   コピーライトホルダーを記載します。
- License: ライセンスを記載します。記載方法は上記の方法と同じです。ライセンスは同じだが、ファイル毎にコピーラ

イトホルダーが異なる場合、ラインセンスのショートバージョンのみを記載し、本文をまとめて記載すること もできます。例を図3に示します。

• Comment:

Files: \* Copyright: foo bar <foo@example.org> License: GPL-2+ Files: debian/\* Copyright: Nobuhiro Iwamatsu <iwamatsu@debian.org> License: GPL-2+ This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this package; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA On Debian systems, the full text of the GNU General Public License version 2 can be found in the file '/usr/share/common-licenses/GPL-2'.

#### 図3 ライセンスの繰り返し例

コメントを記載します。このフィールドはオプション扱いです。

上記を組み合わせることによって debian/copyright ファイルを DEP5 準拠にします。

その他、擬似フィールドとして ライセンスの許諾情報を記載する License-Grant フィールド、ラインセスファイル へのパスを記載する License-Reference フィールドを使っている場合もあります(Bug#786450<sup>\*32</sup>)。

## 11.3 DEP5 の問題点と対策

上記で説明したフォーマットを定義した DEP5 ですが、ファイルが多くなるほど記載することが難しくなり、あま り更新されないという問題があります。また DEP5 はポリシーでもオプショナルなので、移行があまり進んでいない という問題もあります。

ここでは debian/copyright の DEP5 化と更新方法について紹介します。

#### 11.3.1 licensecheck を使った DEP5 フォーマット化

指定したディレクトリにあるファイルのライセンスとコピーライトホルダーを出力する licensecheck というツール があります (図 4)。これは昔は devscripts で提供されていましたが、分離され (Bug#828872\*<sup>33</sup>)、licensecheck パッ ケージで提供されるようになりました。-r オプションで指定したディレクトリを再帰的に検索、--copyright オプ ションでコピーライトホルダーを出力するようにします。実行例を図 4 に示します。

```
$ licensecheck -r --copyright .
ell/io.h: LGPL (v2.1 or later)
[Copyright: 2011-2014 Intel Corporation. All rights reserved]
ell/dbus.c: LGPL (v2.1 or later)
[Copyright: 2011-2014 Intel Corporation. All rights reserved]
(省略)
```

#### 図 4 licensecheck コマンド実行例

これだけでは DEP5 フォーマットにならないため、cdbs で提供されている/usr/lib/cdbs/licensecheck2dep5 を

<sup>\*32</sup> http://bugs.debian.org/786450

 $<sup>^{*33}</sup>$  http://bugs.debian.org/828872

#### 使って整形します(図5)。

<pre>\$ licensecheck -rcopyright .   /usr/lib/cdbs/licensecheck2dep5</pre>
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/ Upstream-Name: FIXME Upstream-Contact: FIXME Source: FIXME Disclaimer: Autogenerated by CDBS
Files: ./ell/base64.c ./ell/base64.h ./ell/checksum.c (中略) ./unit/test-uuid.c
Copyright: 2011-2014, Intel Corporation. 2011-2015, Intel Corporation. 2011-2016, Intel Corporation. 2015, Intel Corporation. 2016, Intel Corporation.
License: LGPL (v2.1 or later) FIXME (省略)



DEP5 フォーマットにして出力してくれますが、License フィールドが FIXME になっていたり、ASCII 以外の文字 は文字化けするなど、完璧な出力はしてくれないため、生成されたテキストを修正する必要があります。

### 11.3.2 cme を使った DEP5 フォーマット化と debian/copyright の更新

licensecheck2dep5 より少し賢い出力をしてくれるツールとして cme があります。これは汎用的な設定ファイル編 集ツールなのですが、libconfig-model-dpkg-perl パッケージ をインストールすることにより、debian パッケージ用 のモデルが使えるようになります。debian/copyright を更新したい場合には dpkg-copyright オプションを使いま す。実行すると DEP5 フォーマットで debian/copyright に出力します (図 6 )。\*<sup>34</sup>

```
$ sudo apt-get install cme libconfig-model-dpkg-perl
$ cme update dpkg-copyright
cme: using Dpkg::Copyright model
updating data
(省略)
```

## 図 6 CME による debian/copyright 更新方法

また libconfig-model-tkui-perl パッケージをインストールすると GUI で編集できるようなります (図 7、図 8)。

<pre>\$ sudo apt-get install libconfig-model-tkui-perl debian/copyright を編集したい場合 \$ cme edit dpkg-copyright debian/copyright を更新した後、編集したい場合 \$ cme update dpkg-copyrightedit エディタで直接編集でも大丈夫です。 \$ cme update dpkg-copyright \$ vi debian/copyright</pre>	

location : Files				show only custom
ekment	status	value	standard value	🔍 View: Files
diskpropyright     Parmat     Variavern-Mail     Variavern-Mail     Variavern-Mail     Variavern-Mail     Variavern-Con     Scarco     Scarco     Scarco     Otacialmer     Comment     Copyright     Pine-Excluder     Vise-Excluder     Vise-Ex	ne Aact	https://www.dob.	. Mtga:/www.deb	Part sub ensures
				Description
				Parterns indicating files having the same license and sharing copyright holders. See <u>files pattern documentation</u> for detail
				-
				info Edit

jusz/bin/cme Dekou/Ceovright

図 7 debian/copyright 編集方法

#### 図 8 cme GUI 起動画面

これも licensecheck2dep5 と同様、完璧な出力はしてくれないため、出力された debian/copyright ファイルを確

 $<sup>^{*34}</sup>$  cme は "cme check dpkg-control" といった debian/control に対するチェックなども行えます。この話はまた今度。

認して修正する必要があります。また UTF-8 に対応しているため、ASCII 文字以外でも正しく処理してくれます。

11.3.3 debmake を使った debian/copyright の DEP5 フォーマット化

debmake コマンドの -cc オプションを使って DEP5 フォーマット化された debian/copyright を作成することも できます。

\$ debmake -cc > debian/copyright

cme との違いは ファイルを列挙する点(cme はワイルドカード(\*)でまとめる)と 不要なファイル(例 debian/copyright)の内容まで確認してしまうなどがあります。debmake は ソースパッケージ作成サポートツールな ので、更新機能がまだないのだと思います。個人的には今のところ debian ディレクトリ以下のファイルチェックにも 使える cme をお薦めします。

#### 11.3.4 license-reconcile による debian/copyright チェックサポート

license-reconcile を使うことによって、cme などで捕捉できないファイルのライセンスやコピーライトが debian/copyright が書かれているか、チェックできるようにするツールとして license-reconcile があります。これ は license-reconcile パッケージによって提供されています。例えば、今まで全てのファイルが GPL-3+ でライセ ンスされていたプログラム(図9)に GPL-2+ でライセンスされている hoge.png ファイルが取り込まれたとします。 バイナリファイルなので cme などでは検知できません。

Files: \*
Copyright: 2016 foo bar <foo@example.org>
License: GPL-3+

図 9 debian/copyright 例

debian/copyright に記載されているかチェックできるようにするため、debian/license-reconcile.yml ファイ ルを用意し、図 11 のような内容を記述します。

Rules: rules: -Glob: hoge.png License: GPL-2+ Copyright: 2016 foo bar <foo@example.org>

図 10 debian/license-reconcile.yml 例

license-reconcile コマンドを実行すると以下のようなコピーライトミスマッチエラーが出力されます(図11)。

\$ license-reconcile License mismatch: File hoge.png has license GPL-2+ which does not match GPL-3+.\ at /usr/share/perl5/Debian/LicenseReconcile/App.pm line 222, <GENO> line 3.

図 11 license-reconcile 実行例

debian/copyright に hoge.png に関するフィールド (図 12)を追加し、再度 license-reconcile コマンドを実 行するとラインセンスフィールドチェックエラーが出力されなくなります。

```
Files: hoge.png
Copyright: foo bar <foo@example.org>
License: GPL-2+
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 2 of the License, or
(at your option) any later version.
(省略)
```

図 12 debian/copyright 追記例

#### 11.3.5 debian パッケージ側の対応

上記では licensecheck や cme といった ツールを使うことによって debian/copyright ファイルを DEP5 フォーマットに切り替えることができることを説明しましたが、debian/rules にチェック用のターゲットを 追加することのよって、パッケージのメンテナンス性が向上します。図 13 のように debian/rules へ追記し debian/rules update-debian-copyright を実行することにより、DEP 5 フォーマット の debian/copyright を debian/copyright.auto に出力します。

図 13 debian/rules 追記例

## 11.4 まとめ

DEP5 フォーマットの簡単な内容紹介と、debian/copyright の DEP5 フォーマットするためのツールの使い方、 debian パッケージでの利用方法について説明しました。以下、まとめです。

- DEP 5 は Debian ポリシーの一部。しかしオプショナル扱い。フォーマット詳細は https://www.debian. org/doc/packaging-manuals/copyright-format/1.0/ または /usr/share/doc/debian-policy/ copyright-format-1.0.txt.gz にある。
- licensecheck ツールによって ソースからのライセンスとコピーライトホルダーを抽出可能。そのままでは DEP5 フォーマットにならないため、licensecheck2dep5 を使う。
- cme と libconfig-model-dpkg-perl を使うことによって licensecheck + licensecheck2dep5 同様のことが可能。岩松のお勧めはこちら。
- license-reconcile を使うことによって cme などで補完できないファイルのチェックができる。
- 毎回 cme や licencecheck などのコマンドを実行するのではなく、debian/rules に書いておくとメンテナンス が楽になる。

# 12 Debian Trivia Quiz

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 された内容と Debian Project News からです。

問題 1. stable 版の jessie でも提供が開始された iceweasel の 問題 6. unstalbe にて perl が Transition されました。さて、 アップデート。パッケージ名が変わっています。何になったで バージョンは何に Transition されたでしょうか しょうか A 5.22 A iceweasel-esr B 5.24 B firefox C 6.0C firefox-esr 問題 7. Linux Kernel において次の 問題 2. Debian Administrator 's Hanbook(Debian 管理者 LTS(=LongTermSupport) とするバージョンの候補が ハンドブック)というドキュメントがあります。ついに日本語 上がってきました。さてバージョンは何でしょうか 版がリリースされました。翻訳作業に多大な貢献をしていただ A 4.8 いた方は誰でしょうか。 B 4.9 A Ryuunosuke Ayanokouzi C 4.10 B Norimitsu Sugimoto C Hideki Yamane 問題 8. debain パッケージのソースコードをダウンロードする 方法はあるか質問がありました。DD の方たちが案内したサー 問題 3. つい先日終了した debconf16。開催場所はどこだった ビスはどれでしょうか。 でしょうか。 A srcs.debian.org A Montreal, Canada B srcs.debian.net B Cape Town, South Africa C sources.debian.net C Kyoto, Japan 問題 9. HP エンタープライズ社から Debian Project ヘサー 問題 4. Debconf16 では次のリリース「Stretch」のスケジュー バを寄付していだきました。このサーバを使って debian.org ルを告知しました。一番最初に行われる Transition freeze の のサービスの載せ替えが実施されます。さて、どの組み合わせ 予定はいつでしょうか が正しいでしょうか A 2016-11-05 A ftp-master.debian.org & www.debian.org B 2016-12-05 B ftp-master.debian.org  $\succeq$  security.debian.org C 2017-02-05 C www.debian.org & security.debian.org 問題 5. unstable にて GCC が Transition されました。さて、 バージョンは何に Transition されたでしょうか A 5.4 B 6.1

C 7.0

#### 本資料のライセンスについて

本資料はフリー・ソフトウェアです。あなたは、Free Software Foundation が公表した GNU GENERAL PUBLIC LICENSE の "バージョン 2" もしくはそれ以降が定める条項に従って本プログラムを再頒布または変更することができます。

本プログラムは有用とは思いますが、頒布にあたっては、市場性及び特定目的適合性についての暗黙の保証を含めて、いかなる保証も行ないません。詳細については GNU GENERAL PUBLIC LICENSE をお読みください。

#### ソースコードについて

本資料のソースコードは Git を使って git://anonscm.debian.org/tokyodebian/monthly-report.git から ダウンロードできます。以下に方法を示します。

\$ git clone git://anonscm.debian.org/tokyodebian/monthly-report.git

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software-to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

 a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering

access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRICHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NORLIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAM, SUCH DATA BEING OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## Debian オープンユーズロゴ ライセンス

Copyright (c) 1999 Software in the Public Interest Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN ON EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
## 13 Debian Trivia Quiz 問題回答

Debian Trivia Quiz の問題回答です。あなたは何問わかりましたか?

- 1. C リーガル問題がひと段落し、debian でも firefox として提供できることになりました。従来の jessie では iceweasel ver38esr が提供されていました。今回は firefox-esr パッケージとして ver45esr の提供が開始されま した。
- A Debian 管理者ハンドブックは無料の web 版及び電子書籍版、有料の書籍版があります。今後の翻訳作業 継続のために寄付を募っています。なお、リリースアナウンスのページは次の URL は次の通りです。https:// debian-handbook.info/2016/get-the-japanese-version-of-the-debian-administrators-handbook/
- B https://debconf16.debconf.org/の案内にあるとおり、南アフリカのケープタウンで行われました。日本から参加した人がいますので、後でどんな感じだったか聞いてみましょう。例年通りであれば、セッションのビデオが後日公開されると思います。
- 4. A Transition freeze が 2016-11-05、Mandatory 10-day migrations が 2016-12-05、Soft freeze が 2017-01-05、 Full freeze が 2017-02-05 の予定となっています。新しいパッケージを Stretch に含めたい場合はお早めに。
- 5. B debian-devel-announce にて GCC-6 へ Transition されました。GCC-6 では C++14 が実装されていま す。GNU Web サイトの「Porting to GCC 6」の記事を確認して新 GCC へ移行してください。メール: 「Transition news: GCC 6 enabled by default」https://lists.debian.org/debian-devel-announce/ 2016/08/msg00001.html
- 6. B 2016年5月にリリースされた perl-5.24 へ Transition されました。2015年クリスマスには Perl-6 リリースの大ニュースがありましたが、Debian で採用されるには先が長そうです。メール:「transition: perl」https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=830200
- B stretch のリリース計画では linux-4.10 を採用する予定としたためフリーズ時期を遅らせることにしていました。LTS なカーネルのリリースが早いからといってすでにアナウンス済のフリーズ時期が早まることはないと思いますが、注視しておいたほうがいいかもしれません。メール:「Linux 4.9 will be next LTS」 https://lists.debian.org/debian-release/2016/08/msg00147.html
- 8. C 韓国で情報工学を学んでいる大学院生の方からセキュリティの研究の一環で全ソースコードがほしいとの

ことでした。そのほかに debmirror コマンドを使うとよいというアドバイスもありました。情報元:https: //lists.debian.org/debian-devel/2016/09/msg00118.html

9. B 寄付いただいたサーバはカナダ、アメリカ、オーストラリアにあるサーバの更新に利用する予定です。具体 的には Debian の主要サービスである ftp-master.debian.org と security.debian.org に割り当てし、その他の コアサービスの向上にも役立てる予定です。https://www.debian.org/News/2016/20161003

## -『あんどきゅめんてっど でびあん』について —

本書は、東京および関西周辺で毎月行なわれている『東京エリア Debian 勉強会』および『関 西 Debian 勉強会』で使用された資料・小ネタ・必殺技などを一冊にまとめたものです。収録 範囲は 2016/06 ~ 2016/11 まで +2016/04 関西の収録漏れ分。内容は無保証、つっこみなどが あれば勉強会にて。

