Grand Unified Debian

銀河系唯一のDebian専門誌

東京エリア/関西Ｄｅｂｉａｎ勉強会

あんどきゅめんてっど でびあん 2015 年冬号 2015 年 12 月 31 日 初版発行

デビアン勉強会

目次

1	Introduction	2
2	Debian update	3
3	Debian と脆弱性対応のあれこれをまとめてみた	7
4	Debian GNU/kFreeBSD セットアップガイド 2015 年版	11
5	APT1.1 超☆牛さんパワー炸裂!	19
6	Debian で HTTP/2 を試す	24
7	NTT フレッツ網経由で Native IPv6	29
8	wiki:Subkeys	33
9	DebConf15 ビデオ紹介	37
10	Debian と arm64 サポート	41
11	Debian パッケージング道場 git buildpackage の使い方	45
12	Debian Trivia Quiz	52
13	Debian Trivia Quiz 問題回答	56

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。

2 Debian update

野島 貴英

2.1 Jessie の状況

- 2015 年 4 月 18 日 Jessie リリース (Debian 8)
- 2015 年 6 月 8 日 Debian 8.1 リリース
  104 個のパッケージが修正（バグ対応が主)。うち 42 個パッケージがセキュリティバグの修正。

早速 Debian 8.1 へアップデートしましょう！

2.2 ちょっとだけ Jessie おさらい

Jessie 特徴を抜粋

- UEFI ブートをサポート
- リリースアーキテクチャに aarm64,ppcel64 が搭載され、ia64,sparc,s390 は外されました
- kFreeBSD が外れました (;_;)
- インストール途中のメニューで各種デスクトップ環境を選べるようになりました。
  (Gnome/Xfce/KDE/Cinnamon/MATE(マテ)/LXDE が選択可能)
- systemd がデフォルトの init システムになりました。
- Linux は 3.16.x 版が採用されています
- utf8 対応。テキストファイルは全部 utf8 に。
- main バイナリパッケージは実に 42,274 パッケージもあります。

2.3 Jessie スクリーンショット

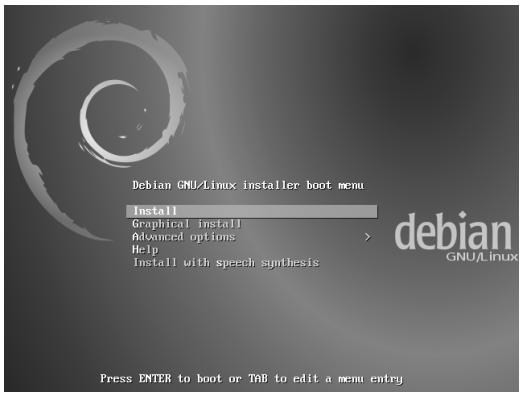


図 1　インストーラ起動直後

図 2　グラフィカルインストール

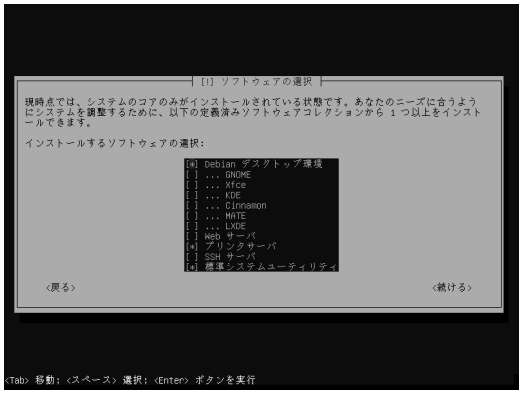


図 3　テキストインストール

図 4　GNOME 環境

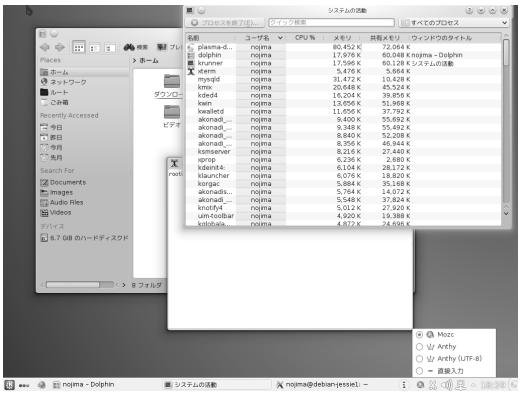


図 5　KDE 環境

図 6　cinnamon 環境

2.4 リリースその他

2015 年 4 月 30 日 Debian GNU/Hurd 2015 リリース

- GNU Hurd 0.6 ベース
- GNU Mach 1.5 搭載
- とりあえずの動作なら、仮想環境で試せます。是非お試しあれ。

2.5 Debian GNU/Hurd 2015 Live

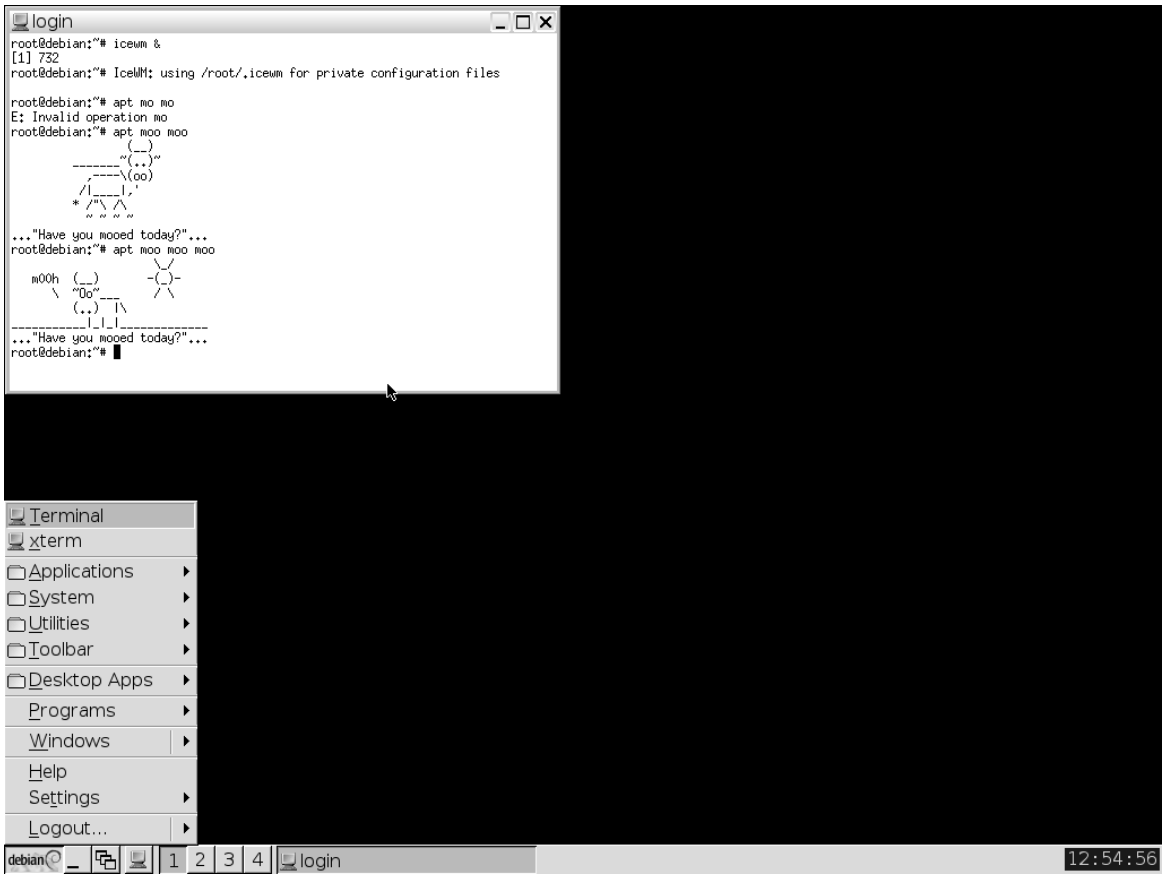
Live イメージもあるよ！

`https://www.gnu.org/software/hurd/hurd/running/debian.html` から抜粋

```
# wget http://people.debian.org/~sthibault/hurd-i386/debian-hurd.img.tar.gz
# tar xz < debian-hurd.img.tar.gz
# kvm -m 512 -drive cache=writeback,file=debian-hurd-20150320.img
```

2.6 Debian GNU/Hurd 2015 Live

Debian GNU/Hurd 2015 Live の様子。



2.7 次期バージョンのコード名

- Debian 9 Stretch ←次回リリース
- Debian 10 Buster ←次々回リリース

2.8 Stretch スケジュール（予定）

以下は Strech の開発に関する予定：

- 2016 年 9 月 5 日 Transition Freeze
- 2016 年 11 月 5 日 Soft Freeze
- 2016 年 12 月 5 日 Freeze

参照　DebConf15 のリリースチームの動画：`http://caesar.acc.umu.se/pub/debian-meetings/2015/debconf15/Onwards_to_Stretch_and_other_items_from_the_Release_Team.webm`

なお、Freeze はリリースの事ではないので注意。Strech に搭載する新規のパッケージの受付を完全にやめ、RC バグ潰しに専念しますという意味。

2.9 gcc-5/libc++6 移行作業中

現在、開発版 (sid) では、Stretch にて新しいバージョンのコンパイラでパッケージを構築できるようにするため、一旦 gcc-5 系列及び、libc++6 の元でバイナリビルドできるようにする大作業が行われています。

- ABI レベルで変更
- C++11 対応
- GFortran でコンパイルするパッケージもこちの影響を受ける

2.10 httpredir.debian.org 稼働開始

ネットワーク観点から、もっとも近い debian の mirror サイトを紹介するサービスであった、

- cdn.debian.net ... DNS クエリで、もっとも近い mirror サイトを返す
- http.debian.net ... http redirect を使って、もっとも近い mirror サイトを返す

が終了し、代わりのサービスとして、正式に httpredir.debian.org が稼働しました。
が、しかし本来、

```
# cat /etc/apt/souces.list
deb http://httpredir.debian.org/debian/ jessie main
deb-src http://httpredir.debian.org/debian/ jessie main
deb http://httpredir.debian.org/debian/ jessie-updates main
deb-src http://httpredir.debian.org/debian/ jessie-updates main
#
```

で、うまく近い mirror が紹介されるはずが、何故か日本は精度が悪い状況です。インストーラによって設定される ftp.jp.debian.org の方が現在は良いです。

3 Debian と脆弱性対応のあれこれをまとめてみた

野島 貴英

3.1 はじめに

コンピュータセキュリティの勉強会にて登壇を頼まれました。ちょうど良かったので、Debian の脆弱性対応をおさらいして、発表してみることにします。

3.2 Debian のセキュリティチーム

Debian パッケージの脆弱性問題を専門に扱っている心臓部のチームの連絡先は以下の通りです。

- security@debian.org または
- team@security.debian.org

3.3 脆弱性に関する ML

Debian の脆弱性に関するアナウンス・議論が行われている ML は次のとおりです。

- Debian パッケージのセキュリティに関するアナウンス
  debian-security-announce@lists.debian.org
  アーカイブ：
  `https://lists.debian.org/debian-security-announce/`
- Debian パッケージの脆弱性に関するオープンな議論
  debian-security@lists.debian.org
  アーカイブ：`https://lists.debian.org/debian-security/`

3.4 脆弱性対応が行われないアーカイブエリア

Debian パッケージ群は、main/contrib/non-free のアーカイブエリアでそれぞれ提供されています。実は、脆弱性対応について、contrib/non-free のアーカイブエリアのパッケージについては、基本的に除外されています。

こちらの理由としては、contrib と non-free に収められたソフトウェアのライセンスの問題によります。これらのソフトウェアはライセンスの都合上、勝手に直すことが許諾されていない場合があるためとなります。もちろん、contrib と non-free であっても、ライセンス上、直しても良いものもあり、こちらについては通常の通りセキュリティチームにて脆弱性対応が図られることがあります。

3.5 脆弱性に関しての対応の流れ

Debian ユーザあるいは Debian の関係者にかかわらず、もし脆弱性の問題を見つけたら、

PAT1: 既知の問題 (公開済脆弱性情報）の場合は、security タグをつけて BTS してください。

PAT2: 既知でない場合は、そっと security@debian.org または team@security.debian.org にメールで連絡 (英文) し、あとは指示に従えば良いです。

もちろん、脆弱性対策パッチも書いたのであれば、報告の際にパッチも一緒に送ると喜ばれます。

3.5.1 既知の脆弱性でない場合の動き

既知の脆弱性でない場合、セキュリティチームは Debian 以外のベンダ（ディストリビューション等含む）関係者らにも脆弱性情報が共有され、CVE などの脆弱性情報データベース登録にも協力する動きが取られます。

3.5.2 既知の脆弱性の収集

Mitre 社 (CVE の DB を管理している会社）から、CVE の情報を定期的に取り込んでいます。こちらの情報に基づき、Debian に関係ある・なし、重要度を判定して仕分けされ、バグ追跡システムに登録していく仕組み（インフラ）があります。

3.6 脆弱性対応状況の確認方法

以下のサイトで状況を確認できます。

1. セキュリティ観点から確認したい場合
   - `https://security-tracker.debian.org/tracker/`
   - 様々な視点から、巷の脆弱性データベースと Debian の各パッケージの対応状況を確認可能です。
2. QA 観点から確認したい場合
   - `https://tracker.debian.org/`
   - Debian パッケージについて、現在のバグの修正状況、パッケージのリリース状況がパッケージ毎にわかります。
3. BTS した・されたものを確認したい場合
   - `https://bugs.debian.org/`
   - BTS した結果がどう扱われているか、進行しているか確認できます。

3.7 その他トピック

以上で、まとめとしては終わったので、昨今の Debian のセキュリティに関するトピックをいくつか紹介します。

3.7.1 hardening

C/C++ で書かれたプログラムについて、gcc の機能を活用して、セキュリティ強化を行った Debian パッケージを作る試みです。hardening 有効時、ビルド時に実際に付け加えられる gcc のオプションは以下の通りです。

```
-fstack-protector-strong -Wformat -Werror=format-security -D_FORTIFY_SOURCE=2
```

3.7.2 Debian パッケージ済みの脆弱性静的解析ツール群

Debian でパッケージ済の脆弱性静的解析ツール群があります。

項番	パッケージ名	概要
1	flawfinder	C/C++ にてセキュリティ上問題の起きそうな部分を指摘。
2	rats	C, Perl, PHP, Python のコード上問題の起きそうな部分を指摘。
3	pscan	C/C++ にて format 文の文字列について問題の起きそうな部分を指摘。

表 1　Debian パッケージ済み脆弱性静的解析ツール

3.7.3　lintian

lintian は、Debian パッケージについて、自動でスキャンして問題点を解析して警告してくれるツールであり、ある程度のセキュリティ対策の為の対応がパッケージ開発にあたり必須になっており、こちらが過不足なく行われているかをスキャンして開発者に教えてくれます。

昨今にて搭載されたセキュリテイ対応の例として、梱包されているドキュメントに HTML ソースがあった場合、外部リンクが含まれているかをスキャンする事が追加されました。これは、もし、パッケージに含まれるドキュメントが HTML であった場合に、外部リンクへアクセスするようなタグが混ざっているような事が無いようにします。もし、そういったタグやリンクが残っていると、ドキュメントをブラウザで開いた時に、うっかり悪意のあるサイトへ自動的に誘導されてしまうのを防ぎます。

3.7.4　systemd

systemd にはセキュリティに関して強化を図ることが出来るオプションがいくつもあります。こちらをどう活かして、systemd の*.service ファイルを作るかというお話です。

良い文章として、「Security Features in systemd」`http://0pointer.net/public/systemd-nluug-2014.pdf` がありますので、見てみるとよいでしょう。

3.7.5　Reproducible Builds

Debian の 22,000 を超えるソースパッケージを一旦再ビルドし、すでに配布されている Debian パッケージのバイナリと照合する試みの事です。こちらにより、Debian パッケージにいつの間にか悪意のあるバイナリが含まれているような事が無いかをチェックするのが狙いです。Debian の他にも、Fedora/OpenSUSE/NixOS でも行われているとのことです。

詳しくは、

Debian　`http://reproducible.alioth.debian.org/presentations/2014-02-01-FOSDEM14.pdf`
Fedora　`http://securityblog.redhat.com/2013/09/18/reproducible-builds-for-fedora/`

を参照ください。

3.7.6　LTS(Long Time Support)

Debian の各バージョンについて、セキュリティアップデートについてのサポート切れまでの期間を 3 年→ 5 年へ延長する試みです。但し、数万もあるソフトウェア全部について 5 年もサポートするのは非現実的なので、限られたパッケージのセキュリティアップデートのみ延長します。

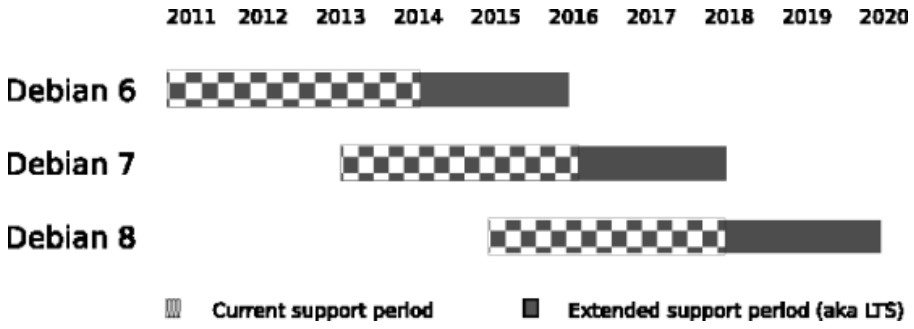


図 7　LTS のサポート期間

LTS は無料でも利用できる一方、有償サポート契約も用意されています。Freexian 社と有償サポート契約を行うのですが、払った値段に応じて、サポートしてくれるパッケージの選択の要望が通りやすくなるという特典があるようです。日本の会社にて契約締結実績は GREE 社があります。

有償サポート契約について詳しくは「Debian Long Term Support」`https://www.freexian.com/en/services/debian-lts.html` を参照ください。

3.7.7　debian-security-support パッケージ

debian-security-support パッケージを導入し、check-support-status コマンドを実行すると、現在お使いの Debian 機に導入されいているパッケージの脆弱性のサポートについて、サポート切れ、もしくは、何らかの理由により脆弱性対策にあたり制限を加えざるを得なかったもののリストが取れるようになりました。

このコマンドの動作としては、

- /usr/share/debian-security-support/security-support-ended
- /usr/share/debian-security-support/security-support-limited

に記載されたパッケージのサポート状況に関する制限の情報と、実際に導入されていてるパッケージ名を照合することにより動作します。もし、これらのファイルから、制限があるパッケージがマシンに見つかった場合は、警告を出してくれます。なお、LTS では、どこまでパッケージについて、脆弱性対応のサポートがなされているかの確認ができます。

3.8　おわりに

Debian の脆弱性対応についてまとめてみました。Debian のセキュリティ維持の為、様々な努力が払われていることがわかりました。

4 Debian GNU/kFreeBSD セットアップガイド 2015 年版

杉本 典充

4.1 はじめに

Debian GNU/kFreeBSD は、FreeBSD カーネルで動作する Debian です。Debian は単なる Linux ディストリビューションではなく、ユニバーサルオペレーティングシステムを目指しており、その例として Debian GNU/kFreeBSD があります。[*1]

ただ Debian GNU/kFreeBSD はその特異さゆえに Debian GNU/Linux と異なる点があります。今回は、Debian GNU/kFreeBSD に触れるにあたり、どのようにセットアップを行うとよいか説明します。

4.2 Debian Ports と Debian GNU/kFreeBSD

Debian Ports[*2]とは、さまざまな CPU やカーネルで動作するように移植を行うプロジェクトのことです。

FreeBSD カーネルで起動する Debian を作るプロジェクトがあり、その Debian のことを「Debian GNU/kFreeBSD」と呼んでいます (k は kernel のこと)。現在では Intel CPU のアーキテクチャのみあります (kfreebsd-amd64、kfreebsd-i386)。Debian 6 (Squeeze) ではテクノロジープレビューとして初めて stable リリースされました。Debian 7 (Wheezy) でも継続して stable リリースされたのですが、Debian 8 (Jessie) では Drop となったためリリースされていません。[*3]

4.3 Debian GNU/kFreeBSD のインストール

4.3.1 インストールイメージの入手

`https://www.debian.org/devel/debian-installer/` にある daily-images を使ってインストールします。kfreebsd-amd64 版の mini.iso をダウンロードし、USB メモリに dd してインストールディスクを作成します。kfreebsd-i386 版の mini.iso を利用しても構わないのですが、ファイルシステムに ZFS を使う場合はメモリ不足になりがちなため注意してください。

mini.iso を dd した USB メモリを差して PC を起動すると Debian Installer が起動します。なお、現時点の kfreebsd 版 Debian Installer は以下の制約があります。[*4]

- UEFI ブートに対応していない

[*1] 私が知っている限り「Universal Operating System」という記述は本家 web サイトトップページの title タグでしか見たことないです。
[*2] `https://www.debian.org/ports/`
[*3] `https://lists.debian.org/debian-devel-announce/2014/09/msg00002.html` において、stable を維持しつつ sid の開発を進めるにはそれなりに人手が必要であるが、kFreeBSD を作業する人手は不足していることが指摘されています。
[*4] 本家 FreeBSD-10.1 のインストーラは UEFI ブートに対応し、ディスク形式を GPT としてインストールすることが可能です。

- ディスク形式は MBR のみに対応している (GPT は非対応)

4.3.2 インストーラの表示言語

kfreebsd 版 Debian Installer は、日本語の表示ができません (インストーラでフレームバッファが有効になっていないと思われる)。そのため、LANG=C でインストールを進めます。

4.3.3 パーティション構成とファイルシステム

Debian GNU/kFreeBSD をインストールするときは root パーティションを MBR の基本パーティションにする必要があります（拡張パーティションにインストールすると grub のインストールに失敗します）。swap パーティションは拡張パーティションに作成しても問題ありません。

この前提があるため、プリインストールの Windows とデュアルブートしたい場合は、以下のパーティション構成でほぼ決まりになります。[*5]

```
# fdisk -l /dev/ada0
Disk /dev/ada0: 298.1 GiB, 320072933376 bytes, 625142448 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x33d61950

Device      Boot     Start       End   Sectors   Size Id Type
/dev/ada0p1 *         2048   2459647   2457600   1.2G  7 HPFS/NTFS/exFAT
  -> Windows 7 のインストーラが自動で確保する領域
/dev/ada0p2        2459648 375142399 372682752 177.7G  7 HPFS/NTFS/exFA
  -> Windows 7 の OS をインストールした NTFS パーティション
/dev/ada0p3      580083712 625140399  45056688  21.5G  7 HPFS/NTFS/exFAT
  -> ノート PC メーカーのリカバリー領域
/dev/ada0p4      375142400 580083711 204941312  97.7G a5 FreeBSD
  -> Debian GNU/kFreeBSD をインストールする ZFS ストレージプール領域
  -> ZFS パーティションの中に root ボリュームと swap ボリュームを作成します
```

4 つの基本パーティションにうち、1 つを ZFS ストレージプール (LVM の物理ボリュームに似ている概念) に割り当てます。そして ZFS ストレージプールから root ボリュームと swap ボリュームを作成します (別途 boot ボリュームを作成して/boot パーティションへ割り当てしなくても問題ありません)。

ディスクすべてを kfreebsd に割り当てることができ、かつメモリ搭載量が少ない環境 (古い PC や仮想マシン) へインストールする場合は UFS を選択したほうがよいと思います。[*6]

4.4 Debian GNU/kFreeBSD 固有の Debian パッケージ

Debian GNU/kFreeBSD 固有のパッケージの例を紹介します。

4.4.1 kfreebsd-image パッケージ

Debian GNU/kFreeBSD の kernel イメージを収録したパッケージです。unstable には kfreebsd-image-10.1 があり現状のデフォルト kernel になっています。experimental には kfreebsd-image-10.2[*7]、kfreebsd-image-11[*8] もあります。

[*5] ファイルシステムを UFS にする場合は基本パーティションの上限 (4 つ) を超えることがあるため、ディスク内の Windows 用リカバリ領域を削除する、D ドライブを削除するなどの事前準備が必要になります。

[*6] インストール時に作成する UFS パーティションは soft update が無効になっています (つまり同期書き込みする設定になっている)。rescue mode から”tunefs.ufs -n enable ”partition-path””を実行すると soft update を有効にして非同期書き込みに変更できますが、耐障害性が低下しますのでご注意ください。

[*7] FreeBSD-10.2 は 2015 年 8 月 14 日 UTC にリリースされていますので次第にこちらがデフォルトになるでしょう。

[*8] 現在の FreeBSD-CURRENT です。FreeBSD-CURRENT は、debian でいうところの unstable と似た位置付けであり、最新の開発版です。

4.4.2 zfsutils パッケージ

zfsutils パッケージは ZFS を操作するコマンドを含んだパッケージです。インストール時のファイルシステムに ZFS を選択した場合はデフォルトでインストールされます。

kfreebsd-image-10.1 で利用できる ZFS のバージョンは ver 28 となっています。

```
$ zpool upgrade -v
(snip)
28  Multiple vdev replacements
```

4.4.3 freebsd-utils パッケージ

FreeBSD 固有のコマンドを含んだパッケージです。/sbin/mount_*、/usr/sbin/jail などが入っています。

4.4.4 freebsd-smbfs パッケージ

freebsd-smbfs パッケージは、Windows ファイル共有 (SMB 共有) へアクセスするためのパッケージです。インストールすると、/usr/sbin/mount_smbfs コマンドが使えるようになります。

Windows ファイル共有先を mount するには以下のコマンドを実行します。

```
# mount_smbfs -E UTF-8:CP932 -I {ファイルサーバの IP アドレス} -U {smb ユーザ名} //{ファイルサーバの IP アドレス}/{dir} {mount 先 dir}
```

4.4.5 freebsd-ppp パッケージ

freebsd-ppp パッケージは、ダイアルアップする/usr/sbin/ppp コマンドを含んでいます。3G や LTE に対応した USB モデムを使う場合に必要となります。

Debian GNU/Linux では ppp パッケージや wvdial パッケージでダイアルアップしますが、kfreebsd では使えないため注意が必要です。

ppp 接続の例については後述します。

4.4.6 pf パッケージ

FreeBSD kernel がもつ Packet Filter と呼ばれるいわゆるファイアウォール機能を制御するコマンド/sbin/pfctl を含んだパッケージです。[*9]

/sbin/pfctl の設定ファイルは/etc/pf.conf であり、Linux の iptables 用設定ファイルと中身が全く異なります。

4.5 Windows と Debian GNU/kFreeBSD のデュアルブート設定

Debian GNU/kFreeBSD の boot loader は grub2 を利用しています。grub2 で Debian GNU/kFreeBSD と Windows をデュアルブートしたい場合は以下の操作を行い、grub にメニューを追加します。((hd0,2) の部分はインストール環境に合わせて変更してください))

```
# cd /etc/grub
# vi 40_custom.kfreebsd

#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
menuentry ``Windows (loader)'' {
  insmod part_msdos
  insmod ntfs
  set root=(hd0,2)
  chainloader +1
}

# update-grub
```

[*9] Linux kernel の netfilter 機能と制御コマンド iptables に相当します。

4.6 ハードウェアとソフトウェアのセットアップ

4.6.1 有線 LAN

有線 LAN は利用するドライバによってデバイス名が変化します (intel の PC 向けなら em0、realtek なら re0)。設定ファイルは Debian GNU/Linux と同じ/etc/network/interfaces ですが、allow-hotplug 句は linux で使われる udev が提供している機能であることから kfreebsd では利用できないため注意が必要です。

そのため、有線 LAN 接続環境がない状況で OS を起動すると有線 LAN による DHCP の IP アドレス取得がタイムアウトするまで login プロンプトが出てこなくなります (起動に時間がかかる)。私はノート PC に Debian GNU/kFreeBSD をインストールする場合は以下コマンドを手動で実行してネットワークへ接続するようにしています。

```
# vi /etc/network/interfaces

#auto em0  <-コメントにします
iface lan_home inet dhcp


# ifup em0=lan_home
```

4.6.2 無線 LAN

無線 LAN は Thinkpad X220 に搭載している「Intel Centrino advanced-N 6205」で動作することを確認しています。そのため、同じ iwn デバイスと認識される「Intel Wireless WiFi Link 4965」以降の Intel 製無線 LAN カードであれば動作すると思います。

無線 LAN を利用するために kfreebsd-image-10 系の最新版 kernel、firmware、無線 LAN デーモン”wpasupplicant”をインストールします。[*10]

```
# vi /etc/apt/sources.list

deb http://ftp.jp.debian.org/debian/ unstable main contrib non-free

# apt-get install kfreebsd-image-10-amd64
# apt-get install firmware-iwlwifi wpasupplicant
# reboot
```

FreeBSD の無線 LAN インタフェースは、物理インタフェースと論理インタフェースに分かれています。そのため、論理インタフェースを生成する必要があります。

```
# ifconfig iwn0 （これが物理インタフェース名）
iwn0: flags=8802 metric 0 mtu 2290
ether xx:xx:xx:xx:xx:xx
media: IEEE 802.11 Wireless Ethernet autoselect (autoselect)
status: no carrier
nd6 options=23

# ifconfig wlan create wlandev iwn0
wlan0

# ifconfig wlan0（これが論理インタフェース名）
wlan0: flags=8802 metric 0 mtu 1500
ether xx:xx:xx:xx:xx:xx
inet6 fe80::xxxx:xxxx:xxxx:xxxx%wlan0 prefixlen 64 scopeid 0x6
ssid " channel 5 (2432 MHz 11g)
country US authmode WPA2/802.11i privacy OFF txpower 15 bmiss 10
scanvalid 60 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7
roam:rate 5 protmode CTS wme
media: IEEE 802.11 Wireless Ethernet autoselect (autoselect)
status: no carrier
    nd6 options=23
```

接続する無線 LAN アクセスポイントの認証情報設定ファイルを作成します。

[*10] kfreebsd で無線 LAN が使えるようになったのは 2014 年 8 月頃と思われます。 https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=642468

```
$ wpa_passphrase apname1 appassword > wpa_apname1.conf
$ cat wpa_apname1.conf
network={
  ssid=''apname1''
  #psk=''appassword''
  psk=e9fdcb43eba09b6342df30f14275625c8494e534799a82d6639b6124434ea627
}
```

無線 LAN アクセスポイントへ接続し、DHCP で IP アドレスを取得します。IP アドレスは論理インタフェースに付与されます。

```
# wpa_supplicant -i wlan0 -c ./wpa_apname1.conf
Successfully initialized wpa_supplicant
ioctl[SIOCS80211, op=20, val=0, arg_len=7]: Invalid argument
ioctl[SIOCS80211, op=20, val=0, arg_len=7]: Invalid argument
wlan0: Trying to associate with yy:yy:yy:yy:tt:tt (SSID='apname1' freq=2432 MHz)
wlan0: Associated with yy:yy:yy:yy:yy:yy
wlan0: WPA: Key negotiation completed with yy:yy:yy:yy:yy:yy [PTK=CCMP GTK=CCMP]
wlan0: CTRL-EVENT-CONNECTED - Connection to yy:yy:yy:yy:yy:yy completed [id=0 id_str=]
wlan0: WPA: Group rekeying completed with yy:yy:yy:yy:yy:yy [GTK=CCMP]

# dhclient wlan0

# /sbin/ifconfig wlan0
wlan0: flags=8843 metric 0 mtu 1500
ether xx:xx:xx:xx:xx:xx
inet6 fe80::xxxx:xxxx:xxxx:xxxx%wlan0 prefixlen 64 scopeid 0x6
inet 192.168.1.100 netmask 0xffffff00 broadcast 192.168.1.255
ssid apname1 channel 5 (2432 MHz 11g) bssid yy:yy:yy:yy:yy:yy
country US authmode WPA2/802.11i privacy ON deftxkey UNDEF
AES-CCM 2:128-bit txpower 15 bmiss 10 scanvalid 60 bgscan
bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
wme roaming MANUAL
media: IEEE 802.11 Wireless Ethernet OFDM/48Mbps mode 11g
status: associated
        nd6 options=23
```

wpasupplicant コマンドで無線 LAN アクセスポイントへ接続を試みたがエラーが発生し接続できない場合があります。その場合は以下を試すと接続できる場合があります。

- 接続先 SSID を 2.4GHz 帯のものに変更する。
- “`# ifconfig wlan0 -ht40`” を実行する。[*11]

4.6.3 USB モデムによるダイアルアップ

3G や LTE の USB モデムを使ってダイアルアップ接続することができます。NTT ドコモから出ている「L-02C」という LTE に対応した USB モデムを例に接続する手順を説明します。

まずは ppp コマンドと USB モデム処理に使うコマンドをインストールします。

```
# apt-get install freebsd-ppp usb-modeswitch
```

L-02C を PC に差すと CD-ROM デバイスとして認識します。そのため、L-02C をモデムモードに変更するコマンドを実行します。

```
# vi /etc/usb_modeswitch.d/L02C.conf
#######################################################
# LG L-02C LTE modem

DefaultVendor= 0x1004
DefaultProduct=0x61dd

TargetVendor= 0x1004
TargetProduct=0x618f

MessageContent=55534243123456780000000000000061b000000020000000000000000000000
NeedResponse=1
CheckSuccess=20

# usb_modeswitch -c /etc/usb_modeswitch.d/L02C.conf
```

[*11] デュアルチャネル接続を無効にして、20MHz 幅の電波で通信するように指示するコマンドです。

```
# ls /dev/cua*
/dev/cuaU0.0       /dev/cuaU0.1       /dev/cuaU0.2       /dev/cuaU0.3
/dev/cuaU0.0.init  /dev/cuaU0.1.init  /dev/cuaU0.2.init  /dev/cuaU0.3.init
/dev/cuaU0.0.lock  /dev/cuaU0.1.lock  /dev/cuaU0.2.lock  /dev/cuaU0.3.lock
```

次に ppp コマンドの設定ファイルを作成し、ppp 接続します。利用する回線によって適時 APN、ユーザ名、パスワードは変更してください。PPP 接続に成功すると tun インタフェースが生成され、IP アドレスが付与されます。

```
# vi /etc/ppp/ppp.conf

default:
  set log Phase Chat LCP IPCP CCP tun command
  ident user-ppp VERSION
  set device /dev/cuaU0.2
  set speed 38400
  set dial ``ABORT BUSY TIMEOUT 2 \
  \"\" \
  AT OK-AT-OK \
  AT+CFUN=1 OK-AT-OK \
  AT+CMEE=2 OK-AT-OK \
  AT+CSQ OK \
  AT+CGDCONT=1,\\\"IP\\\",\\\"apn.ne.jp\\\" OK \
  AT+CGACT? OK-AT-OK \
  AT+CGATT? OK \
  AT+CGCLASS? OK \
  AT+COPS? OK \
  ATD*99***1# CONNECT''
  set timeout 180
  enable dns

myppp:
  set phone ``*99***1#''
  set authname ``apnuser''
  set authkey ``apnpass''
  set timeout 300
  disable ipv6
  set ifaddr 10.1.0.1/0 10.1.0.2/0 255.255.255.255 0.0.0.0
  add default HISADDR

# ppp -foreground myppp
```

4.6.4 ビデオドライバ

現在、主流のビデオカードは Intel 社の CPU 内蔵グラフィックス、AMD 社の Radeon シリーズ、NVIDIA 社の GeForce シリーズがあります。Debian GNU/kFreeBSD は FreeBSD 向けに提供されるプロプラエタリドライバのビルド環境がないため、オープンソース版ドライバを利用する必要があります。そのため、Intel CPU 内蔵グラフィックスまたは Radeon シリーズのビデオカードを利用することをおすすめします。[*12]

Intel CPU 内蔵グラフィックスのビデオカードを利用する場合は以下のパッケージをインストールします。

```
# apt-get install xserver-xorg-video-intel
```

Radeon シリーズのビデオカードを使用する場合は以下のパッケージをインストールします。

```
# apt-get install xserver-xorg-video-ati
```

次は KMS(kernel mode settings) を有効にします。以下コマンドを実行するとコンソール画面の解像度が上がります[*13]。KMS を有効にした状態にすると、X Window System で液晶モニタの解像度を最大限に利用でき、xrandr コマンドで外部モニタ出力もできるようになります。

```
# kldunload i915
# kldload i915kms
```

再起動後も自動で kernel module をロードするように設定します。

```
# vi /etc/modules
i915kms
```

[*12] NVIDIA 社ビデオカードを利用する場合、xserver-xorg-video-nouveau パッケージは Linux 版のみの提供であるため使えず、オープンソース版ドライバ xserver-xorg-video-nv は開発がすでに止まっておりかなり古いビデオカードしか対応してません。

[*13] 昔の kfreebsd では、i915.ko を load したまま i915kms.ko を load すると reboot してしまう現象が起こっていたため念のため unload しています。現在の kfreebsd-image-10.1 の kernel で試したところ、load したままでも大丈夫ではあるようです。

4.6.5 locale の再設定

Debian Installer では LANG=C を選択してインストールしているため、出力メッセージが英語になっています。そのため locale を日本語に変更します。(ただし、コンソール環境では日本語メッセージが文字化けするので注意)

```
# dpkg-reconfigure locales
-> ja_JP.UTF-8 を選択する
```

4.6.6 X Window System のキーボード設定

FreeBSD の xorg では hal を使ってキーボードのレイアウトを自動判定しています。しかし、hal は upstream によるメンテナンスをすでに終了しており、kfreebsd でも debian パッケージの提供は終了しています。そのため X Window System 起動時のキーボードレイアウトはデフォルトの英語キーボードと判定されます。

キーボードレイアウトの変更は xorg.conf で直接指定する、ウィンドウマネージャのキーボード設定を利用するなどして対応する必要があります。

4.6.7 web ブラウザ

web ブラウザは Debian GNU/Linux 同様に iceweasel パッケージが提供されています。しかし chromium パッケージは kfreebsd に存在しません。

また Adobe Flash Player は Linux 用のバイナリとして提供されるため、Flash を見たい場合は gnash などをインストールする必要があります (ただし動作の安定度は未知数)。

4.6.8 USB 3.0

kfreebsd-image-10-amd64 パッケージの kernel において、xhci.ko が static link されていることを確認しています。ただし、USB 3.0 をもつ PC に Debian GNU/kFreeBSD をインストールしたことがないため動作は未確認です。

4.6.9 サウンドドライバ

FreeBSD は OSS という仕組みでサウンドを出力します (ALSA は Linux 専用のサウンドシステムです)。最近の PC には High Definition Audio 規格のチップが搭載されることが多いため、snd_hda.ko ドライバでサウンドを出力することができます (snd_hda.ko は kernel に static link されています)。

また、pulseaudio パッケージをインストールし、audacious を使って pulseaudio 経由で音楽を再生できることは確認しています。

4.6.10 電源関係

CPU クロックの制御は powerd パッケージの powerd が行っています。現在動作中の CPU クロック数は sysctl コマンドで取得できます。

```
$ sysctl dev.cpu.0.freq
dev.cpu.0.freq: 800
```

バッテリー残量を取得するには acpiconf コマンドを実行します。

```
$ /usr/sbin/acpiconf -i 0
Design capacity:62160 mWh
Last full capacity:26300 mWh
Technology:secondary (rechargeable)
Design voltage:11100 mV
Capacity (warn):1315 mWh
Capacity (low):200 mWh
(snip)
State:discharging
Remaining capacity:95%
Remaining time:1:25
Present rate:17681 mW
Present voltage:12186 mV
```

サスペンドとハイバーネートについては未確認です。

4.6.11 USB メモリの mount

FreeBSD では USB メモリを PC に差すと/dev/da0s1 のように認識します。FAT16 または FAT32 の領域をもつ USB メモリを/mnt/usb へ mount するには以下のコマンドを実行します。

```
# mount_msdosfs -L ja_JP.UTF-8 -D CP932 /dev/da0s1 /mnt/usb
```

exFAT を mount するのは exfat-fuse パッケージを、NTFS を mount するには ntfs-3g パッケージを使うと思われますが、試したところどちらもエラーが出て mount できませんでした。

4.6.12 Linux エミュレーション

FreeBSD kernel には linux バイナリ互換機能があり、Linux の i386 バイナリを実行することができます。[*14]

debootstrap で linux-i386 環境を用意し chroot することによって linux-i386 形式のバイナリを実行することができます。この機能を使うことにより linux バイナリのみ提供されるソフトウェアを kfreebsd 上で利用することができます。[*15]。

4.6.13 仮想化

FreeBSD には FreeBSD Jail というコンテナ型仮想化環境を実行する機能があります。freebsd-utils パッケージをインストールすることで利用できます。[*16]

Debian GNU/kFreeBSD に qemu パッケージはありますので、他の OS を使う必要がある場合は利用してもよいでしょう。(ただしあまり使ったことがないため、動作の安定度は未知数)

その他に FreeBSD には OS を完全仮想化して動作させる virtualbox、bhyve がありますが、Debian GNU/kFreeBSD にはまだ移植されていません。

4.7 おわりに

Debian GNU/kFreeBSD のインストール方法とセットアップ方法について説明しました。動作確認ができていない機能や移植されていない機能もまだ多くありますが、OS を開発したい方にはよい腕試しの場になると思います。興味のある方はまず使ってみるところから始めるのはいかがでしょうか。

4.8 参考文献

- 「Debian_GNU/kFreeBSD - Debian Wiki」 `https://wiki.debian.org/Debian_GNU/kFreeBSD`
- 杉本典充 (2015). 「Debian GNU/kFreeBSD における Jail 構築を試してみた」 `http://tokyodebian.alioth.debian.org/pdf/debianmeetingresume201502-presentation-sugimoto.pdf`

[*14] FreeBSD kernel の kernel module である linux.ko が処理しています。

[*15] 本家 Java、Adobe Flash Player、Adobe Reader などが該当します。

[*16] 本原稿の発表当時 (2015 年 11 月) は jls コマンド、jexec コマンドがまだありませんでしたが、その後 bugfix されて使えるようになっています。 `https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=806562`

5　APT1.1 超☆牛さんパワー炸裂!

野島 貴英

5.1　はじめに

debian-devel-announce に、”Moo! 9th preview of APT 1.1 released: Go and test new supercow powers” というタイトルで、APT 1.1 のアナウンスが流れました。

今回は、Debian の重要なツールの 1 つである APT について、1.1 に搭載された特徴をネタにして、いろいろ調べてみました。

5.2　Debconf15 のセッション

2015/8/15-8/22 で開催された Debconf15 のセッションで、APT1.1 のセッションが開かれました。こちらビデオが公開されていますので、興味ある方は是非ご覧ください。

ビデオ：”This_APT_has_Super_Cow_Powers”

`http://meetings-archive.debian.net/pub/debian-meetings/2015/debconf15/This_APT_has_Super_Cow_Powers.webm`

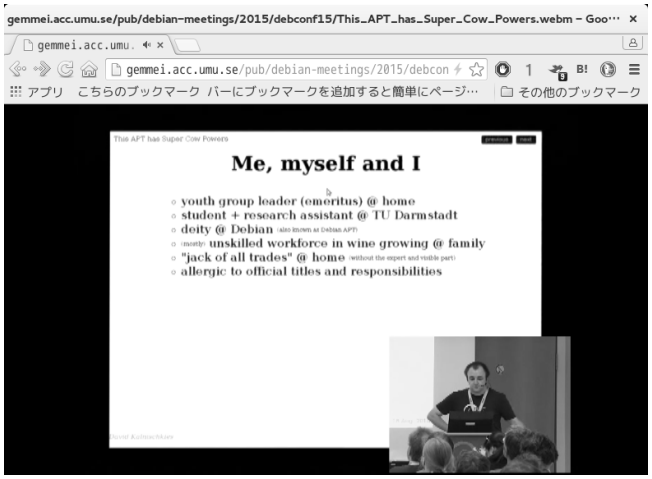


図 8　Debconf15 の APT1.1 のセッションの様子

5.3 APT 1.1 を評価してみる

2015/8/22 現在、APT 1.1 は experimental リポジトリにあります。早速、試して見たい方は以下のようにすることで導入できます。

```
$ sudo vi /etc/apt/source.lists
deb http://ftp.jp.debian.org/debian/ experimental main contrib non-free
deb-src http://ftp.jp.debian.org/debian/ experimental main contrib non-free
$ sudo apt-get update
$ sudo apt-get -t experimental install apt
$ apt
apt 1.1~exp9 (amd64)
Usage: apt [options] command
... 中略...
 full-upgrade - upgrade the system by removing/installing/upgrading packages
 edit-sources - edit the source information file
```

手元の Debian sid 環境に導入してしばらく使っていますが、特に動作に問題は見当たりませんでした。apt update,apt full-upgrade, apt autoremove など一通りの動作は問題なくできており、致命的なバグにも特に遭遇していません。是非お試しあれ。

5.4 APT 1.1 の特徴

jessie 搭載の APT 1.0.9.8 に比べての違いを述べてみます。

先に紹介した Debconf15 のビデオによれば以下の点が違いとなります。

- リポジトリの情報のセキュリティ検証が強化された。
- deb822 形式でリポジトリを指定するやり方にて機能強化。(/etc/apt/*.sources ファイル)
- httpredir.debian.org を受けて、処理の途中経過の表示を変更。
- Pinning がちゃんと動くようになった。
- 依存関係をする場合、ローカルにおいた.deb ファイルを直接指定してもインストールでき、ソースビルドの依存関係を指定する方法が柔軟になった。

まずは、man apt、man apt-get で記載されている内容で、トピックを絞って違いを紹介します。

5.4.1 apt autoremove

APT1.1 に搭載された apt コマンドの autoremove 命令について説明します。

何かパッケージをインストールした場合、依存関係を満たすためだけにインストールされたパッケージが過去にあるとします。ここで、現在はその依存関係からも外されており、もはや全く使われていないパッケージがあります。このコマンドをつけて apt コマンドを起動すると、こういった使われていないパッケージを削除することが出来ます。

```
$ sudo apt autoremove
```

5.4.2 autoremove 仕組み

autoremove はどのようにして消去すべきパッケージを見つけるのでしょうか？

依存関係が見当たらないパッケージの中には、利用者が自分で明示して入れたパッケージもあります。そのため、必ずしも他のパッケージで依存していないということだけを条件にして、autoremove でパッケージを消すようなことは避けなければなりません。autoremove により、自動で消去して良いパッケージを判断する基準は次の通りです。

Step 1. /var/lib/apt/extended_states ファイルの記録に、過去、依存関係を満たすためにパッケージを導入したかどうかの記録である"Auto-Installed: 1" と記されているパッケージを消去の候補とする。

Step 2. すでにインストール済パッケージのどれからも依存関係に無いかどうか？

Step 3. さらに、

- Recommends として提案されているパッケージはインストールして欲しいと明示した場合、
- Suggest として提案されているパッケージはインストールして欲しいと明示した場合、

のいずれにも該当していないか？

以上の Step 1.〜3. の判断を経たパッケージが、自動で消去して良いパッケージとして扱われます。

5.4.3 man apt-get との違い

次に man apt-get で見た 1.0.9.8 との違いについて紹介します。

- `indextargets`
  apt-get update で更新されるファイルと状態を deb822 形式で表示します。
- `--allow-downgrades`
  特定パッケージをダウングレードすることにより依存関係が満たせるときに、ユーザに尋ねず実行してまうというオプションです。
- `--allow-remove-essential`
  何らかの理由により Debian システムの必須パッケージ（essential パッケージとして分類されている）ものを消せば依存関係が満たせるときに、消してよいか？を尋ねず消してしまうオプションです。
- `--allow-change-held-packages`
  何らかの理由により Hold 扱いにしたパッケージを削除すれば依存関係が満たせるときに、消してよいか？を尋ねず消してしまうオプションです。
- `--no-allow-insecure-repositories`
  リポジトリにある Release ファイル (InRelease ファイル）の GPG による署名が確認出来ない等、セキュリティ上問題があるとみなされたリポジトリが含まれた場合、apt-get update 操作を失敗させます。

5.4.4 リポジトリ堅牢化

APT 1.1 はリポジトリのセキュリティの正当性評価が強化されています。正当性評価の元となるファイルに Release ファイル (InRelease ファイル）があります。

```
$ curl http://cloudfront.debian.net/debian/dists/unstable/InRelease
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

Origin: Debian
Label: Debian
... 中略...
MD5Sum:
 e9f9b477f2430a7d0e2dd686da1af507 30975818 Contents-amd64.gz
 d158f809191a841bedf9ff50e34e0ebe 30421142 Contents-armel.gz
... 中略...
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1

iQIcBAEBCAAGBQJV1+IMAAoJEItIrWJGklVTgloP/O+XAch/TMtTSfH+N1QFl+q2
Woas1LpWhHDO12U6vuPq5wghCPYE5ctNuDxFtTy9j01lsf6kWXPDh1QupNENDNHr
lfZ7Qa9gFr8W3tH1tnPwsSqcQmu9bMkROsRDVSfcFlDioVhN/h+jWW7j7J7nrZrE
... 中略...
```

InRelease ファイルを見るとわかるとおり、

- リポジトリに含まれる様々なファイルは全て md5sum 付きで InRelease ファイルに記録
- さらに InRelease ファイルも電子署名による正当性確認が出来るようになっている

となります。

今回 APT1.1 では、基本的に InRelease ファイルの無い、あるいは、他に必要なファイルが欠落しているなど、セキュリティ観点からの正当性確認が出来ないリポジトリは取り扱いを完全にやめる設計にしたとのことです。

5.4.5 httpredir.debian.org 対応

2015/5 月頃、Debian ユーザに最も近い mirror サーバーを HTTP Redirect で apt に教えてくれるサービスが稼働しました。つまり、ユーザは/etc/apt/sources.list に、httpredir.debian.org を指定すれば、ユーザに最も近い mirror サーバーへリダイレクトされます。

ここで、リダイレクトされた結果どこのサーバから取得するのか？がわかると便利な事が多いです。このため、APT1.1 の apt/apt-get はリダイレクトされた先の情報を表示するように変更されました。

5.4.6 参考：httpredir.debian.org の様子

httpredir.debian.org が何を返却するのかを以下に示します。

```
$curl -v http://httpredir.debian.org/debian/dists/sid/InRelease
> GET /debian/dists/sid/InRelease HTTP/1.1
> Host: httpredir.debian.org
> User-Agent: curl/7.44.0
> Accept: */*
>
< HTTP/1.1 302 Found
< Date: Sat, 22 Aug 2015 03:54:12 GMT
< Location: http://cloudfront.debian.net/debian/dists/sid/InRelease
< Content-Type: text/plain
... 省略...
```

見てのとおり、cloudfront.debian.net にリダイレクトが指示されるのが確認できると思います。

cloudfront.debian.net?

前ページのリダイレクト先にて、`http://cloudfront.debian.net/` とリポジトリが提案されています。これは、2 年前に debian-cloud チームでアナウンスがあった、AWS の cloudfront という CDN の仕組みを使ってデータ配布を行う試みのリポジトリです。(`https://lists.debian.org/debian-cloud/2013/05/msg00066.html`)
もともと、CDN はユーザに最も効率的なサーバを提示してデータを配る仕組みであり、AWS の cloudfront は相当な規模とサービスエリアを持つ CDN サービスですので、そもそもこちらがあるなら、AWS のサービスがカバーしている国では、httpredir.debian.org を使わなくてすみそうな気もします。しかしながら、ソフトウェア自由をモットーとする Debian としては、一企業のサービスに依存しないようにすることが重要ですので、Debian としては、httpredir.debian.org を維持・運用する必要があります。

5.5 deb822 形式

APT 1.1 では、deb822 形式でリポジトリを指定するやり方にて機能強化が図られました。ここでは、deb822 形式とはどんなものかを紹介します。

APT1.1 が手元にあれば、簡単に deb822 形式でリポジトリの情報を表示させる事が出来ます。

```
$ apt-get indextargets
MetaKey: main/source/Sources
ShortDesc: Sources
Description: http://ftp.jp.debian.org/debian sid/main Sources
URI: http://ftp.jp.debian.org/debian/dists/sid/main/source/Sources
Filename: /var/lib/apt/lists/ftp.jp.debian.org_debian_dists_sid_main_source_Sources
Optional: no
Codename: sid
... 中略...
```

出力されるフォーマットを見るとわかるのですが、RFC822 ヘッダの形式によく似ています。ここから、deb822 と名前を取ったようです。

特徴として、RFC822 と同様ですので、ヘッダを増やせば、簡単に機能拡張できるという点が上げられます。

また、動作未確認ですが、DebConf15 のビデオによれば、

/etc/apt/source.list.d/xxxx.sources

（末尾が、.sources である事が必要）という名前で deb822 形式で置いておくと、こちらを sources.list に指定したのと同様の動作を apt/apt-get は行うとのことです。

5.6 牛さんパワー健在!

APT 1.1 にも “moo” 命令は健在です。息抜きに、こちらを紹介しておきます。
man apt には記載の無いオプションですが、以下に apt の場合の起動方法を載せます。

```
$ apt moo
$ apt moo moo [--color]
$ apt moo moo moo
$ faketime '1997-04-01' apt moo
```

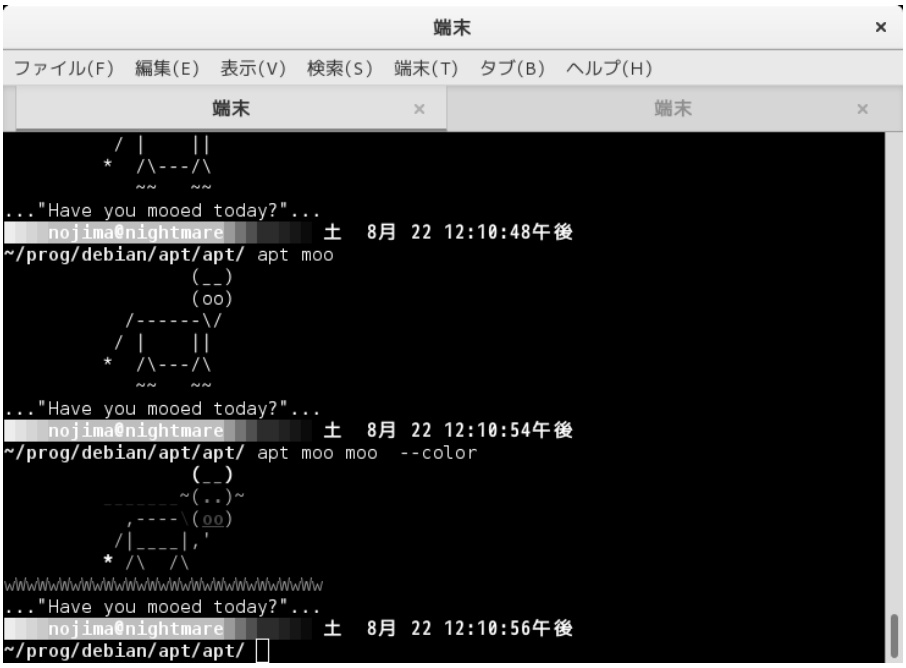


図 9 apt moo コマンド結果例

実行するとわかるのですが、ASCII キャラクタで描画された各種の牛の絵が表示されます。

5.7 おわりに

まだまだ、APT1.1 について、今回ここでは書ききれない程の変更が加えられているようですが、このへんにしておきます。実際、git で落として差分を確認しましたが、実に 2 万行を超える変更が行われていました。正式リリースになって、ドキュメントも充実すると良いですね。

6 Debian で HTTP/2 を試す

野島 貴英

6.1 はじめに

2015/5 に HTTP/2 が RFC7540 として遂に文章化されました。また、最近でも、ほうぼうで WEB ページあるいはサービスについて、HTTP/2 の対応度合いについて聞かれるようになってきました[*17]。

ここでは、Debian で、HTTP/2 の環境をちょっと作って試してみました。

6.2 ところで HTTP/2って？

WEB ブラウザがサーバと通信する際に、HTTP/1.x(x は 0,1 の数字）が長年 (HTTP/1.1 は 15 年以上も！) 使われています。しかしながら、昨今の WEB ページは、HTTP/1.x が策定された頃に比べて格段にリッチなページとなっており、１ページを表示する為に必要な通信量は格段に増えています。HTTP/1.x のままでは、WEB の通信が非効率となってしまいました。

HTTP/1.1 の欠点を克服するために、google 社で SPDY が開発され、さらに SPDY を参考にして、沢山の人の貢献により、次世代の HTTP 通信規格が策定されました。これが HTTP/2 となります。

6.3 HTTP/2 の特徴

HTTP/2 の特徴としては以下の通りです [1]。

- テキスト電文ベースではなく、バイナリ電文を使います。
- １本の TCP コネクション上で、複数のリクエスト・レスポンスを多重化してやりとりできるように設計されています。
- リクエスト・レスポンスに使われるヘッダ情報を無駄の無い電文にし、さらに圧縮し、より効率的に通信できるようにしています。実はモバイル端末などでは、パケットの往復にかかる時間が長いので、リクエスト・レスポンスの開始１パケット目にできるだけ情報を詰め込むことは通信時間を縮めるのに非常に有効です。
- １つのリクエストで、ブラウザが続けて必要とするデータをまとめてレスポンスできる機能が入りました（サーバプッシュという機能。)[2]

6.4 HTTP/2 さらに詳しく

これ以上 HTTP/2 プロトコルについて、詳しく調べたい人は、

[*17] 某有名携帯電話イベントで HTTP/2 をアプリ開発者に全力推奨している件を見て慌てたりしたのは秘密...

- HTTP/2 本家
  https://http2.github.io/
- 高速・大規模ネットワーク時代に向けて改良された HTTP/2 プロトコル http://www.atmarkit.co.jp/ait/articles/1409/18/news135.html
- twitter の#http2study タグ
- http/2 Advent Calender 2014 http://qiita.com/advent-calendar/2014/http2

などなど、多数の良い解説がありますので是非ご覧ください。これ以上細かい HTTP/2 のプロトコル仕様についてはここでは割愛します。

6.5 HTTP/2 の良いデモサイト

論より証拠で、HTTP/2 が優れているか？を試せる非常に良いデモサイトがあります。是非お試し下さい。なお、Debian sid の chromium/iceweasel で動作確認を確認できています。

https://http2.golang.org/gophertiles

6.6 HTTP/2 の通信開始の仕方

HTTP/2 を使ってブラウザからアクセスするとき、現状、TLS の ALPN/NPN で HTTP/2 を指定して、やりとりを開始するやり方しか現行ブラウザには実装されていません。というわけで、事実上、HTTP/2 のサイトは、全部フル SSL 化されている状況となります [3]。

一方、TLS を使わない場合、HTTP/1.1 のリクエストヘッダに特別なヘッダを混ぜることにより、HTTP/2 へアップグレードして、以降 HTTP/2 でやり取りをするという手法があります。しかしながら、こちらはブラウザが対応していません [4]。

6.7 Debian で HTTP/2 をお手軽に楽しむ

HTTP/2 を Debian でお手軽に楽しむには以下の環境を用意します。

- クライアント側準備
  chromium か、iceweasel
- サーバー側準備
  nghttp2,Apache Traffic Server 等など

6.8 クライアント側準備

以下に chromium と、iceweasel について、HTTP/2 用を評価するのに便利なセットアップについて載せます。

6.8.1 chromium

chromium を使う場合は次の通りです。まず、Debian に chromium ブラウザを導入します。

```
$ sudo apt-get install chromium
```

次に、chromium を起動して左隅みに現れる「Apps」→「Web Store」をアクセスし、「HTTP/2 and SPDY indicator」を導入してください。

以上の操作を行った chromium で HTTP/2 対応のサイトにアクセスすると、青い稲妻マークがアドレスバーに

表示されるようになります。

図 10　chromium で HTTP/2 のサイトにアクセス

6.8.2　iceweasel

iceweasel を使う場合は次の通りです。まず、iceweasel と、xul-ext-spdy-indicator を Debian に導入します。

```
$ sudo apt-get install iceweasel xul-ext-spdy-indicator
```

以上の操作を行った iceweasel で HTTP/2 対応のサイトにアクセスすると、青い稲妻マークがアドレスバーに表示されるようになります。

図 11　iceweasel で HTTP/2 のサイトにアクセス

6.9　サーバ側準備

いよいよ Debian にサーバ側を準備します。

6.9.1　HTTP/2 に対応しているサーバ

どんなサーバが HTTP/2 に対応しているかは、

Implementations
`https://github.com/http2/http2-spec/wiki/Implementations`

を参照ください。

6.9.2　nghttp2 パッケージ

Debian sid にて、HTTP/2 対応サーバのパッケージとして、nghttp2 があります。ここではこちらを使ってサーバを作ることにします。

```
$ sudo apt-get install nghttp2 ssl-cert
```

なお、閲覧可能なコンテンツとして、groff の付属 html マニュアルをドキュメントルートにした HTTP/2 サーバ

を立ててみます。

なお、*-snakeoil.*というファイルは、ssl-crt パッケージを導入すると勝手に作成される自己証明書ファイルとなります。

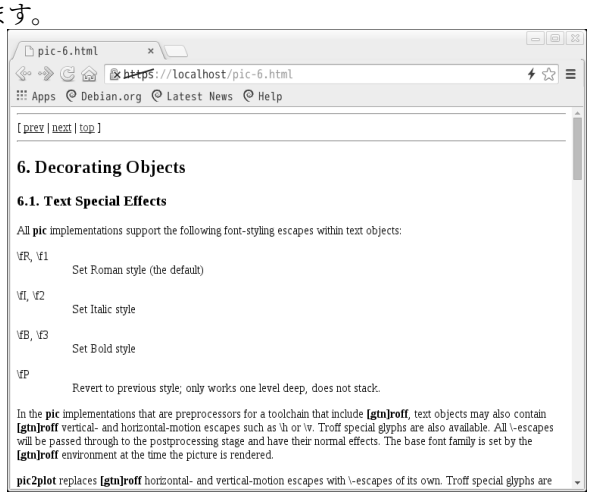
```
$ sudo nghttpd -D -d /usr/share/doc/groff-base/html/ \
  443 /etc/ssl/private/ssl-cert-snakeoil.key \
  /etc/ssl/certs/ssl-cert-snakeoil.pem
```

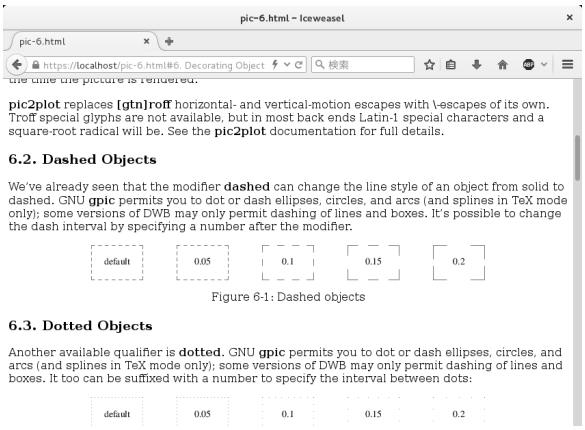
6.10 アクセスしてみる

ブラウザで、アクセスしてみます。

アクセス先：`https://localhost/pic-6.html`

chromium/iceweasel 共に無事に HTTP/2 対応を示す青い稲妻マークが URL 表示部分に付いていることが判ります。





6.11 proxy サーバで既存サイトの HTTP/2 化をやってみる

さて、nghttpd は軽量の HTTP/2 対応 WEB サーバではあるのですが、やっぱり apache のような高機能な WEB サーバを使って HTTP/2 を実現したいというニーズがあると思います。（例：php のサイトを HTTP/2 化したい等）

今度は、apache をバックエンドにして、nghttp2 付属の proxy サーバを使い、サイトの HTTP/2 化を行ってみます。

今回用意しようとしている環境の概念図を載せます。

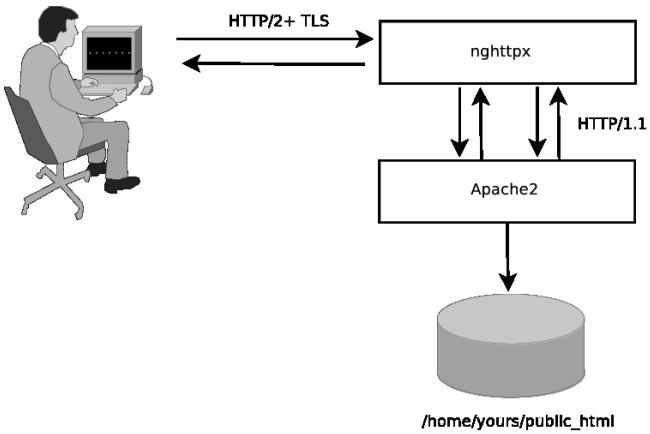


図 12　proxy サーバで HTTP/2 化を行う環境の概念図

proxy と apache の環境を Debian に用意します。手順は次の通りです。

Step 1. sudo apt-get install apache2 nghttp2 ssl-cert

Step 2. sudo a2enmod userdir

Step 3. cd /home/yours/; mkdir public_html

Step 4. cd public_html; cp -a /usr/share/doc/groff-base/html .

Step 5. sudo vi /etc/nghttpx/nghttpx.conf

```
 ----nghttpx.conf の中身ここから----
frontend=0.0.0.0,443
backend=127.0.0.1,80
private-key-file=/etc/ssl/private/ssl-cert-snakeoil.key
certificate-file=/etc/ssl/certs/ssl-cert-snakeoil.pem
workers=1
----ここまで----
```

Step 6. sudo systemctl start apache2.service

Step 7. sudo nghttpx -D –conf /etc/nghttpx/nghttpx.conf

以上できましたら、いよいよ先に用意したブラウザからアクセスしてみます。無事 apache 側に用意したサイトが、HTTP/2 対応できている事がわかります。

アクセス先：`http://localhost/~yours/html/pic.html`

6.12 おわりに

HTTP/2 も Debian を使えば簡単に実験できます。また、沢山のファイルで構成されるページがあると、HTTP/2 は非常に威力を発揮します。この機会に HTTP/2 を是非お試し頂き、その威力を確認してみてください。

参考文献

[1] HTTP/2 Frequently Asked Questions,https://http2.github.io/faq/

[2] 初めての HTTP/2 サーバプッシュ,http://labs.gree.jp/blog/2014/12/11987/

[3] wikipedia HTTP/2 の章,https://ja.wikipedia.org/wiki/HTTP/2

[4] HTTP/2 プロトコルネゴシエーション方法と ATS での実装,http://techblog.yahoo.co.jp/infrastructure/http2/ats_http2_pn/

7 NTT フレッツ網経由で Native IPv6

Roger Shimizu

7.1 はじめに

NTT フレッツを使っている自宅で IPv6 インターネットが動かなかったんです。
NTT NGN 網を使うと、どこかから IPv6 アドレスが割り当てられますし、ゲートウェイも自動的に設定されますが、Internet に繋がらない！
原因は、Dual スタック対応済み DNS が AAAA record (IPv6 address) の結果を、A record と一緒に返します。

- Google 's DNS: 8.8.8.8 / 8.8.4.4
- NTT 's DNS: 129.250.35.250 (関東) / 129.250.35.251 (関西)

OS 側では、IPv6 があれば、IPv4 より IPv6 が優先的に使われます。
しかし、IPv4 へのフォールバックが発生してネットアクセスが非常に遅くなってしまう。
NTT ではそういう状況が把握されているようです、その解決方法はなんと「IPv6 を無効」[*18]となります！

7.2 Debian ではどうやって IPv6 を無効にするの？

kmod 設定ファイル (squeeze まで使用可能): /etc/modprobe.d/aliases

```
alias net-pf-10 off
```

sysctl 設定ファイル (動的に変更可能): /etc/sysctl.conf

```
net/ipv6/conf/all/disable_ipv6 = 1
```

起動時の kernel パラメータ (最初から無効になる):

```
ipv6.disable=1
```

/etc/default/grub の GRUB_CMDLINE_LINUX 変数に追加が必要です。Debian Installer なら、起動時に`<TAB>`をしてから上記のパラメータを入力できます。

[*18] https://flets.com/customer/ipv6_display.html

7.3 IPv6 体験のための トンネル方式の使い方

Native 環境が見つからなければ、トンネル方式でなど色々体験する方法があります。

- 6in4 (proto-41) (例、`Hurricane Electric` **さんの** `tunnelbroker` **サービス**[*19])
- Teredo (例、Debian で miredo というパッケージをインストールすればすぐに使えます)
- SixXS
- AICCU
- AYIYA
- 6to4 (via 192.88.99.1)
- 6over4 (fe80:: & IPv4)
- freenet6

体験・検証ぐらいならそれで良いけど、従来 IPv4 の性能と比べて遅くて、通常に IP トンネルを使用することはおすすめしないと考えております。

7.4 Tunnel で遅くなる原因の解析

普通にウェブアクセスが Tunnel 経由なら遅くなる原因は DNS + CDN となります：

- DNS のレスポンス時間
- CDN のアクセス時間

DNS はどこを使うべきか？例として、アメリカ西海岸の tunnelbroker.net を使うなら、ケース別で解析して見ましょう。

- Local DNS
  プロバイダー (ISP) から提供される DNS となり、IPv4 で CDN が日本のサーバに直接アクセス出来て、特に問題がないけれど、IPv6 だと Host → US Tunnel → JP CDN にしてしまい、往復で 250ms 位になります。
- Remote DNS
  トンネル経由で US 側の DNS となり、毎回 DNS query のコストが 120ms 位になりますし、IPv4/IPv6 両方ウェブアクセス時間も 120ms を加算されるとなってしまいます。

どっち側の DNS を使っても、遅くなることが分かりました。

7.5 NTT フレッツで Native IPv6 ができます！

Native IPv6 のメリットというと、Local DNS を使うことで、DNS Query コストがあまり掛からないし、IPv4/IPv6 両方共にローカル (又は近くに) CDN が使えます。多くプロバイダーは IPv6 が対応するようになりました。

- OCN: `http://service.ocn.ne.jp/ipv6/access/`
- Plala: `http://www.plala.or.jp/ipv6/service/area/`
- So-net: `http://www.so-net.ne.jp/common/IPv6/`
- 他の ISP: `http://www.jaipa.or.jp/ipv6/`

すべては確認していませんが、主な OCN/So-net/Plala などは既に Native IPv6 が対応されているようです。それから、簡単に Dual Stack が構築出来ます！

[*19] `https://www.tunnelbroker.com`

7.6 具体的な IPv6 の設定方法

Dual Stack なら、2 本 PPP セッションが必要。例えば、

- ppp0: IPv4
- ppp1: IPv6

IPv4 の PPPoE 設定は従来通りで良くて、IPv6 の方は IPv4 の設定をベースにして、以下の変更が必要となります。

- cp /etc/ppp/peers/dsl /etc/ppp/peers/dslv6
- echo +ipv6 >> /etc/ppp/peers/dslv6
- /etc/ppp/peers/dslv6 に、元 IPv4 のアカウントを IPv6 版に書き換える
- /etc/ppp/chap-secrets に、IPv6 アカウントのパスワードを追加します (ID は IPv4 / IPv6 別ですが、パスワードは一緒になるケースがほとんどです)

IPv6 PPPoE アカウント (CHAP ID) については、プロバイダーによります。
例としては、(太字は IPv4 の設定に加えた部分です)

- `OCN:`[*20]blah@`ipv6`.ocn.ne.jp
- `OCN biz:`[*21]blah@bizf6.ocn.ne.jp 又は blah@bizd6.ocn.ne.jp
- `Plala:`[*22]blah@`v6h`.plala.or.jp 又は blah@`v6m`.plala.or.jp
- `So-net:`[*23]taro@aa2-`v6`.so-net.ne.jp

また、IPv6 address と default route もそれぞれ設定が必要です。

- IPv6 address は DHCPv6 クライアント (wide-dhcpv6-client など) で取得します。
- IPv6 default route は手動設定となり、[*24][*25]

  ```
  ip -6 r add default dev ppp1
  ```

  で済みます。

設定の参考: `https://youtu.be/bJ9p2j9frtA` (git repo: `https://github.com/rogers0/config/tree/network/flets-native-v4v6`)

7.7 用語定義

- stateless host: IPv6 address と default gateway は RA (ブロードキャスト) による取得します (デフォルト)。
- stateful host: IPv6 address は DHCPv6 クライアントとして、サーバから取得します。
- gateway: IPv4 の NAT 機器のような IP パケット転送の機器となります。IPv4 NAT の場合は Address 変換するんですが、IPv6 の場合はルータ機能を加えられます。

[*20] http://service.ocn.ne.jp/ipv6/access/flow/
[*21] http://www.ocn.ne.jp/business/ftth/withf/spec.html
[*22] http://www.plala.or.jp/ipv6/access/flow/
[*23] http://www.so-net.ne.jp/option/others/ipv6/
[*24] https://bugs.debian.org/477245
[*25] https://github.com/paulusmack/ppp/issues/40

7.8 解決案 0: stateful gateway 向け

デフォルトの stateless host から、stateful host の方式にすると、NTT NGN からの RA (ブロードキャスト) を受けないようになります。

```
sysctl.conf
        net/ipv6/conf/default/accept_ra = 0
        net/ipv6/conf/all/accept_ra = 0
        net/ipv6/conf/eth0/accept_ra = 0
        net/ipv6/conf/wlan0/accept_ra = 0
```

7.9 解決案 1: stateless host 向け

MSFT の KB[*26]によると解決方法がありました。それから、他の記事[*27]も参考にできます。address selection で NTT NGN 用の prefix の優先度を下げると、問題が解決できます。提示する win32 コマンドを Linux に翻訳すると、

```
        ip addrlabel add prefix 2001:c90::/32   label 8
        ip addrlabel add prefix 2404:1a8::/32   label 8
        ip addrlabel add prefix 2408::/22           label 8
        ip addrlabel add prefix 2001:d70::/30   label 8
        ip addrlabel add prefix 2001:a000::/21 label 8
```

となります。参考の設定: https://github.com/rogers0/config/tree/network/stateful_v6host

7.10 ゴールまで足りないもの (TODO)

- Firewall: ip6tables
- stateful IPv6 gateway (allow IPv6 forward)
- stateful/stateless IPv6 host (disallow IPv6 forward)

[*26] https://support.microsoft.com/ja-jp/kb/2551233
[*27] http://www.attn.jp/maz/p/i/policy-table/

8 wiki:Subkeys

かわだてつたろう

Debian Wiki[*28]には Debian に関する様々な情報がまとめられています。そのなかで OpenPGP のサブキー (副鍵) に関するページ wiki:Subkeys[*29] がありましたので、その内容を紹介します。

8.1 鍵とは

OpenPGP は公開鍵暗号方式で秘密鍵と公開鍵の 2 つの鍵から成り立つ暗号方式です。その名前の通り、公開鍵は公開するための鍵、秘密鍵は自分だけが使う鍵です。秘密鍵は署名や公開鍵で暗号化されたメッセージの復号化に使用します。

8.2 副鍵とは

鍵を作成時に必ず作成される鍵ペアが主鍵 (マスターキー) です。この他に作成する鍵ペアを副鍵と呼びます。副鍵は署名、暗号化など通常の鍵として使用でき、主鍵とは別に破棄することもできます。つまり、主鍵と結び付いている独立した鍵ペアといえます。

8.2.1 GnuPG での副鍵

GnuPG では、主鍵は署名にしか使用できません。暗号化するためには副鍵を作成する必要があります。

鍵作成時の次の選択で (署名のみ) を選択していなければ副鍵が作成されているはずです。

```
Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection?
```

このようになっているのは RSA が特許の関係で使用できなかったころ、ElGamal は暗号化のみ、DSA は署名のみにしか使用できないといった歴史的理由もあるようです。

8.3 なぜ副鍵

主鍵は非常に大事です。主鍵の秘密鍵が失なわれるとあなたの信用も失なわれ、信頼の輪をまた一から築きあげることになります。そのため、主鍵の秘密鍵は非常に非常に安全に保管しておく必要があります。しかし、安全にすればするほど日常の使用は不便なものになります。

これを解決するのに副鍵を使います。暗号化用の副鍵と同じように署名用の副鍵を作成し公開すれば、副鍵で署名

[*28] https://wiki.debian.org/
[*29] https://wiki.debian.org/Subkeys

して使うように副鍵で署名して使うことができます。

次のような鍵の変更は主鍵でしかできません。

- 他の人の鍵に署名を追加するか、既存の署名を取り消す
- 新しい UID を追加するか、UID に primary マークを付ける
- 新しい副鍵を作る
- 既存の UID または副鍵を失効する
- UID の preferences を変更する (例: setpref)
- 主鍵または副鍵の期限日を変更する
- 鍵の失効もしくは失効証明書の生成

Web of Trust のリンクは公開鍵と UID の組合せに対する署名です。OpenPGP では主鍵の秘密鍵から UID への署名のリンクで副鍵は関係がありません。主鍵が安全であれば副鍵だけ盗まれても副鍵だけ失効し再作成することで済みます。

8.4 どのようにするか

次の手順をとります。

1. 念のため既存の GnuPG ファイルのバックアップをとる

```
$ umask 077; tar -cf $HOME/gnupg-backup.tar -C $HOME .gnupg
```

2. 署名用の副鍵を作る

```
$ gpg --edit-key YOURMASTERKEYID
gpg> addkey
Key is protected.

You need a passphrase to unlock the secret key for
user: "WHO <who@example.org>"
2048-bit RSA key, ID AAAAAAAA, created 2015-01-01

Enter passphrase: YOURMASTERKEYPASSWORD

Please select what kind of key you want:
   (3) DSA (sign only)
   (4) RSA (sign only)
   (5) Elgamal (encrypt only)
   (6) RSA (encrypt only)
Your selection? 4

RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)

Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)

Key does not expire at all
Is this correct? (y/N) y
Really create? (y/N) y

gpg> save
```

3. $HOME/.gnupg を USB ドライブにコピーするなどして退避させる
4. 主鍵の秘密鍵を取り除いた状態にする GnuPG にそのためのコマンドがないので次の手順を踏む
   (a) 副鍵の秘密鍵をエクスポート

```
$ gpg --export-secret-subkeys YOURMASTERKEYID > secret-subkeys
```

   (b) 主鍵と副鍵の秘密鍵を削除

```
$ gpg --delete-secret-key YOURMASTERKEYID
```

(c) 副鍵の秘密鍵を戻す

```
$ gpg --import secret-subkeys
```

(d) sec が sec#と表示されており、主鍵の秘密鍵が取り除かれていることを確認する

```
$ gpg -K
------------------------
sec#  2048R/AAAAAAAA 2015-01-01
uid                  WHO <who@example.org>
ssb   2048R/BBBBBBBB 2015-01-01
ssb   2048R/CCCCCCCC 2015-01-01
```

5. 鍵のパスワードを変更する

```
$ gpg --edit-key YOURMASTERKEYID passwd
```

これで準備が整いました。後は通常通り使用するだけです。

8.4.1 主鍵が必要な場合は

環境変数 GNUPGHOME か--homedir オプションで退避させた.gnupg ディレクトリを指定して使用します。

```
$ export GNUPGHOME=/path/to/save/.gnupg
$ gpg -K
```

```
$ gpg --homedir=/path/to/save/.gnupg -K
```

8.4.2 cross-certification

署名用の副鍵は主鍵と cross-certification(相互署名) しておくことがすすめられています。

主鍵と副鍵が同一の ID に属していることを保障するために、主鍵と副鍵の鍵束は主鍵によって署名されます。これで鍵束が主鍵に属していることがわかりますが、副鍵から見た場合、本当にこの副鍵が主鍵の持ち主のものかがわかりません。他人が副鍵を入手して主鍵で鍵束に署名した状態と区別がつかないためです。そのため、副鍵でも鍵束に署名するのが相互署名です。

GnuPG で相互署名するには次のコマンドを実行します。

```
$ gpg --edit-key YOURMASTERKEYID
gpg> cross-certify
```

相互署名してない場合、GnuPG が次の警告を出します。

```
gpg: WARNING: signing subkey CCCCCCCC is not cross-certified
gpg: please see http://www.gnupg.org/faq/subkey-cross-certify.html for more information
gpg: Can't check signature: general error
```

次のところを参考にしてください。

- Signing Subkey Cross-Certification --- GnuPG.org[*30]
- [mew-dist 28255] Re: gnupg 1.4.9[*31]

[*30] https://www.gnupg.org/faq/subkey-cross-certify.html
[*31] http://www.mew.org/ml-archives/mew-dist/2008-April/027942.html

8.5 それから何する

キーリング、キーサーバーへ登録します。

```
$ gpg --send-key YOURMASTERKEYID
```

8.6 まとめ

手順が少し手間ですが、主鍵を日常の使用と切り離せるのは安心感があります。

とはいえ、主鍵を安全にしておかないといけないことに変わりはありません。そのためには Gnuk Token[*32] などを使うのがよいかもしれません。

[*32] http://www.fsij.org/category/gnuk.html

9 DebConf15 ビデオ紹介

野島 貴英

9.1 はじめに

毎年 1 回、世界中の Debian Project 関係者及び熱心なユーザらが集まり、ハッカソンをしたり、発表をしたりするイベントとして、DebConf があります。16 回目[*33]の開催の DebConf15 が、2015/8/15-22 の間、ドイツのハイデルベルクで開かれました。

公式 URL は `http://debconf15.debconf.org/` となります。

ここでは、Debconf15 で行われたセッションのうち、字幕ファイルが用意されているものについて紹介してみます。

9.2 DebConf15 セッションのビデオ

DebConf では、Video Team が各セッションをビデオに撮り公開しているため、いつでもセッションの内容を見ることができます。なお、Debian はフリー（自由）にこだわるため、フリーなフォーマットである、webm が動画フォーマットとして利用されています。

掲載先：`http://debconf15.debconf.org/videostream.xhtml`

しかしながら、iphone/Android のスマートフォンで気軽に見たいという今時のニーズもあるかと思います。幸い、youtube でも DebConf15 のビデオが公開されていましたので紹介しておきます。

youtube:`https://www.youtube.com/playlist?list=PLz8ZG1e9MPlz2bUTzfgJhOJCxwT866D4w`

9.3 DebConf15 ビデオ字幕編

DebConf は世界中から Debian Project 関係者、及び、ユーザが集まるイベントですので、公用語は全て英語になります。発表も英語です。英語を母国語としない人にとってはヒアリングが苦手な方もいらっしゃいます。こういった人のために、現状、数は少ないですが、いくつかの英語の字幕が起こされています。

字幕取得先:`http://ftp.acc.umu.se/pub/debian-meetings/2015/debconf15/subtitles/english/`

字幕ファイルの使い方は次の通りです。

Step 1. 先の URL から、*.srt ファイルを取得する。

Step 2. totem/vlc/mplayer などで DebConf15 の動画を開き、字幕というメニューを選んで対応する.srt ファイルを指定します。ファイル名はセッションの名前になっています。

字幕付き再生を Debian sid 上で行っている様子を載せます。

[*33] DebConf 0 があるため

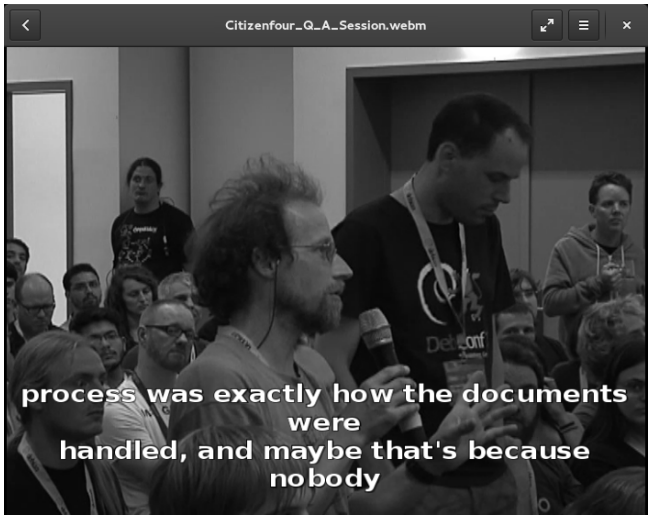


図 13　字幕付き再生例

9.4　今回のビデオ紹介

今回紹介予定の具体的なセッション名は、

- Stretching out for trustworthy reproducible builds
- Thanks for maintaining a desktop environment. But is it accessible?

となります。

9.5　Stretching out for trustworthy reproducible builds

ドイツの有名な Chaos Computer Club[*34]にも所属されている Debian 開発者らによる、Reproducible Builds についてのセッションです。

図 14　Stretching out for trustworthy(略) の発表

如何にビデオの内容をかいつまんで紹介します。

[*34] Wikipedia-jp で引いてみて下さい。ドイツの有名なコンピュータ技術のエキスパート集団。

9.5.1 動機

- The 31st Chaos Communication Congress (31C3)[*35]にて、パッケージのバイナリにトロイの木馬が巧妙に仕掛けられているか？を調べるには Reproducible Builds をしたほうが良いという発表を行ったとのことです。
- 31C3 のわずか数カ月後に今度は Edward Snowden さんにより、CIA の Strawhourse というコード名に関する CIA conference 2012 の内部文章がリークされました。内容は、MacOS/iOS の SDK に不当な改造を行い、生成されるバイナリに CIA が利用するためのトロイの木馬を仕掛けるという驚くべき内容でした。これにより、Reproducible Builds が益々急務になったとのことです。なお、リーク文章は、`https://theintercept.com/document/2015/03/10/strawhorse-attacking-macos-ios-software-development-kit/` で参照できます。

9.5.2 セキュリティ以外の良い点

- ビルド環境によらず同じバイナリができる、また、クロスビルドの確認ができるようになる、
- Debug package がいつでも（バイナリ作ったあとでも）作れるとか、
- FTBFS[*36]が早くわかるとか、
- バージョン上げた時の.deb の差分が小さくなるとのことです。

9.5.3 現在の Reproducible Builds 状況は次の通り

- Bitcoin/Tor/Coreboot は完了している。
- Debian/FreeBSD/NetBSD/OpenWrt は進行中。

9.5.4 工夫と苦労

- 環境変数 SOURCE_DATE_EPOCH に時刻 (エポック秒) を指定すると、その時刻でビルドしたようにビルドするように様々なツールを改造し upstream へ提供し取り込んでもらったそうです。なお、これだけでは足らないパッケージが沢山あったらしく、ビルドの日付が埋め込まれる部分を Reproducible Builds 出来ないと BTS したりして対策も多数したとのことです。
- tar にビルド環境の都度のユーザ名、グループ名が混じってしまう件を対策したそうです。
- ファイルシステムと locale 環境変数 (LANG,LC_ALL 変数) との違いによるソートの振る舞いの違い、プログラムの出力が異なってしまう件の対策をしたそうです。

言われてみると「なるほど！」と気がつく事ばかりで、考えてみれば相当に苦労するような内容ばかりでした。

9.5.5 視聴後の所感

よく、巷では簡単に Reproducible Builds は、パッケージのセキュリティ確認の為と簡単に紹介されますが、実は Debian を構成する重要なソフトウェア・パッケージの多くに手を加えなければ実現できないという大変な偉業を果たしていたという内容でした。思わず、これらの偉業に拍手をしたくなりました。

9.6 Thanks for maintaining a desktop environment. But is it accessible?

Debian Project にて Accessibility を担当されている方の発表となります。Accessibility に関しての現状と苦労がわかる発表内容となっています。

プレゼン資料は：`http://brl.thefreecat.org/2015-08-22-debconf.pdf`

[*35] Chaos Computer Club 主催の毎年行われるイベント
[*36] Fails To Build From Source の略

図 15　Thanks for maintaining a desktop...(略) の発表

9.6.1　Accessibility についてうっかりすると忘れがちになる大変重要な事

Free Software は、問題があったり、気に入らなかったら、自分で直せるということが基本であるが、Accessibility の機能を必要としている人は、基本的に治したくても直せない場合が多いので、コミュニティーによる修正・改善が必須です。

9.6.2　Linux Desktop 環境の Accessibility の現状

- Linux で動作する Desktop 環境は、GNOME が Accessibility が最もよくできている状況です。
- しかしながら、GNOME3 を持ってしても、Windows に比べると 10 年単位で遅れており、Apple の製品に比べると石器時代の代物と言われても仕方が無い状況です。
- 弱視の人には、合成音声によるサポートは厳しい場合（そもそも発音しにくいワードの場合など）があるため、理想的には、Piezo braille cell をサポートすべきです。

9.6.3　Linux Desktop 環境の Accessibility の現状の仕組みと開発方法

先のプレゼン資料、及び、ビデオにて

- Linux の Accessibility のフレームワーク、
- Linux の Accessibility のテスト環境など

が紹介されています。先述のプレゼン資料を参照ください。Accessibility がどのようにできていて、どうテストすべきかについて、非常に良い資料となっています。

9.7　おわりに

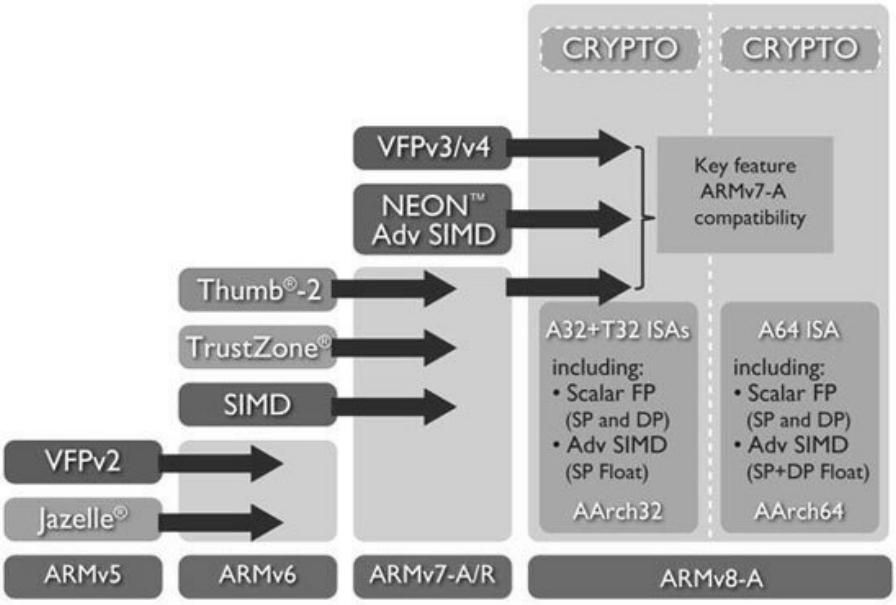
紙面の関係で今回はセッションを 2 つのみ紹介しましたが、他にも非常に興味深いセッションがあります。なにぶん英語のセッションなので、見て理解するのに苦労する状況ですが、その努力を払っただけの収穫があるのが、Debconf のビデオです。是非、１つ見て、内容を考えてみませんか？きっと、今まで見えていた世界ががらっと変わるような体験が出来ると思っています。

10 Debian と arm64 サポート

岩松 信洋

10.1 arm64

- Debian 8.0 から arm64 サポートが入った
- Debian でサポートする ARM アーキテクチャ
  - armel
    32bit / litte endian / ARMv5t
    古い NAS(QNAP、Buffalo、etc) やルータで使用されている ARM SoC で利用可能。
  - armhf
    32bit / litte endian / ARMv7 + VFP3（浮動小数点演算ユニット）
    Raspberry Pi 2 などで利用可能。
  - arm64 ← New!
- Raspbian
  - 32bit / litte endian / ARMv6 + VFP2（浮動小数点演算ユニット）
    Raspberry Pi 1 などで利用可能。
  - Raspbian in not Debian
- ARMv8 (ARM Version 8)



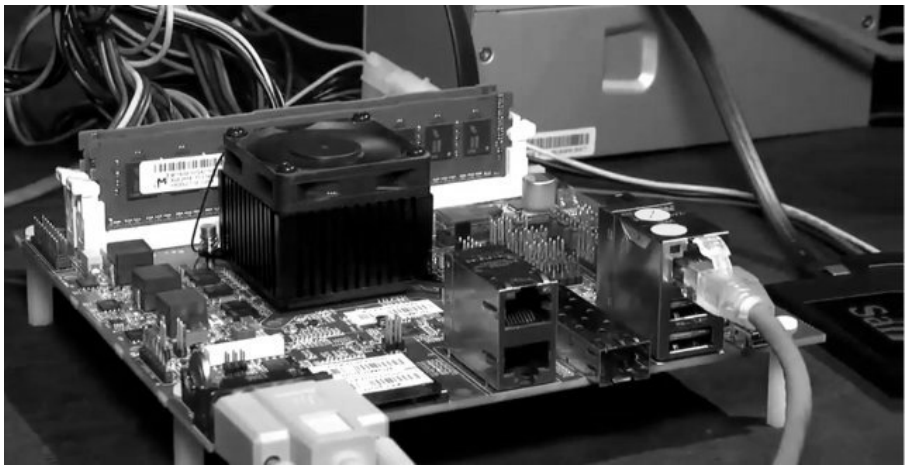
- ARMv8 (ARM Version 8)
- オリジナルコアとしては Cortex-A57、ARM Cortex-A53 と Cortex-A72 がある
- 正式名称は AArch64
- Linux kernel では わかりにくいということで arm64 に `https://lkml.org/lkml/2012/7/15/133`
- Debian もこれに追従して arm64 とした
- コンパイラなどのトリプレットは aarch64-linux-gnu
- GCC の定義は __aarch64__
- 紛らわしいので注意

10.2 Debian ARM 開発体制

- 2012 年から開発開始
- 開発に参加している多くの Debian Developer が Linaro 所属
  Steve McIntyre、 Wookey、Riku Voipio など
- GCC/binutils:Matthias Klose (GCC Upstream, Ubuntu Developer でもある)
- libc:Aurelien Jarno、libc メンテナチーム
- Linux kernel:Ben Hutchings(Linux 3.2 LTS メンテナ)、Ian Campbell（Xen、Allwinner SoCs 関連)、その他大勢

10.3 Debian ARM 開発体制

- Buildd: Applied Micro の X-gene を使ったサーバで運用中 `https://buildd.debian.org/status/architecture.php?a=arm64&suite=sid`
- SoC: X-C1 / 2.4Ghz / 8 コア 独自コア



10.4 クロスコンパイル環境

- Jessie リリース後 Debian のクロスコンパイル環境が変わった
- 今までは Emdebian から提供されているパッケージを使うか、ユーザ自身でパッケージ化する必要があった。
  ← 手間がかかる。
- GCC メンテナによりクロスコンパイル用パッケージが提供されるように
  - クロス用 binutils → binutils ソースパッケージ
  - クロス用 libc → cross-toolchain-base ソースパッケージ

– クロス用 GCC → gcc-5-cross ソースパッケージ

```
$ sudo apt-get install gcc-5-aarch64-linux-gnu
```

- リリース対象外のアーキテクチャは未サポート

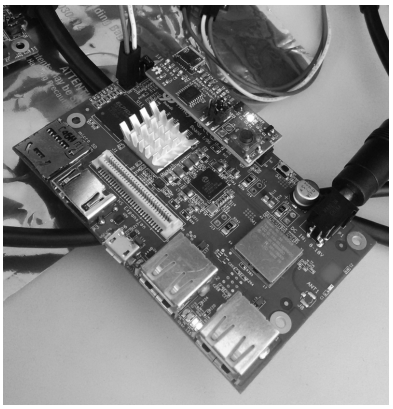
10.5 ユーザランドイメージ

- インストーラが用意されている
  `https://www.debian.org/CD/http-ftp/#stable`
- cdebootstrap を使うのが簡単

```
$ sudo cdebootstrap --foreign --arch arm64 \
      jessie /tmp/root http://http.debian.net/debian/
```

10.6 サポートボード

- Debian では ARM リファレンスボード（Juno）と X-gene（Applied Micro）のみサポート。
- arm64 のボードは値段が高い（10 万円以上）上に入手性が悪い。
- 96boards (Linaro Community Board Program) から入手するのがよさげ。約 1 万円。



  – Linux カーネルパッケージが更新され次第、Debian でもサポートする予定。
- QEMU を使って開発することも可能。ただし QEMU 2.0 以降。

10.7 ベンチマーク

ベンチマーク	Raspberry Pi 2	ODROID-XU4	Hikey
Dhrystone-2	1006.6	3994.1	2943.7
Double-Precision Whetstone	361.0	1024.9	680.3
Nbench 2.2.3 Integer	20.419	61.227	30.803
Nbench 2.2.3 FP	8.434	25.369	11.889

- Raspberry Pi 2: Broadcom BCM2836 900MHz ARM Cortex-A7 4 core
- ODROID-XU4: Samsung Exynos5422 Cortex-A15 2Ghz and Cortex-A7 Octa core CPUs
- Hikey: HiSilicon Kirin 6220 Cortex-A53 1.2Ghz Octa core

ODROID-XU4 > Hikey > Raspberry Pi 2

- いまのところ ARMv7 の方が速い。
- Cortex が遅いという話も。独自コアの SoC はそこそこ速い。
- といってもこのまま 32bit ARM を使っても 2038 年問題が。
- Cortex-A72 に期待。

10.8 まとめ

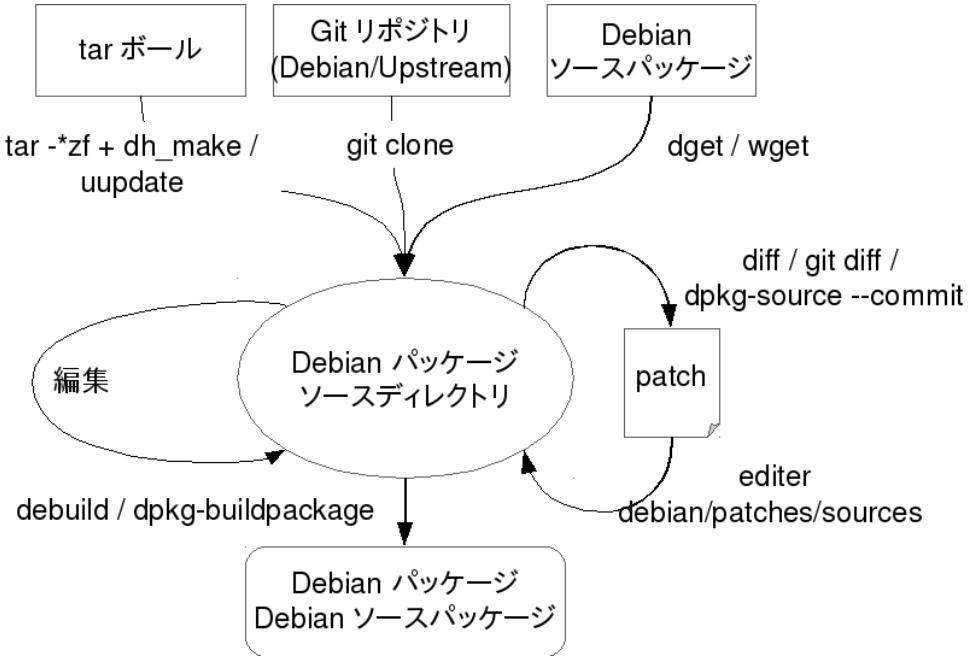
- Debian では ARM64 が既に使える環境が整っている。
- Upstream や 周辺組織との連携も十分。
- クロスコンパイル環境もオフィシャルでサポートされるようになり、今まで以上に開発しやすくなっている。
- ボードの供給問題がネック。一般向けは 96boards 頼り。
- 上記の理由もあり、サポートボードは少ない。今後頑張って増やします。

11 Debian パッケージング道場 git buildpackage の使い方

岩松 信洋

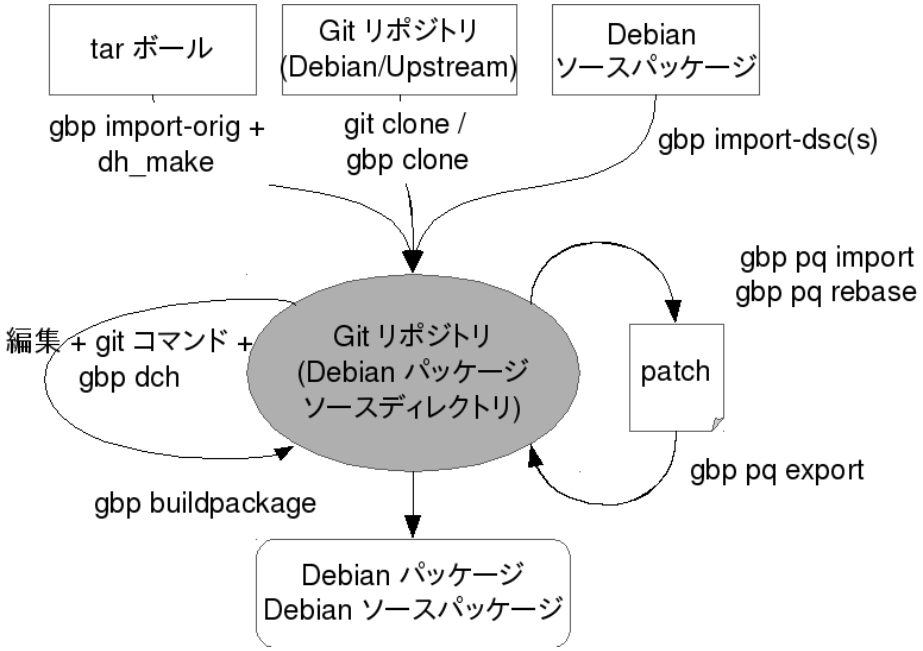
Debian パッケージングチュートリアル[*37]であまり解説されていない gbp (git buildpackage) の基本的な使い方について説明する。

11.1 VCS で管理しない場合



[*37] https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.ja.pdf

11.2 VCS で管理する場合



11.3 Upstream から tar ボールがリリースされている場合

Upstream から tar ボールがリリースされている場合、tar ボールを Git リポジトリにコミットした後、パッケージ化を行う。

- tar ボールをダウンロードする

```
$ wget package-name-0.0.1.tar.gz
```

- Git リポジトリを作成する

```
$ git init package-name
$ cd package-name
```

  必要であれば git config で Git の設定を変更する。

- tar ボールを指定して、ソースコードを Git にコミットする

```
$ gbp import-orig --pristine-tar \
        ../package-name-0.0.1.tar.gz
$ git tag
$ upstream/0.0.1
$ git branch
* master
  upstream
```

- Upstream のコードは upstream ブランチで管理され、同時にタグが設定される。
- pristine-tar オプションはオリジナルの tar ボールの差分を保存するための仕組みを有効にする。Upstream のコードは upstream ブランチで管理され、Debian パッケージを作成するときに、そこから orig.tar.gz ファイルを作成する。この時に作成されるファイルが同じものにならない場合がある。Debian では取り込んだ tat.gz ファイルのハッシュ値と Debian ソースパッケージとしてアップロードされる tar.gz ファイルが同じである必要があるため、本オプションを用いて orig.tar.gz との差分をバイナリパッチとして保存し、orig.tar.gz を構築する度に再適用することで、ハッシュ値が同じ orig.tar.gz を再構築できるようにしている。

- dh_make で debian ディレクトリの雛形を作成する

```
$ dh_make -p package-name_0.0.1
```

- debian ディレクトリ内をいろいろ変更する

```
$ いろいろ修正
$ git add debian
$ git commit
```

- パッケージ化作業ができたら パッケージを構築する

```
$ gbp buildpackage --git-pristine-tar
```

- piuparts などでインストール、アンインストールのテスト
- 最後にクリーンな環境でビルドテスト

```
$ git-pbuilder
or
$ gbp buildpackage --git-pbuiler --git-pristine-tar
```

- タグを設定して リモートリポジトリにプッシュする

```
$ gbp buildpackage --git-tag-only
$ git push
```

11.4 upstream に更新があった場合

- upstream の tar ボールを取得する

```
$ wget package-name-0.0.2.tar.gz
```

- tar ボールを指定してリポジトリにソースコードをコミットする

```
$ gbp import-orig --pristine-tar \
                ../package-name-0.0.2.tar.gz
```

`--uscan` オプションでダウンロード → コミットが一度にできる。また、ソースコードは自動的にマージされます。マージしたくない場合は`--no-merge` を指定して実行する。

- debian/changelog を修正

```
$ dch -i
or
$ gbp dch
```

- パッケージをビルド

```
$ gbp buildpackage --git-pristine-tar \
                --git-pristine-tar-commit
```

11.5 Upstream が Git で管理されている場合

- Upstream では tar ボールでリリースされず、Git のタグのみでリリースされる場合もある。
- Github で開発されているプロジェクトが良い例。
- このような場合は リポジトリをクローンした後、Debian 独自のブランチルールを用いてソースパッケージの管理を行う。

- リポジトリをクローンし、作成されたディレクトリに移動する

```
$ git clone git://example.org/git/package-name.git
$ cd package-name
```

- ベースにしたいバージョンのコードをチェックアウトする

```
$ git reset --hard 0.0.1
```

- タグを設定する

```
$ git tag upstream/0.0.1
```

upstream ネームスペースは gbp デフォルト参照先。Upstream のタグを使いたい場合や独自のネームスペースを使いたい場合は gbp の upstream-tag が利用できます。

```
[git-buildpackage]
upstream-tag = v%(version)s
```

- dh_make で debian ディレクトリの雛形を作成する

```
$ dh_make -p package-name_0.0.1
```

- debian ディレクトリ内をいろいろ変更する

```
$ いろいろ修正
$ git add debian
$ git commit
```

- パッケージ化作業ができたら パッケージを構築する

```
$ gbp buildpackage --git-pristine-tar \
                --git-pristine-tar-commit
```

tar ボールを使う場合と異なるのは`--git-pristine-tar-commit` オプションを指定すること。このオプションを指定することによってタグから orig.tar.gz を生成する。

- piuparts などでインストール、アンインストールのテスト
- 最後にクリーンな環境でビルドテスト

```
$ git-pbuilder
or
$ gbp buildpackage --git-pbuilder
```

- タグを設定して リモートリポジトリにプッシュする

```
$ gbp buildpackage --git-tag-only
$ git push
```

11.6 upstream に更新があった場合

- upstream のリポジトリ情報を取得する

```
$ git remote update
```

- 変更をマージ

```
$ git tag upstream/0.0.2 0.0.2
$ git merge upstream/0.0.2
```

- debian/changelog を修正

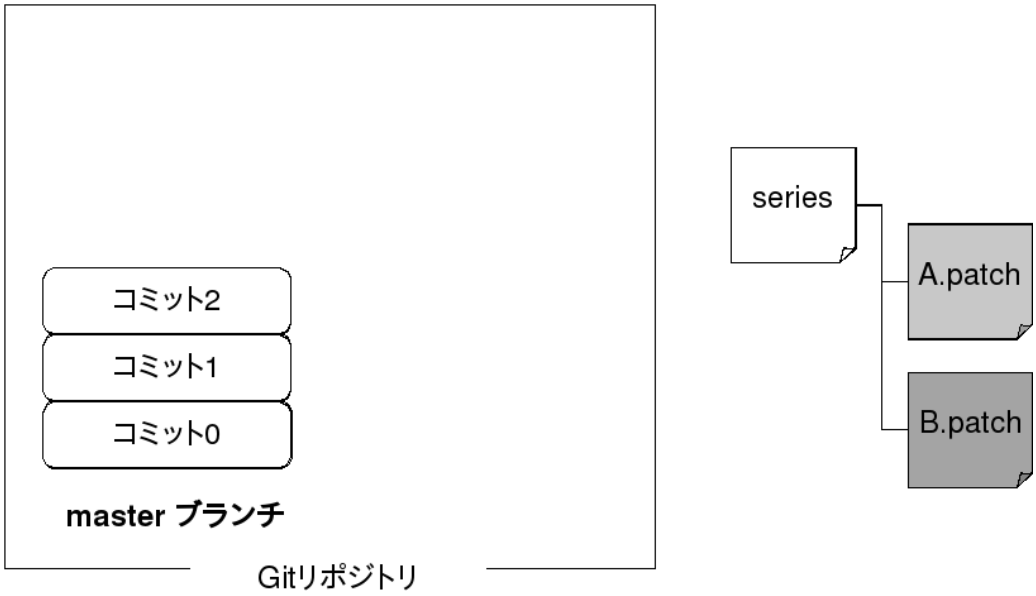
```
$ dch -i
or
$ gbp dch
```

- パッケージをビルド

```
$ gbp buildpackage --git-pristine-tar\
                --git-pristine-tar-commit
```

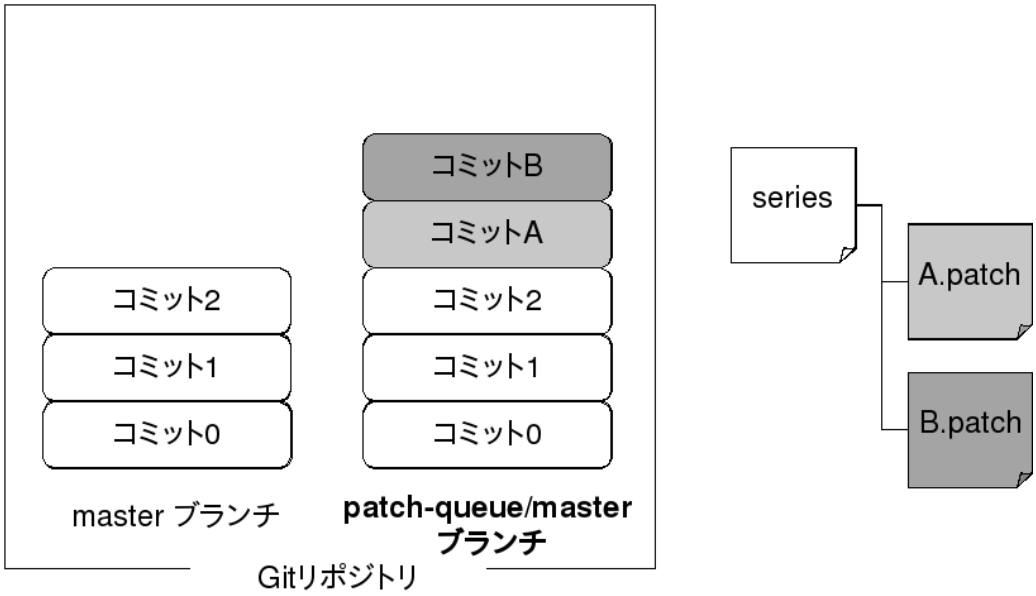
11.7 アップストリームのソースコード変更

1. gbp pq import
2. Upstream ソースコードの修正
3. git commit
4. gbp pq export
5. git commit



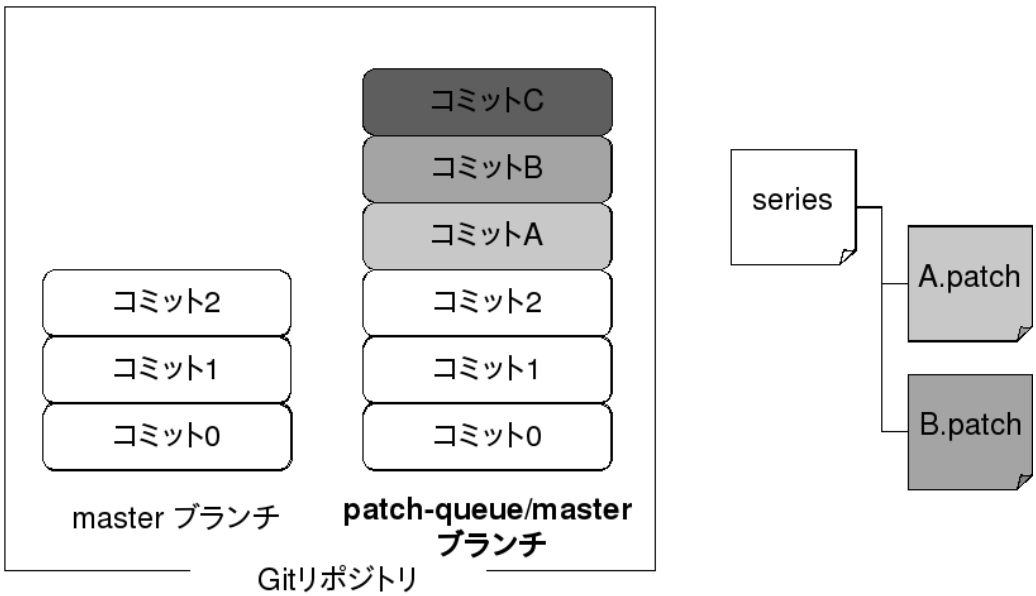
11.8 gbp pq import

1. HEAD を patch-queue/master ブランチとしてチェックアウト
2. debian/patches/series にあるパッチをコミット



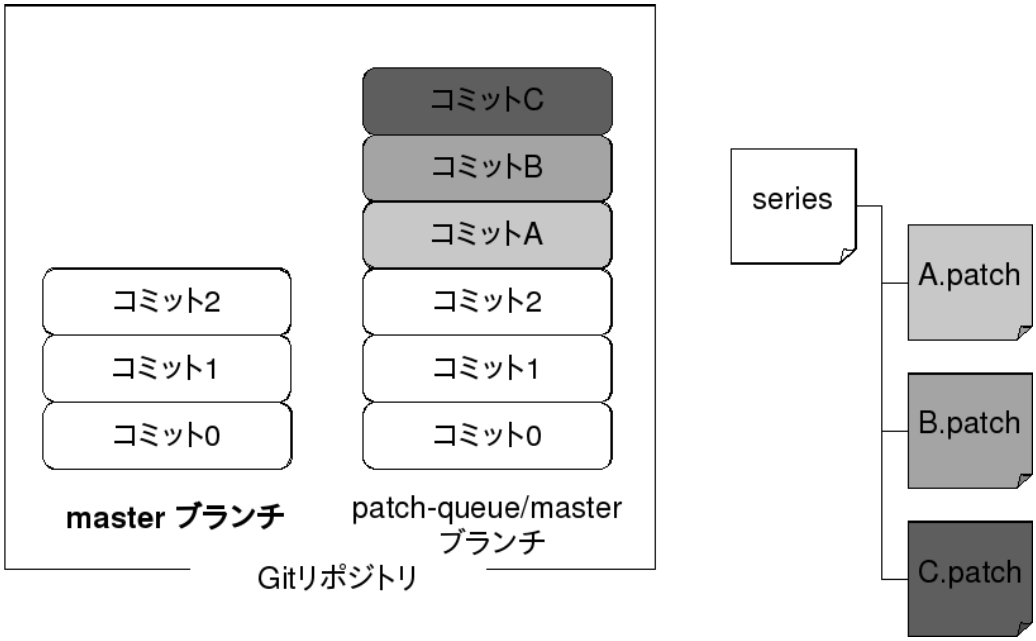
11.9 修正 & git commit

1. Upstream ソースコードの修正
2. git commit



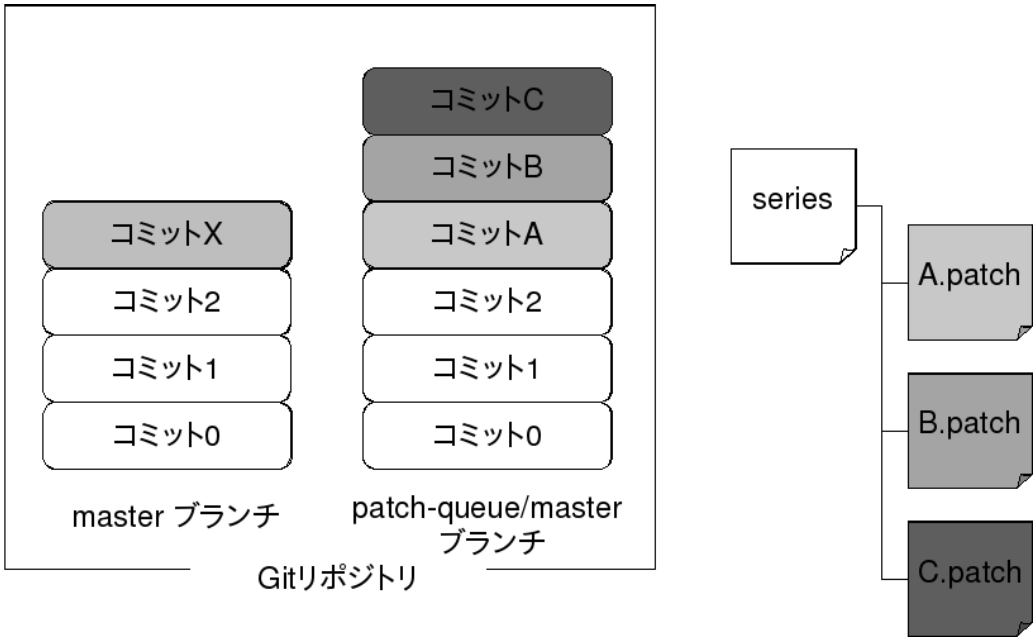
11.10 gbp pq export

1. patch-queue/master と master ブランチの差分をパッチとして debian/patches に出力
2. debian/patches/series を更新
3. master ブランチをチェックアウト



11.11 git commit

1. パッチ更新をリポジトリにコミット



11.12 まとめ

- gbp (git buildpackage) はデファクトスタンダート
- gbp import-* でリポジトリ取り込み
  `--pristine-tar` を忘れずに
- gbp dch で debian/changelog を更新
- gbp buildpackage でパッケージビルド
- gbp pq でパッチ操作

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. Debian 8.1 がリリースされました。いつだったでしょうか？
☐ A 2015/6/6
☐ B 2015/6/13
☐ C 2015/6/20

問題 2. 2015/6/10 にて、unstable 版のソースパッケージの数はいくつになったでしょうか？
☐ A 21,000
☐ B 22,000
☐ C 23,000

問題 3. AutomaticDebugPackages の提案とは何？
☐ A 大統一 Debian の岩松さんのデバッグパッケージの件を実施する
☐ B 自動でデバッグ出来るようにする
☐ C -dbg パッケージを止め、.ddeb パッケージを作る

問題 4. 6/22 にて、backport のチームが、特定の条件を満たすパッケージをごっそり消したのは、どの backports?
☐ A squeeze-backports
☐ B wheezy-backports
☐ C jessie-backports

問題 5. 7/7 にて、sid では、特定バージョンの GCC と libstdc++ でコンパイル・動作出来るようにして欲しい旨のアナウンスが流れました。どの組み合わせ？
☐ A gcc 6/libstdc++6
☐ B gcc 4/libstdc++5
☐ C gcc 5/libstdc++6

問題 6. 7/8 にて、複数の upstream から提供されている libav*群のライブラリについて、提供元の upstream を変更するとの連絡がありました。どの upstream に変更となったのでしょうか？
☐ A FFmpeg
☐ B libav.org
☐ C VideoLAN

問題 7. DPL の Neil MacGovern が reddit に開いた「DPL だけど、何か質問ある？」というスレで、DPL にとっても凄いと思うディストリビューションとしてあげられてたものはどれ？
☐ A 当然 Debianっしょ！
☐ B ArchLinux
☐ C ubuntu

問題 8. 7/20 に dgit の新しいバージョンのものがリリースされました。どのバージョンになった？
☐ A 0.1
☐ B 0.3
☐ C 1.0

問題 9. 7/21 に Debian Installer Stretch Alpha 1 がリリースされました。変更点は以下のどれ？

☐ A UEFI ブートを搭載

☐ B ネットワーク IF が MAC アドレスになる

☐ C インストール時の UI が text モードから graphical モードになった

問題 10. 2015/10/22 の DPN のメールから定期的にながれていたいくつかのトピックが Web のみに掲示されるようになりました。どこに掲示される？

☐ A http://www.debian.or.jp/

☐ B http://www.debian.org/

☐ C http://bits.debian.org/

問題 11. 2015/11/7 にて、とあるインスタントメッセンジャー用プロトコルのサービスが全 Debian Developer で使えるようになったとのアナウンスがありました。プロトコルの名前は次のどれ？

☐ A XMPP

☐ B IRC

☐ C IP Messanger

問題 12. 2015/10/30 にて、Debian にて、2 回目の公募が行われた役まわりは次のどれ？

☐ A 2016 DPN

☐ B technical commitee

☐ C 2016 Debian JP 会長

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。

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, they do not excuse you from 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 COPYRIGHT 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 NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF 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.

ソースコードについて

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
```

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. A いくつかの脆弱性対策や、バグフィックスが行われたパッケージが取り込まれました。Debian 8(Jessie) をインストールしたばかりの人は、早速アップグレードしましょう！

2. B 遂に 22,000 を超えたそうです。バイナリパッケージの数は 45,542 との事。益々増えていくようです。

3. C 今まで、デバッグシンボルは-dbg パッケージで配布されていました。こちらの-dbg パッケージは他のデバッグとは何ら関係のないパッケージと一緒に mirror され、利用者がとても少ないが、mirror 先の資源を消費します。提案は、デバッグシンボル.ddeb というパッケージにし、mirror 先も減らす（しない？）という事を検討するものです。

4. B jessie で利用できないパッケージを、wheezy-backports からごっそり消したとのことです。backports に含まれるどのパッケージがどうなっているか？どうして欲しいか？については、freeze の期間と freeze 後のわずかな期間の間に、backport 担当から backports チームに自発的にタイムリーに相談して来て欲しいとの勧告も行われました。

5. C まずは、Debian sid では、gcc 5/libstdc++6 でコンパイル・動作出来るようにパッケージメンテナの方は修正対応をして欲しい旨のアナウンスがありました。今回、ABI ベースでも変更になったり、C++11 に対応となったりで影響が諸々発生します。また、この影響で、GFortran 側も module 14 へ移行となるので、GFortran を使っているパッケージメンテナも対応が必要とのことです。

6. A libav*というマルチメディアのデータを扱うライブラリなのですが、一旦 libav.org が提供しているものに変更となったのですが、また FFmpeg が提供しているものに戻ってきた状況です。議論のサマリは https://wiki.debian.org/Debate/libav-provider/ffmpeg

7. B 質問に、DPL が ArchLinux が凄いと答えていました。wiki の充実ぶりがとにかく素晴らしいとのこと。

8. C Debian アーカイブを git で操作できるツールの dgit が 1.0 がリリースされたとのことです。早速 debian sid に収録されています。dgit clone package 名とすると、https://git.dgit.debian.org/ で管理されているものが手元に clone されます。

9. C Strech で利用されるであろう、インストーラプログラムのα 1 がリリースされました。もちろん、Strech がリリースされたわけではないので注意。変更点は数々あり、デフォルトの CPU アーキテクチャが amd64 になったりした。詳しくは debian-devel-announce を参照

10. C 2015/10/22 の DPN のメールから、従来流れていた DPN のメールの構成が刷新されました。こちらに伴い、セキュリティについてのアナウンスと、新しい DD/DM のアナウンスは、DPN のメールには含まれず、Web ページに掲載されるのみとなりました。

11. A XMPP はオープンなインスタントメッセンジャー用プロトコルの１つ。なお、Debian 関係者の利用にあたって詳しくは、http://rtc.debian.org を参照。

12. B 2 名、Debian Developer の方で technical comittee で活躍できる方募集とのことです。現職 technical comittee の人らが、2015/12/31 で任期が切れてしまうため、それまでに候補者を挙げる必要があるとのこと。

『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 2015/06〜2015/11 まで東京エリアは第 127 回から第 133 回までおよび、関西エリアは第 99 回から第 104 回まで (関西第 99-101 回,104 回はもくもく会、LT のため収録無し)。内容は無保証、つっこみなどがあれば勉強会にて。

あんどきゅめんてっど でびあん 2015 年冬号

2015 年 12 月 31 日　　初版第 1 刷発行

東京エリア Debian 勉強会/関西 Debian 勉強会 （編集・印刷・発行）