



Grand Unified Debian



銀河系唯一のDebian専門誌

東京エリア/関西Debian勉強会



あんどきゅめんとっど でびあん 2018年冬号 2018年12月30日 初版発行

デビアン勉強会

目次

1	Introduction	2
2	Debian Update (OSC 2018 Tokyo/fall)	3
3	Debconf18 参加レポート	10
4	Debian で自宅にリモート接続 / OpenVPN 編	15
5	Debian で Atom、Debian で Visual Studio Code	18
6	debian における nginx の設定例	22
7	Rethinking of debian/watch - debian/watch を再考する	30

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 をがりがりとす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 界限で起こった出来事、などなど) について話し合う会です。

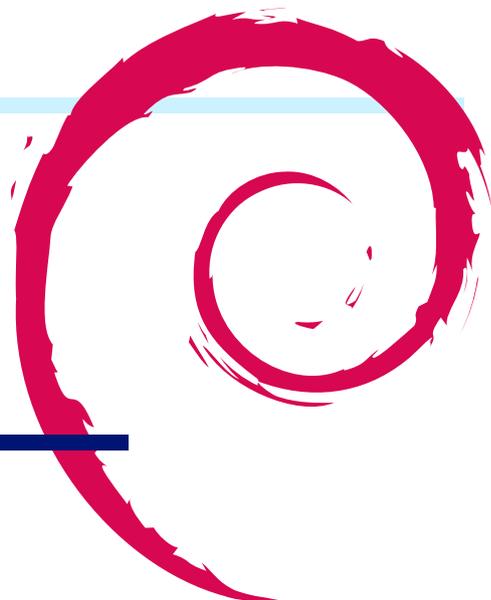
目的として次の三つを考えています。

- 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 (OSC 2018 Tokyo/fall)

杉本 典充



2.1 目次

- Debian について
- Debian JP Project と Debian 勉強会
- Debian 9 (Stretch) の紹介
- Debian Updates
 - Timeline
 - DebConf18
 - DebConf19 (予告)
 - Release Schedule
- 日本語による Debian の情報

2.2 Debian について

2.2.1 Debian とは

Debian とは

- フリー/オープンなユニバーサルオペレーティングシステムを作成しようとするボランティアベースのプロジェクト
- 世界規模で開発が行われており、63ヶ国、約 1000 名の Debian 公式開発者が開発を行っている
- パッケージメンテナや翻訳などの貢献者も入れるともっと多くの開発者が参加していることになる

表 1 Linux ディストリビューションの開発体制

ディストリ	企業	ボランティア
RHEL	RedHat	なし
CentOS	RedHat	あり
Ubuntu	Canonical	あり
Debian	なし	あり

利用用途

- Linux サーバ
 - 例：web サーバのシェア：<https://w3techs.com/technologies/details/os-linux/all/all>
- 組込デバイスのベース OS（多くの CPU で動作する）
- デスクトップ PC、ノート PC などの普段利用するコンピュータの OS

「Debian」ベースな派生 OS の源流

- Ubuntu や Raspbian といったディストリビューションのベースは Debian
- 派生先のディストリビューションと相互に情報交換をして開発している

最新の安定版

- 2018 年 10 月の時点で、最新版は **Debian 9.5** (コードネーム: Stretch)*¹
 - パッケージ数は約 **51000** を提供
 - 公式にサポートする CPU アーキテクチャは **10**
- 約 **2 年**毎にリリース
- 次のリリース Debian 10 (コードネーム: Buster) は 2019 年中頃にリリースすると思われる
- コードネームはトイ・ストーリーのキャラクターを採用

2.2.2 Debian Project の特徴

- Debian 社会契約
 - Debian 開発者たちが目指すフリーソフトウェアコミュニティの在り方
- Debian フリーソフトウェアガイドライン (DFSG)
 - Debian 社会契約の一部
 - Debian が考えるフリーソフトウェアの定義
 - オープンソースの定義のひな形にもなっている
- Debian Policy
 - <https://www.debian.org/doc/debian-policy/>
 - Debian パッケージの区分、内容やルール、ファイル配置の方針を定義

2.2.3 Debian で覚えておいてほしいこと

- Debian はフリー/オープンなオペレーティングシステム (OS) を作成しようとするボランティアベースのプロジェクト
- 自分たちの考えるフリーという言葉に関する定義、開発目的、パッケージングポリシーを厳格に決めている
- 世界中に 1000 人以上の開発者がおり、他のディストリビューションのベースとして採用されている
- 約 2 年毎にリリースが行われ、多くのパッケージとアーキテクチャをサポートしている
- 上記のような特徴から様々なところで利用されている Linux ディストリビューション

2.3 Debian JP Project と Debian 勉強会

2.3.1 Debian JP Project とは

- 日本で Debian を普及させることを目的とした任意団体
- 活動内容
 - Debian の日本語による情報発信

*¹ 追記：2018-11-10 に Debian 9.6 をリリースしている。

- ユーザとの情報交換
- Debian 開発者、パッケージメンテナの育成など

2.3.2 Debian 勉強会とは

成り立ち

- 2005 年 1 月開始
- Debian Developer 上川さん発起人
- 東京と関西で月に一回コンスタントに開催している Debian 開発者、ユーザによる勉強会
 - 東京エリア Debian 勉強会 (<https://tokyodebian-team.pages.debian.net/>)
 - 関西 Debian 勉強会 (<https://wiki.debian.org/KansaiDebianMeeting>)

Debian 勉強会で解決したい問題

- 問題
 - ML と IRC で情報交換していた
 - face-to-face であう場所がない
 - まとまったドキュメントが出てこない
- Debian 勉強会の提案
 - 定期的に集まる
 - 資料を作成する (GPL で！)

<https://salsa.debian.org/tokyodebian-team/monthly-report>

Debian 勉強会の実際

- Debian 界限やパッケージング関連の話題など専門の人に話を聞く
- Debian で気になった事柄を調べてレポートする
- 前回の内容 (東京 2018 年 9 月・第 166 回) :
 - 場所: 朝日ネットさん
 - 「Rethinking of debian/watch rule」 (@kenhys)
 - buster に向けた日本語環境のタスク整理

https://tokyodebian-team.pages.debian.net/2018-09_tokyodebian_langja_bof.txt
- 各地のイベントで Debian 普及活動
 - OSC2018 北海道、OSC2018 京都、OSC2018 東京
 - 関西オープンフォーラム
 - Debian/Ubuntu ユーザミートアップ in 札幌

2.4 Debian 9 (Stretch) の紹介

2.4.1 Debian 9 (Stretch)

最新の安定バージョン

- Debian 9.0 は、2017-06-17 にリリース
- このリリースは、Debian Project の創始者 Ian Murdock 氏に捧げるリリースになっている
- 2018 年 10 月時点の最新のマイナーリリースは Debian 9.5 であり、2018-07-14 にリリース*2

*2 追記 : 2018-11-10 に Debian 9.6 をリリースしている。

サポートアーキテクチャ

- Debian GNU/Linux でサポートされる CPU アーキテクチャ*3
 - amd64, i386 (i686 以降)
 - armel, armhf, arm64
 - mips, mipsel, mips64el
 - ppc64el, s390x
- サポートされていない移植版として、昔の CPU や RISC-V、FreeBSD や Hurd のカーネルを利用したものが
ある

ソフトウェア

- Linux カーネルは 4.9
- ツールチェイン (GCC 6.3.0, binutils 2.28, glibc 2.24), LLVM 3.7.1, 3.8.1, 3.9.1
- Perl 5.24.1, Python 2.7.13/3.5.3, Ruby 2.3.3, PHP 7.0.19, Go 1.7.4, OpenJDK 8
- GNOME 3.22, KDE 5.8, Xfce 4.12.3, lxde 0.99.0, lxqt 0.11.1
- MariaDB 10.1.23, PostgreSQL 9.6.3, sqlite 3.15
- OpenSSL 1.1.0, GnuPG 2.1.18/1.4.21
- クロスコンパイラがデフォルトでサポート
- etc..

困ったときの報告先

- 何かおかしい動作や不具合を見つけた場合はバグレポートをお願いします
- バグレポートの仕方
 - <https://www.debian.org/Bugs/Reporting.ja.html>

2.5 Debian Updates

2.5.1 Timeline

- 2017/12/09: Updated Debian 8.10 released
- 2017/12/09: Updated Debian 9.3 released
- 2018/03/10: Updated Debian 9.4 released
- 2018/06/23: Updated Debian 8.11 released
- 2018/07/14: Updated Debian 9.5 released

- 2018/04/18: Debian Project Leader Election 2018 Results
 - 2017 年の Debian Project Leader(DPL) の Chris Lamb さんの信任投票となり、信任された
 - Chris さんの所信表明は、次の URL で参照できる
<https://www.debian.org/vote/2018/platforms/lamby>
- お得な情報 : 「Bits from the DPL」
 - 月に 1 回”debian-devel-announce@lists.debian.org”の ML に Chris さんがプロジェクトの進捗を報告している
 - 時間のない方でもこれを読んでおけば Debian Project の大まかな動きがわかる

- 2018/05/01: Call for testing openssl TLS 1.3 support
 - OpenSSL 1.1.1 から TLS 1.3 を利用できるように呼び掛けるアナウンスを実施

*3 <https://www.debian.org/releases/stable/amd64/release-notes/ch-whats-new.ja.html>

- 2018/5/31: Alioth shutdown
 - 2003 年から開発者向けに CMS、リポジトリサーバとして稼働していた alioth.debian.org が提供終了
 - 後継の salsa.debian.org へ機能を引き継いだ
 - <https://salsa.debian.org/public> で gitlab が稼働し、Kubernetes (k8s) と組み合わせた CI/CD も実行可能
- 2018/06/05: Data Protection team delegation
 - EU で施行された GDPR に対応するためのチームとメンバーを紹介
 - アドバイスが欲しい場合は Debian Data Protection team へ問い合わせることができる

2.5.2 DebConf18

開催日程

- <https://debconf18.debconf.org/>
- 台湾 新竹市 (Hsinchu) の國立交通大學で開催
- アジア地域での開催は初めて
- DebCamp 2018/7/21 - 7/27
- OpenDay 2018/7/28
- DebConf 2018/7/29 - 8/5



図 1 DebConf18 ロゴ

イベント参加者

- 各国にいる Debian 開発者が集まった
- 参加者数は、300 名を超える人たちが集まった



図 2 DebConf18 参加者の集合写真

セミナー

- セミナー発表はビデオを公開中 <https://debconf18.debconf.org/schedule/>

- 日本から行った人たちがまとめた参加セミナーの情報
 - https://tokyodebian-team.pages.debian.net/2018-08_tokyodebian_debconf18_bof.txt
- セミナー発表のタイトルの一部抜粋
 - Bits from the DPL
 - Debian Science BoF
 - Ignoring Negativity
 - News from the APT team
 - Debian Brasil and DebConf19
 - Debian Diversity Team discussion session
 - Report from the Debian EFI team about the support of Secure Boot on Debian
 - Making Debian more user friendly by changing the web start page (www.debian.org)
 - GDPR in Debian
 - Debian Go Packaging Team BoF
 - The Free Software Foundation, Debian, and the free software movement

イベント中の生活

- Code of Conduct を読み、節度ある生活をする^{*4}
- ベジタリアン、ハラールの方を考慮した食事を用意
- 宿泊は学生寮に滞在可能
- wiki を読んで行動する <https://wiki.debian.org/wiki/DebConf18>
- お知らせはアナウンスメールを読む <https://lists.debian.org/debconf-announce/>
- セブンイレブン、ファミリーマートが大学構内にあり便利
- 大学内のスポーツ施設を有料で利用可能

ハックラボ

- 今回は Noisy Hacklab、Quiet Hacklab の 2 部屋を用意
- インターネット回線、電源を完備
- コーヒーやビールを飲みながらハックし、議論し、親交を深める

DayTrip

- DebConf 中に 1 日、参加者たちが一緒に出かける企画
- <https://wiki.debian.org/wiki/DebConf18/DayTrip>
- 工場見学ツアー、文化施設見学ツアー、お茶会ツアー、台北観光ツアー、自転車でツーリング、川遊び

チーズ&ワインパーティ

- 自分の国のワインとチーズを持ち寄るホームパーティ
- 異文化理解や新たな発見をするための交流の場
- <https://wiki.debian.org/wiki/DebConf18/CheeseWineBoF>

カンファレンスディナー

- 現地の料理を楽しみながら懇親するディナー
- スポンサーの方々から挨拶もあります

^{*4} <https://debconf.org/codeofconduct.shtml>, https://www.debian.org/code_of_conduct

2.5.3 DebConf19 予告

- ブラジル南部のクリチバ市にある Federal University of Technology (UTFPR) で開催
- <https://wiki.debian.org/DebConf/19>



図 3 DebConf19 ロゴ

2.5.4 Release Schedule

- 2018/09/30: Debian 10 (Buster) Release Schedule
 - 2019-01-12 : Transition freeze
 - 2019-02-12 : Soft freeze
 - 2019-03-12 : Full freeze
- Debian 10 は 2019 年中頃のリリースになるのではないかと推測

2.6 日本語による Debian の情報

- Debian JP Project
 - <https://www.debian.or.jp>
- 東京エリア Debian 勉強会
 - <https://tokyodebian-team.pages.debian.net/>
- 関西 Debian 勉強会
 - <https://wiki.debian.org/KansaiDebianMeeting>
- Twitter
 - @debian_jp
- 雑誌 Software Design 技術評論社発行
 - 「Debian Hot Topics」、GPG 関連の記事など（隔月連載）

3 Debconf18 参加リポート

東京エリア Debian 勉強会 Debconf18 参加者



3.0.1 DebConf18

本号の前述のページでも紹介していますが、下記のように Debconf18 が開催されました。

- <https://debconf18.debconf.org/>
- 台湾 新竹市 (Hsinchu) の國立交通大學で開催
- アジア地域での開催は初めて
- DebCamp 2018/7/21 - 7/27
- OpenDay 2018/7/28
- DebConf 2018/7/29 - 8/5

以下 Debconf18 参加者の東京 Debian 勉強会での振り返り Debconf18 参加リポートです。興味あるセッションがあればスケジュールページにビデオ <https://debconf18.debconf.org/schedule/> が公開されていますので、見返してみたいかがでしょうか。

Saturday (28 jul) - Open Day

- パネルディスカッション (Panel Discussion: Story of Debian contributors around the world) 参加 (やまね)
<https://debconf18.debconf.org/talks/143-panel-discussion-story-of-debian-contributors-around-the-world/>

Sunday (29 jul)

- Software transparency: package security beyond signatures and reproducible builds

- パッケージにマルウェアが混入しないようにする、入った場合に検出できるようにする試み
- Certificate Transparency のソフトウェア版
- 標的型バックドアの阻止が目的

- Bits from the DPL

- DPL からのこの 1 年間の報告

- Debian Science BoF

- コンピュータサイエンスで利用するパッケージをメンテナンスしている Debian のチームが、興味のある人たちへの導入ガイダンスを行う
- 台湾、中国の学生さんの参加が多かった→理系学生に Debian に興味を持ってもらう方策としてこの辺りを進

めて、日本でも理化学系ソフトウェアのパッケージングを進めたりするのは良いかも知れない

- Continuous Key-Signing Party introduction

- キーサインの説明、hash チェックを行った。
- キーサインの作業自体は個別にやってほしい、というスタンスのやり方だった

Monday (30 jul)

- Learn how to triage bugs
 - Debian BTS の検索の仕方と bug report メール の書き方の話
 - ワークショップで 2 時間枠だったが、通常の発表方式で 45 分枠の方がよかった気がする
 - Tools inside Blends framework
- 発表者は Debian Med の人 この人が現状大部分コミットしてる模様
- 課題は 1 人だけで作業しているパッケージが多い点らしい

- Rust packaging BoF

- 変換ツール debcargo を作ったらしいがその依存関係が Debian にまだ入れられてない
- cargo はパッケージングはされてるものの 依存関係を別パッケージにできていないのが一因

- Ignoring Negativity

- ネガティブな発言を受けたときにどのように受け流していくかの心得
- いい内容だと思います

- git-debrebase - new tool for managing Debian packaging in git

- git で管理されているアップストリーム特に /debian ディレクトリがあるようなものを修正するのに便利そう

- Challenges of GNU Ring - a free and universal distributed communication platform

- GNU の分散メッセージングアプリ
- アカウント名の共有にブロックチェーンを使っているらしい
- 中国人はやはり検閲事情に詳しい

- di-netboot-assistant: Install Debian in your LAN

- News from the APT team

- dpkg チームから: /var/lib/dpkg/info/ の .list は廃止されて .mtree になるらしい
- パッケージの圧縮フォーマットに zstd を追加する検討について話があったが、結論は出てなかった。

- OpenRC: A Featureful and Hackable Init System

- OpenRC をアピール
- オフトークで sysvinit 語り

Tuesday (31 jul) - Debian Brasil and DebConf19

- ポルトガル語で実施されたセッション
- ブラジルで開催予定の DebConf19 の準備の打ち合わせ
- 台湾の会場とブラジル現地をビデオチャットでつないで打ち合わせする会議
- 食事、会場、宿泊場、お金、スポンサーなどイベント開催の悩みは多い
- ビデオをどうやればうまくできるかは経験値が足りない

- SPI BoF

- 特定の人が SPoF にならないようにしたい という話も どのチームでもそうなんだなあという感想

- Debian Diversity Team discussion session

- あらゆる人を受け入れる態勢にしよう という話
- 言語が障壁になっている例を noga jun さんと身をもって示すことができた

- Report from the Debian EFI team about the support of Secure Boot on Debian

- Microsoft が署名した grub shim (grub をチェーンロードするだけの EFI ブートローダー) を使う
- Secure Boot カーネルパッケージで起動すると Debian が署名したカーネルモジュールだけが使える
- DKMS 使うには Secure Boot を off にしないと使えない

- Making Debian more user friendly by changing the web start page (www.debian.org)

- Mr.FAI による debian の web サイトのトップページのみを、まずは他のディストリビューションのよいところを取り込んでサンプルを作ってみた。そのページのレビューや意見を聞きたい、というセッション

- Status of Japanese (and CJK) typesetting with TeX in Debian

- 中国語圏から TexLive に何も意見が来ない... という話だったが やっぱり来なかった
Thursday (02 aug) – Free software wins and losses in numerical methods education for engineers
- フリーでない MATLAB 捨てて Jupyter Notebook で授業の事前課題や宿題を出す話

- Reproducible Buster and beyond

- 人件費のための寄附が足りないらしく最近停滞してるらしい...

- Segregated Dynamic Linking (keeping the peace between incompatible dependencies)

- リンカの一部機能を切り出してライブラリにした そのため glibc の次期バージョンにパッチが取り込まれないと使えない 内部的に使っていた機能を分割することになって deadlock してしまう

- GDPR in Debian

- GDPR の debian のチームがあることの紹介

- Lightning Talk

- leeny squeeze で debconf18 の web サイトが見れない
- ビデオに字幕つけてほしい
- arm port の歴史

- Open Hardware

- 中国のしんせんの話

- カンファレンスディナーに参加。食べきれないくらいの充実っぷり。

Friday (03 aug)

- Short talk の発表をした (林)

- <https://debconf18.debian.org/talks/95-rethinking-of-debianwatch-rule/> - <https://debconf18.debian.org/talks/66-delta-upgrades-revisited/> 新しい方式だと容量削減もできとオトクらしい

- My crush on GNU Guix

- GNU Guix の紹介 - Nix の GNU 版 guile で実装されている
- Nix と同様 セキュリティアップデートは困難
- "guix import ..." で言語モジュールからのパッケージ変換がある程度できるらしい

- The Free Software Foundation, Debian, and the free software movement

- non-free からは逃れられないのだろうか
- AI 自動生成物のライセンスも課題らしい

- Pocket Debian, in my high school

- 教育向けにコンピュータを学習に利用する場合の取り組み
- 教師は準備に時間をかけられないし、子どもたちはうまく操作できない。難しい
- 元々KNONPIX を使っていた
- OS イメージの開発をやるプロジェクトがある

- Rethinking of debian/watch

- watch ファイルの現在の使われ方、オプションの統計調査。
- 簡単な watch ファイルの書き方をするための提案や generator を作ってみた話

- Delta upgrades revisited

- パッケージの更新分を差分 (delta) だけ転送できるようにパッケージの構造を変更を加える試み

- DNS Transport Security in Debian

- systemd-resolve unstable にある、デフォルト無効
- <https://packages.debian.org/stretch/dnscrypt-proxy>
- dnsprivacy.org <https://dnsprivacy.org/wiki/>

- Gnuk+GnuPG Explained for Debian Developers and Users

- 最新の楕円暗号 ed25519 に対応している唯一のトークン
- チップはむき出しであるので、ハードウェア的に強固かと言われるとちょっと…各自でなんとかしましょう、という感じ。
- 持っていったトークンはすべて売り切れた

- Debian Go Packaging Team BoF

- vgo (セマンティックバージョンニング対応 go) が次期バージョンで導入されたらパッケージングしやすくなるかも
- 共有ライブラリには当面ならなそう

Saturday (04 aug)

- DebConf20: In... your city?

- ローカルチームで張り付ける人が4人は必要
- vegan、ベジタリアン向け対応が必須

- Build tools for applying Debian to Embedded Systems

- mini Debian Conference Japan から少し進展 他の同様なプロジェクトと協調しようとしているところらしい
- Deby (meta-debian layer) の話が少し

- Hanjpn-IM Project : Building an experimental Input Method for typing Japanese using Korean Hangul

- ハングル入力かな出力と IME の話

- イベント開催の運営で気づいたこと (スタッフ、設備など)

- ビールサーバー重要
- コーヒーサーバー重要
- 宿泊場所がカンファレンス会場から近くてよかった。
- グッズが良かった。コンパクトなカバンに竹のコップ。金属製箸 & スプーン。
- T シャツとタオルもあった。
- Welcome Guide PDF がよかった。地図、困ったときの中国語文例など
- 政府関係と協力してやるのがイベント開催には重要であることと、参加者情報の提供の話は先に詰めておく必要があったこと
- カンファレンスブース (FrontDesk) 運営はとてもゆるかった。
- hacklab が程よい広さでよかった。
- セミナーの時間枠は 45 分枠はちょうどよかった。

- 他の開発者との交流 / 異文化交流

- noisy hacklab / quiet hacklab に分けたのは良かった。noisy で皆ビール飲みながら色々話げできた
- keysign が会話のきっかけになるので良い
- easybyke 便利
- easycard 便利

- 困ったこと

- 英語力が低く会話がうまくできなかった。英語勉強が必要。
- 事前の移動手段を調べる。MRT と HSR はいいけど、市バスはわかりにくい。インターネットで見えるバスロケーションシステムは役立った、が英語で書いてない
- タクシーとかストリットフードとかために英語や日本語はきけなかった。台湾人と仲良くしたから助かった!
- KSP で hotmail を smarthost にしていると、メールの中身がおかしくなるようで、受け付けてもらうのにトラブルシューティングしてもらった- exchange の問題らしい?

4 Debian で自宅にリモート接続 / OpenVPN 編

杉本 典充



4.1 アジェンダ

- 自己紹介
- VPN について
- OpenVPN の紹介
- OpenVPN の設定 (サーバ、クライアント)
- おわりに
- 参考資料

4.2 自己紹介

- Norimitsu Sugimoto (杉本 典充)
- dictoss@live.jp
- Twitter: @dictoss
- 大学生のときから Debian を使っています
- 仕事はソフトウェア開発者をやってます

4.3 VPN について

- IPsec 系
 - L2TP/IPsec
- SSL-VPN 系
 - SSTP
 - SoftEther
 - OpenVPN
- 利用を推奨しない過去の技術
 - PPTP (セキュリティ強度が低く、突破済み)

4.4 OpenVPN の紹介

- SSL-VPN を使った VPN 接続を行うソフトウェア
- <https://openvpn.net/>
- ライセンスは GPLv2
- TLS 1.0~TLS 1.2 による暗号通信をサポート
- サーバ側の機能とクライアント側の機能の両方をもつ
- デフォルトのポート番号は 1194/UDP で、TCP も利用可能

4.5 Debian における OpenVPN

- Debian 8
 - openvpn-2.3.4、openvpn-2.4.0 (jessie-backports)
- Debian 9
 - openvpn-2.4.0、openvpn-2.4.4 (stretch-backports)
- openvpn の 2.3 系と 2.4 系の違い
 - 2.3.3 から TLS 1.2 が利用可能。ただし使える暗号は TLS 1.0 由来のものに限られる (AES-CBC、RSA、DHE)
 - 2.4 から AES-GCM や、ECDHE、ECDSA など TLS-1.2 で利用可能な高度な暗号をサポート
 - サーバとクライアントで 2.3 系と 2.4 系が混在する場合、古いバージョンがサポートする設定に合わせる必要がある

4.6 OpenVPN の設定の解説

4.6.1 OpenVPN の設定 (サーバ)

- "apt-get install openvpn" でインストール
- 証明書や鍵の生成 (easy-rsa 2)^{*5}
- サーバ側の設定のサンプル^{*6}
- 設定ファイルは "/etc/openvpn/*.conf" を使用し、複数の VPN 設定を同時実行するマルチインスタンスが可能
 - 例) 443/TCP と 1194/UDP の両方を listen する

4.6.2 OpenVPN の設定 (サーバ側を一部抜粋)

```
tls-server
tls-version-min 1.2
port 443
proto tcp
dev tun0
persist-key
persist-tun
push ``dhcp-option DNS 192.168.2.1''
push ``route 192.168.2.0 255.255.255.0''
client-to-client
client-config-dir /etc/openvpn/ccd
tls-auth /etc/openvpn/easy-rsa/2.0/keys/ta.key 0
(snip)
```

^{*5} <https://openvpn.net/index.php/open-source/documentation/miscellaneous/77-rsa-key-management.html>

^{*6} <https://github.com/OpenVPN/openvpn/blob/master/sample/sample-config-files/server.conf>

4.6.3 OpenVPN の設定 (クライアント)

- "apt-get install openvpn" でインストール
- クライアント側の設定のサンプル*7
- 設定ファイルは、"/etc/openvpn/client.conf" とすること
- systemctl start openvpn で起動

4.6.4 OpenVPN の設定 (クライアント側を一部抜粋)

```
client
tls-client
tls-version-min 1.2
dev tun
proto tcp
remote yourserver-fqdn 443
nobind
ca /etc/openvpn/ca.crt
cert /etc/openvpn/myclient1.crt
key /etc/openvpn/myclient1.key
tls-auth /etc/openvpn/ta.key 1
log /var/log/openvpn.log
log-append /var/log/openvpn.log
```

4.6.5 おわりに

- Debian パッケージで提供している OpenVPN を説明しました
- サーバ側とクライアント側の設定を説明しました
- 速度重視の場合は、UDP を使ってください
- つながらない、切れやすい場合は、openvpn の設定ファイルの MTU 値を少し小さくしてみてください
- 北海道の広大な地でも、VPN 接続できればノート PC からどこでも自宅のネットワークへ接続できます

4.6.6 参考情報

- <https://wiki.debian.org/OpenVPN>
- <https://openvpn.net/index.php/open-source/documentation.html>

*7 <https://github.com/OpenVPN/openvpn/blob/master/sample/sample-config-files/client.conf>

5 Debian で Atom、Debian で Visual Studio Code

吉野 与志仁



5.1 はじめに

皆さんはどんなエディタを使っていますか。最近では Atom^{*8} や Visual Studio Code^{*9} といったものを使っている方もいるでしょう。このようなエディタを Debian で “フリーに” 使えるようになるための道のりをご紹介します。

5.2 “公式サイト” からのインストール

“普通に” インストールするには、エディタの “公式サイト” からダウンロードできる deb ファイルを持ってくればできます^{*10}。ただ、これは後述の “フリー” ではありません。

5.3 Snappy

Snappy は、Canonical が元々は Ubuntu 向けに開発したツールです。

Debian でも

```
sudo apt install snapd
```

で使うことはできるようです。

- Atom^{*11}

```
sudo snap install --classic atom
```

ただ、この Snap パッケージだと現状日本語入力できないようです^{*12}。

- Visual Studio Code^{*13}

```
sudo snap install --classic vscode
```

日本語入力については Atom と同様とみられます。

これら Snap パッケージは、実際には “公式サイト” から deb ファイルをダウンロードしてきて展開したものを固めているだけのようです。なので中身は “公式サイト” と同じと言えます。

^{*8} <https://atom.io/>

^{*9} <https://code.visualstudio.com/>

^{*10} 最近のソフトウェアは deb ファイルを配布していることがかなり多くなりましたね。

^{*11} <https://blog.ubuntu.com/2017/05/11/atom-is-now-available-as-a-snap-for-ubuntu>

^{*12} <http://gihyo.jp/admin/serial/01/ubuntu-recipe/0515?page=2>

^{*13} <https://blog.ubuntu.com/2017/05/19/visual-studio-code-is-now-available-as-a-snap-on-ubuntu>

5.4 APT

APT では、

- `sudo apt install atom`
- `sudo apt install vscode`

とは今のところできません*¹⁴。できるようになってほしいですね！

5.5 “バグ” 報告

Debian はすべてボランティアです。

- できるようにしたい (ITP)
- できるようになってほしい (RFP)

という要望は Debian に対するバグ報告という形で登録することができます。ただ、登録はされますが、ボランティアなのでやりたい人がやるかやってもらうかしがありません。

実際登録されています。

- <https://bugs.debian.org/747824>
ITP: `atom -- hackable editor`
- <https://bugs.debian.org/898259>
RFP: `vscode -- Microsoft Visual Studio Code`

atom は Debian に入れたい！という意思のある方がいらっしゃるようです*¹⁵。

vscode は Debian に入れてほしいという要望が登録されています。

5.6 APT line

Debian の `/etc/apt/sources.list` ファイルには使うパッケージレポジトリの場所を指定します（通称 APT line）。

Debian 公式パッケージのレポジトリは

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

といった行になりますが、ここでこのレポジトリは大きく 2 つにわかれます。

- main という部分（コンポーネント） – 厳密にはこれが Debian です。
- contrib, non-free コンポーネント – 正確には Debian ではありません。

ですので入れるなら Debian main に入ってもらいたいですね。

5.7 Debian main

Debian main には Debian フリーソフトウェアガイドライン (DFSG) に適合する“フリー”なパッケージのみが入っています*¹⁶。

*¹⁴ もちろん Microsoft のレポジトリを使えば `vscode` を APT でインストールすることはできます。しかし“フリー”ではありません。

*¹⁵ なお、記事をまとめている時点（2018 年 12 月）では標題が ITP から RFP に変更されています。これは、入れたいと言っていた方のその後の動き（パッケージ化作業等）が見られなかったため、もしやりたい人が他にいたら作業しやすいように要望に格下げされたためです。

*¹⁶ もちろん、ブランド力のある企業の“公式サイト”のバイナリを信用する、というのも一つの考え方です。一方、“フリー”な Debian 公式パッケージのバイナリを信用するのも一つの考え方です。実は各人が誰を、何を信用するのか、という問題でもあります。

https://www.debian.org/social_contract.ja.html#guidelines

DFSG では例えば次のことが求められています。

- だれでも使える
- パッケージのソースコードを見られる。そしてこれは全パッケージについて言えることなので、つまり
 - そのアプリ本体のソースコード
 - そのアプリが使っているライブラリのソースコード
 - そのアプリをビルドするためのツール（コンパイラ、ビルド支援ツール）のソースコードを見られる
- 改造もできる

そのため、main に入れるには“すべての”ソースコードを手に入れて、それを使ってビルドする必要があります。Debian 公式レポジトリ以外からダウンロードしながらビルドすることもできません^{*17}。

5.8 Atom

- GitHub - atom/atom: The hackable text editor <https://github.com/atom/atom>
ソースコードは MIT ライセンスで、DFSG に適合しています。
- <https://atom.io/>
GitHub レポジトリの releases にある自動ビルド deb と同じものをダウンロードできる模様です。

Atom は、特にライブラリとして Node.js モジュールの Electron^{*18} というものを使っています。

そのため、Debian に Electron パッケージがあるなら（Debian 内にソースコードがあるということなので）それを使えばよいのですが、まだありません。

つまり Debian に入れるためには、Electron のソースコードも手に入れて、それを使ってビルドする必要があります。

5.9 Visual Studio Code

Visual Studio Code は実際には 2 つあります。

- Visual Studio Code - Open Source <https://github.com/microsoft/vscode>
このソースコードは MIT ライセンスで、これ自体は DFSG に適合しています。
- Visual Studio Code <https://code.visualstudio.com/>
これは“Visual Studio Code - Open Source”のソースコードを Microsoft がカスタマイズしてビルドしたもので、バイナリパッケージ（deb ファイル）はダウンロードできます。しかし、この Microsoft がカスタマイズした部分のソースコードがありません。具体的には“Microsoft による利用状況データ収集”機能などの部分が気になります。なお、Microsoft の開発者は設定ファイルが違うだけと言ってはいます^{*19}。

Microsoft の出しているバイナリなのだから信用して使います、というのも一つの考え方です。そして Microsoft の意図も理解できます。ですが、実際そこで何が行われているかはわかりません^{*20}。すべてのソースコードがみんなから見えるのは大切ではないでしょうか。

なお、Atom と同様 Electron を使っているので、Debian に入れるためには Electron のソースコードも手に入れて、それを使ってビルドする必要があります。

^{*17} そのため、時折話題になったりする、ライブラリに悪意のあるコードが仕込まれるようなことがあっても、それが勝手に Debian に持ち込まれることはありません。

^{*18} <https://electronjs.org/>

^{*19} 2015 年時点 <https://github.com/Microsoft/vscode/issues/60#issuecomment-161792005>

^{*20} TypeScript 部分ならバイナリパッケージであっても読めるところはありますが

5.10 パッケージ化

Debian には、ソフトウェアを Debian パッケージの形にすることで入れることができます。

- ふつうはライブラリごとに別パッケージにします。例えば、electron パッケージを作ってからそれを使って atom パッケージや vscode パッケージを作ります。
- Node.js 関連のライブラリ・アプリの Debian パッケージ化をしたい人が集まっている Debian JavaScript Maintainers というチームがあります。<https://wiki.debian.org/Javascript>
- Electron などのパッケージ化作業の状況なども見られます^{*21}。<https://wiki.debian.org/Javascript/Nodejs/Tasks>

興味がある方はぜひ参加してみたいかでしょうか。

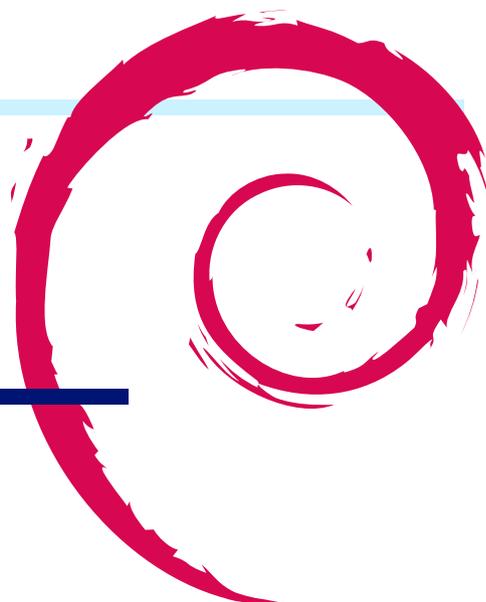
5.11 まとめ

- Debian で Atom や Visual Studio Code をアプリのサイトからダウンロードすれば動かすことはできます。
- Snappy でインストールする方法もありますが、現時点ではうまく動かないところがあるようです。
- Debian 公式レポジトリに atom パッケージや vscode パッケージはまだありません。作るには、ビルドツールや使っているライブラリすべてが Debian 公式パッケージになっている必要があります。
- Debian はボランティアなので、やりたい人がやるかやってもらうかしありません。
- “Debian main から APT でインストールできる” ということは “そのアプリのすべてのソースコードを見られる” ということと同じです。
- 公式パッケージにするために必要なパッケージを少しずつ Debian に入れている人々がいます。

^{*21} ただ、現在進められている Electron のパッケージ化作業は、実は main ではなく contrib に入れるための作業です。この作業では、“ビルド済み Electron バイナリを自動ダウンロードしてくれる” ライブラリ (electron-download) を Debian に入れようとしています。Electron は巨大なので、普通に electron ライブラリをビルドに使うと内部的には electron-download が使われています。本体部分をダウンロードしてくるわけですから main にはなり得ませんが、そもそもこの段階で使われている沢山のライブラリをパッケージ化する必要があります。これが完了したら、ようやく Electron バイナリをビルドして main に入れるための作業が始められる予定です。

6 debian における nginx の設定例

杉本 典充



6.1 はじめに

近年、nginx を web サーバとして採用する例が増えています。debian においては nginx を deb パッケージとして提供しています。

本発表では、debian における nginx のインストール、使い方、設定例を紹介します。

6.2 debian における web サーバのパッケージ

UNIX/Linux 環境で利用できる web サーバは、2018 年 11 月現在では apache、nginx の順に人気が高いです*22。

debian では apache、nginx の両方を apt でインストールできるようパッケージを提供しています。Debian 9 (stretch) では以下のパッケージ名とバージョンを提供しています。

web サーバ名	パッケージ名	Debian 9 の提供バージョン
apache	apache2	2.4.25
nginx	nginx	1.10.3

6.3 nginx

6.3.1 nginx とは

nginx (エンジンエックス) とは、2004 年にリリースした web サーバです。nginx は大量のクライアントを同時に捌くことに主眼を置き、linux-2.5.44 で導入した epoll などのいわゆるイベント駆動のシステムコールを利用したサーバアプリケーションです*23。

nginx は HTTP/HTTPS のリクエストを処理するだけでなく、HTTP/HTTPS のリバースプロキシサーバ、mail プロキシサーバ、TCP/UDP 通信のプロキシサーバとして動作するように実装しています*24。

web サイトは <https://nginx.org/> であり、<https://nginx.org/en/docs/> にドキュメントも豊富にあります。

nginx のソースコードは 2-clause BSD-like license で提供しており、debian においては main セクションのパッケージとして配布しています。

*22 <https://w3techs.com/technologies/comparison/ws-apache,ws-microsoftiis,ws-nginx>

*23 いわゆる C10K 問題の解決を図るために実装したと言われています。

*24 <https://nginx.org/en/>

6.3.2 debian における nginx のインストール方法とディレクトリ構成

nginx は apt コマンドでインストールすることができます。

```
# apt install nginx
```

Debian における”nginx”パッケージは拡張モジュールを含んでいるかで以下の 3 種類に分けています。通常の web サーバ、リバースプロキシとしての利用であれば、”nginx”パッケージの指定で十分と思います。

- nginx (または nginx-full)
 - 通常の利用ではこれを利用する
 - upstream が提供する基本的な拡張モジュールを含んでいる
- nginx-light
 - HTTP/HTTPS のリクエストを受け付けるサーバとしてのみ動作すればよい場合に利用する
 - インストールする拡張モジュールを最低限に抑えている
 - SSL/TLS、FastCGI、uWSGI の拡張モジュールも含んでいる
- nginx-extras
 - nginx-full の拡張モジュールに加えて、多くのサードパーティ製の拡張モジュールを含んでいる

6.4 nginx の設定例

6.4.1 nginx パッケージのインストール直後の設定

apt コマンドで nginx をインストールした直後は、以下のディレクトリ構造で設定ファイルが配備されています。

nginx の設定ファイルは、”/etc/nginx/nginx.conf” が基点となっており、nginx.conf が様々なファイルを include することで設定を構成しています。

```
$ tree /etc/nginx
/etc/nginx
├── conf.d
├── fastcgi.conf
├── fastcgi_params
├── koi-utf
├── koi-win
├── mime.types
├── modules-available
├── modules-enabled
│   ├── 50-mod-http-auth-pam.conf -> /usr/share/nginx/modules-available/mod-http-auth-pam.conf
│   ├── 50-mod-http-dav-ext.conf -> /usr/share/nginx/modules-available/mod-http-dav-ext.conf
│   ├── 50-mod-http-echo.conf -> /usr/share/nginx/modules-available/mod-http-echo.conf
│   ├── 50-mod-http-geoip.conf -> /usr/share/nginx/modules-available/mod-http-geoip.conf
│   ├── 50-mod-http-image-filter.conf -> /usr/share/nginx/modules-available/mod-http-image-filter.conf
│   ├── 50-mod-http-subst-filter.conf -> /usr/share/nginx/modules-available/mod-http-subst-filter.conf
│   ├── 50-mod-http-upstream-fair.conf -> /usr/share/nginx/modules-available/mod-http-upstream-fair.conf
│   ├── 50-mod-http-xslt-filter.conf -> /usr/share/nginx/modules-available/mod-http-xslt-filter.conf
│   ├── 50-mod-mail.conf -> /usr/share/nginx/modules-available/mod-mail.conf
│   └── 50-mod-stream.conf -> /usr/share/nginx/modules-available/mod-stream.conf
├── nginx.conf
├── proxy_params
├── scgi_params
├── sites-available
│   └── default
├── sites-enabled
│   └── default -> /etc/nginx/sites-available/default
├── snippets
│   ├── fastcgi-php.conf
│   ├── snakeoil.conf
│   └── uwsgi_params
├── win-utf
└── 6 directories, 24 files
```

コメントの部分を除いた nginx.conf は以下となります。標準で gzip を使ったファイル圧縮転送の機能が on になっています。

```
$ grep -v -e '\s##' -e '\s*$' /etc/nginx/nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;
events {
    worker_connections 768;
}
http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;
    gzip on;
    gzip_disable `msie6`;
    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;
}
```

debian では VirtualHost で運用しやすいようになっており、”default”という HTTP で待ち受けする設定ファイルが標準で入っています。この場合、公開する静的ファイルを”/var/www/html”配下に配置するとクライアントへ HTTP でファイルを配布できます。

```
$ grep -v -e '\s##' -e '\s*$' /etc/nginx/sites-available/default
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name _;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

6.4.2 SSL/TLS 設定

nginx は、SSL 証明書を用いた SSL/TLS の https 通信の待ち受けを行うことができます。

SSL/TLS 設定として、”/etc/nginx/sites-available/ssl” ファイルを生成し、このファイルを”/etc/nginx/sites-enabled/ssl”のシンボリックリンクを作成することで VirtualHost を追加できます。

mozilla が SSL/TLS として web サーバを動かすときに設定例を公開しているため、参考にするとよいです。

<https://mozilla.github.io/server-side-tls/ssl-config-generator/>

SSL/TLS で通信する設定を行う時のポイントは以下です。

- SSL 証明書で中間証明書を必要とする場合は、サーバ証明書と中間証明書を結合して 1 つの”crt”ファイルにする
- ”ssl_protocols TLSv1.2;”とだけ書くと、TLSv1.2 のみで通信を受け付けるサーバにできる
- 過去に判明している openssl 関連の脆弱性対応に考慮した設定とすること
- <https://www.ssllabs.com/ssltest/> などのスキャンを行い、サーバに脆弱性があるか確認する
- ssl_dhparam は、DH 1024bit がデフォルトのため Logjam 対策として 2048bit で処理するよう指定するもの*25

*25 古いクライアントプログラムは DH 1024bit までしか対応していないことがあり、悩ましい。

```
$ grep -v -e '\s#' -e '\s*$' /etc/nginx/sites-available/ssl
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    ssl on;
    ssl_certificate      /etc/ssl/private/server.crt;
    ssl_certificate_key  /etc/ssl/private/server.key;
    ssl_session_cache    shared:SSL:1m;
    ssl_session_timeout  5m;
    ssl_protocols TLSv1.2; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on; # ref: FREAK

    # Diffie-Hellman parameter for DHE ciphersuites, recommended 2048 bits
    #ssl_dhparam /path/to/dhparam.pem;

    # see https://mozilla.github.io/server-side-tls/ssl-config-generator/
    ssl_ciphers HIGH:!aNULL:!MD5;

    access_log /var/log/nginx/ssl_access.log;
    error_log /var/log/nginx/ssl_error.log;
    gzip off; # unuse gzip, ref: BREACH

    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name www.example.com;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

6.4.3 リバース proxy 設定

リバース proxy とは、クライアントと直接接続するのは nginx が担い、クライアントへ応答する実際のデータ生成はバックエンドのサーバが行うように通信を中継する機能です。

特に SSL/TLS 通信の場合、リバース proxy の設定を行うと SSL 証明書を設定するサーバは nginx のみでよく、バックエンドの web サーバは http のみで通信する設定にすることができます。

リバース proxy の設定を行うには、まず upstream の設定を行います。

upstream には、中継するバックエンドの web サーバのホストとポート番号を指定します。複数のバックエンドを設定することが可能で、ロードバランサーのように振る舞うことが可能です*26。

ロードバランスするときのアルゴリズムの指定は、least_conn と ip_hash を指定でき、何も定義しない場合は server の行に書いた weight 値を用いた重み付きラウンドロビンで振り分けます*27。

```
$ sudo vi /etc/nginx/conf.d/upstream_proxy.conf

upstream backend_app1 {
    # least_conn;
    # ip_hash;

    server 192.168.1.100:80 weight=1;
    server 192.168.1.101:80 weight=1;
}
```

proxy するときの upstream への通信はデフォルトで HTTP/1.0 になっています。この設定は「proxy_http_version 1.1;」と書くと、upstream のサーバとの通信を HTTP/1.1 にできます*28。

*26 クライアントの cookie を参照してどのバックエンドに中継するか制御する sticky の拡張モジュールは、deb パッケージには入っていません。サードパーティ製のモジュールで提供されるため nginx のコンパイルが必要です。

*27 http://nginx.org/en/docs/http/nginx_http_upstream_module.html に server 句に利用できるオプションが書いてあります

*28 http://nginx.org/en/docs/http/nginx_http_proxy_module.html#proxy_http_version

```
$ sudo vi /etc/nginx/sites-available/default

server {
    # (snip)
    location ~ ^/proxy/(.*)$ {
        proxy_pass http://backend_proxy1;
        #proxy_pass http://backend_proxy1/yourpath/;

        # if require.
        proxy_http_version 1.1;

        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        #proxy_redirect http://backend_proxy1/ http://www.example.com/;
    }
    # (snip)
}
```

nginx の設定を確認する”nginx -t”を実行し、設定ファイルに間違いがないか確認します。設定に間違いがなければ、nginx を restart します。

```
$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

$ sudo systemctl restart nginx
```

nginx のサーバの”/proxy/”へアクセスし、proxy 先の web 画面が表示されることを確認できれば、設定ができています。

6.4.4 FastCGI 設定

FastCGI とは、常駐した CGI プロセスが外部プロセス（ここでは nginx）と CGI 処理を行うためのリクエストとレスポンスのやりとりを定義したインタフェースの 1 つです。

nginx には CGI を動かす apache でいうところの mod_cgi 相当の機能はありません*29。そのため、nginx では外部のプロセスへ通信を中継することで CGI 処理に対応しており、FastCGI のインタフェースを持つ常駐プロセスと通信が可能です。

今回は FastCGI サーバとして PHP-FPM を使ってみます。PHP のバージョンは 7.0 系を利用するとし、以下コマンドでインストールします。

```
$ sudo apt install php7.0 php7.0-fpm
```

php7.0-fpm は ”/etc/php/7.0”配下に設定ファイルが配置されます。php-fpm の常駐プロセスの設定は、”/etc/php/7.0/fpm/pool.d/www.conf”です。

ここでは、www.conf を inet socket で待ち受けするように設定を変更します。

```
$ sudo vi /etc/php/7.0/fpm/pool.d/www.conf
;listen = /run/php/php7.0-fpm.sock
+listen = 9000
```

php-fpm を再起動すると、9000 ポートで listen しています。これで、/var/www/html 配下の”.php”のスク립トはすべて php-fpm として動作するようになります。

```
$ sudo systemctl restart php7.0-fpm
$ ss -npta | grep 9000
LISTEN  0      128          :::9000          :::*
```

phpinfo() のみを行う PHP スクリプトを配置しておきます。

```
$ sudo mkdir /var/www/html/myphpapp
$ sudo vi /var/www/html/myphpapp/phpinfo.php
<?php
phpinfo();
```

*29 apache で CGI プログラムを利用する場合、mod_cgi を利用して Perl プログラムを動作させることが過去には多かったと思います。

nginx の fastcgi サーバの upstream を設定します。

```
$ sudo vi /etc/nginx/conf.d/upstream_fcgi.conf
upstream backend_fcgi {
    # least_conn;
    # ip_hash;

    server 127.0.0.1:9000;
}
```

次に、nginx の VirtualHost 設定の default ファイル、必要であれば ssl ファイルに “.php” の拡張子を持つファイルへアクセスした場合は fastcgi サーバへ proxy するように設定します。

```
$ sudo vi /etc/nginx/sites-available/default
# (snip)
location ~ /\.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass backend_fcgi;
}
# (snip)

$ sudo vi /etc/nginx/sites-available/ssl
# (snip)
location ~ /\.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass backend_fcgi;
}
# (snip)
```

nginx の設定を確認する “nginx -t” を実行し、設定ファイルに間違いがないか確認します。設定に間違いがなければ、nginx を restart します。

```
$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

$ sudo systemctl restart nginx
```

nginx のサーバの “/myphpapp/phpinfo.php” へアクセスし、phpinfo の画面が表示されることを確認できれば、設定ができています。

6.4.5 WSGI 設定

WSGI (Web Server Gateway Interface) とは、常駐した CGI プロセスが外部プロセス (ここでは nginx) と CGI 処理を行うためのリクエストとレスポンスのやりとりを定義したインタフェースの 1 つです。

FastCGI と同様に、WSGI サーバへ通信を中継することで CGI 処理を行うことができます。

今回は WSGI サーバとして uwsgi を利用します。python のバージョンは Debian 9 (stretch) にパッケージが存在する python3.5、アプリケーションは django のチュートリアルを実装したものを使ってみます^{*30}。

python3、pip3、uwsgi をインストールします。git はアプリケーションのソースコードを clone するためにインストールします。

```
$ sudo apt install python3 python3-pip uwsgi uwsgi-plugin-python3
$ sudo apt install git
```

django のライブラリのインストールと django を用いたアプリケーションを配置します。

```
$ sudo pip3 install -U django==2.0.9

$ cd
$ git clone https://github.com/dictoss/django-tutorial.git
$ sudo mkdir /var/www/wsgi_apps_uwsgi
$ sudo cp -r django-tutorial/2.0/mysite /var/www/wsgi_apps_uwsgi/
$ sudo chown -fR www-data:www-data /var/www/wsgi_apps_uwsgi/mysite
$ ls /var/www/wsgi_apps_uwsgi/mysite
db.sqlite3 manage.py mysite polls
```

uwsgi は、 “/etc/uwsgi” 配下に設定ファイルを配置します。nginx の VirtualHost 設定のように “apps-available”、 “apps-enabled” ディレクトリがあります。

^{*30} <https://github.com/dictoss/django-tutorial>

”apps-available” ディレクトリに”django-tutorial.ini”という WSGI アプリケーションの設定ファイルを配置し、”apps-enabled” ディレクトリにシンボリックリンクを生成します。

```
$ sudo vi /etc/uwsgi/apps-available/django-tutorial.ini
[uwsgi]
uid = www-data
gid = www-data
plugin-dir = /usr/lib/uwsgi/plugins
plugin = python3
base = /var/www/wsgi_apps_uwsgi/mysite
chdir = /var/www/wsgi_apps_uwsgi/mysite
module = mysite.wsgi
callable = application
env =
socket = 0.0.0.0:3031
thunder-lock = true
processes = 2
threads = 32
master = True
vacuum = True
harakiri = 60
max-requests = 512
max-requests-delta = 64
post-buffering = 8192
```

```
$ cd /etc/uwsgi/apps-enabled
$ sudo ln -fs ../apps-available/django-tutorial.ini .
$ tree /etc/uwsgi
/etc/uwsgi
├── apps-available
│   ├── README
│   └── django-tutorial.ini
├── apps-enabled
│   ├── README
│   └── django-tutorial.ini -> ../apps-available/django-tutorial.ini
└── 2 directories, 4 files
```

uwsgi を再起動すると ”/etc/uwsgi/apps-enabled” 配下に設定した WSGI アプリケーションが起動します。django-tutorial.ini の設定では 3031 ポートで listen しています。

```
$ sudo systemctl restart uwsgi
$ ss -npta | grep 3031
LISTEN 0      100        *:3031          :*
```

nginx の wsgi サーバの upstream を設定します。

```
$ sudo vi /etc/nginx/conf.d/upstream_uwsgi.conf
upstream backend_uwsgi1 {
    # least_conn;
    # ip_hash;

    server 127.0.0.1:3031;
}
```

VirtualHost 設定の default ファイル、必要であれば ssl ファイルに proxy するディレクトリ名を決めて wsgi へ proxy する設定をします。

```
$ sudo vi /etc/nginx/sites-available/default
# (snip)
location ~ ^/mysite/(.*)$ {
    include uwsgi_params;

    uwsgi_param SCRIPT_NAME /mysite;
    uwsgi_param PATH_INFO /$1;

    uwsgi_pass backend_uwsgi1;
}
# (snip)
$ sudo vi /etc/nginx/sites-available/ssl
# (snip)
location ~ ^/mysite/(.*)$ {
    include uwsgi_params;

    uwsgi_param SCRIPT_NAME /mysite;
    uwsgi_param PATH_INFO /$1;

    uwsgi_pass backend_uwsgi1;
}
# (snip)
```

nginx の設定を確認する”nginx -t”を実行し、設定ファイルに間違いがないか確認します。設定に間違いがなければ、nginx を restart します。

```
$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

$ sudo systemctl restart nginx
```

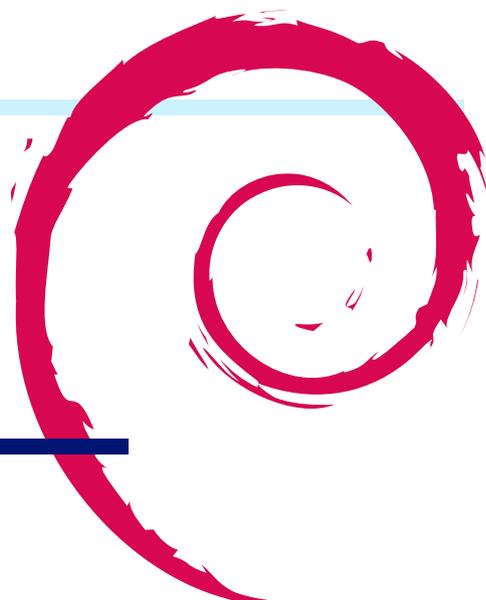
nginx のサーバの”/mysite/polls/”へアクセスし、”What’s up?”の画面が表示されることを確認できれば、設定ができています。

6.5 まとめ

nginx のよく使う設定をまとめてみました。web サーバに nginx を使おうを考えている場合は参考にしてみてください。

7 Rethinking of debian/watch - debian/watch を再考する

林健太郎 @kenhys



7.1 はじめに

Debian パッケージを更新する際は、現在リリースされているバージョンをチェックし、それよりも新しいものがあればアップストリームからリリースされているファイルをまずダウンロードするということをします。

そういったときに便利なのが `debian/watch` ファイルです。このファイルを Debian パッケージに含めておけば、`uscan` というコマンドラインツールを使って新しいバージョンのチェックから、リリースされたファイルのダウンロードまでの一連の作業を自動で行えます。最近 Debian パッケージのビルドによく使われている `git buildpackage` なども `gbp buildpackage` へ `-uscan` オプションを指定するとそのあたりを意識せずに使うことができるようになっています。

それだけを聞くととてもいいものに思えるのですが、この `debian/watch` というファイルはアップストリームがリリースしたファイルのありかを記述する方法に若干癖があります。というわけで、今回はそのあたりを改善するアイデアを検討し、その有効性を試してみた結果を示します。この内容については、2018 年 8 月に台湾で開催された `DebConf18` で発表した内容を本誌向けに若干手直しをして収録したものです。なお本記事で紹介しているデータは 2018 年 7 月末時点のものとなっています。

7.2 DebConf18 での発表資料について

`DebConf18` で発表した内容に関して、当日のセッションの概要や発表に関する動画、スライド資料を示しておきます。

- `DebConf18` の発表内容の詳細ページへのリンク
 - <https://debconf18.debconf.org/talks/95-rethinking-of-debianwatch-rule/>
- 当日の発表スライドへのリンク
 - <https://slide.rabbit-shocker.org/authors/kenhys/debconf18-rethinking-of-debian-watch/>

7.3 `debian/watch` に興味をもったきっかけについて

そもそも、`debian/watch` に興味をもったきっかけが何だったかというのと、とあるバグ報告のメールを目にしたからでした。

- `#899119: Need redirector for osdn.net`
 - <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=899119>

このバグの内容を簡単にまとめると、さわらびフォント向けの debian/watch ファイルを記述するのがとても大変なので、リダイレクターが欲しいという内容でした。このリダイレクターというのは、従来 debian/watch でがんばっていた変換ルールの処理をやらないで済むように、リリースバージョンのリストを生成する処理をサーバー側で行わせるものです。それにより debian/watch 側の記述を簡略化することができるようになります。

リダイレクターが必要なほどの debian/watch って実際どんなものだったのでしょうか。

実際の debian/watch fonts-sawarabi-mincho

```
version=4
opts="uversionmangle=s/-beta/~beta/;s/-rc/~rc/;s/-preview/~preview/, \
pagemangle=s%<osdn:file url="([\^<]*)</osdn:file>%<a href="$1">$1</a>%g, \
downloadurlmangle=s%projects/sawarabi-fonts/downloads%
    frs/redir\.php?m=iij&f=sawarabi-fonts%g;s/xz\\//xz/" \
https://osdn.net/projects/sawarabi-fonts/releases/rss \
https://osdn.net/projects/sawarabi-fonts/downloads/.*/sawarabi-mincho
@ANY_VERSION@@ARCHIVE_EXT@/ debian uupdate
```

これをみて一発で何をやっているかを読み取れたら以下の解説は読み飛ばしてもらって大丈夫です。一言で言うと、OSDN.net が提供する RSS からリリース情報を抽出しています。

この debian/watch ファイルは次の 3 つの mangle とよばれるものに分解すると理解しやすくなります。

- pagemangle
- downloadurlmangle
- uversionmangle

pagemangle はアクセスしたページの内容を置換するのに使います。さわらびフォントの場合は、<https://osdn.net/projects/sawarabi-fonts/releases/rss> という RSS の出力結果を置換するルールを正規表現で記述しています。

pagemangle による置換ルール

```
pagemangle=s%<osdn:file url="([\^<]*)</osdn:file>%<a href="$1">$1</a>%g
```

実際には RSS に含まれる `<osdn:file url="([\^<]*)</osdn:file>` という箇所を HTML の通常のリンク `$1` に変換するということをやっています。リンクになっていればあとは debian/watch でも比較的簡単に扱えるようになります。

downloadurlmangle はダウンロード先を置き換えるのに使われます。

downloadurlmangle による置換ルール

```
downloadurlmangle=s%projects/sawarabi-fonts/downloads%
    frs/redir\.php?m=iij&f=sawarabi-fonts%g;s/xz\\//xz/
```

`projects/sawarabi-fonts/downloads` を `frs/redir\.php?m=iij&f=sawarabi-fonts` に置き換えたり、`xz/` を `xz` に置き換えたりしています。

uversionmangle は拡張子にあたる部分を置き換えるのに使います。

uversionmangle による置換ルール

```
uversionmangle=s/-beta/~beta/;s/-rc/~rc/;s/-preview/~preview/
```

`-beta` を `beta` に置き換えたり、`-rc` を `rc` に置き換えたり、`-preview` を `preview` に置き換えたりしています。

なかなか複雑なことをやっていますね。なんでこんなことになっているのかについては、そもそものバグ報告 #899119 で報告者のやまねさんは次のように述べています。

They sometimes changes download way to reduce download access by preventing bot, so debian/watch file is complicated and it annoyed us. Implementing redirector in qa.debian.org would improve this situation.

ざっくりいうと bot 避けなんだろうけど、それによって debian/watch が複雑なものになっちゃうので qa.debian.org でリダイレクターを実装したほうがいいよね、という趣旨のようです。

とはいえそのためにはサーバー側の変更だけでなく、クライアント側の両方の対応が必要です。この状況を改善できないかというのが debian/watch に興味をもったきっかけでした。

7.4 debian/watch について詳しくなるには

debian/watch を改善するにしても、何から手をつけたらいいかは当時よくわかっていませんでした。そこで、まず debian/watch をよりよく知るところからはじめました。

その際に参照するとよいのが、<https://wiki.debian.org/debian/watch> です。こちらには debian/watch の概要と具体的なサイトごとのサンプルがまとまっています。サンプルとして掲載されているのは次のサイトの例です。

- Bitbucket
- GitHub
- Gitlab(Salsa)
- Google Code
- LaunchPad
- PyPI
- Sourceforge

また、これからパッケージに debian/watch を用意するにあたってよくある間違いの例も 8 つ掲載されています。

このようによく間違いやすいポイントがまとまっているのはよいのですが、そもそも debian/watch の正規表現が複雑になりすぎるといえる点がある点はそのままでし、正規表現を駆使するが故に間違いやすいというのはつらいものがあります。この状況を改善するにはどうすればいいでしょうか。

7.5 debian/watch の現状を知るために

なにかものごとを改善するにしても、まずは現状を知らないといけません。そこで、debian/watch の現状を調べてみるところからはじめました。データもなしに改善というのではあまり説得力がないからです。

都合のよいことに、データを集めるのにちょうどよい API が公開されていることがわかりました。

- <https://sources.debian.org/doc/api/>

ソースコードを検索する機能を提供している sources.debian.org が提供している API です。パッケージリストは次の API で取得することができます。

- <https://sources.debian.org/api/list>

ここでいうパッケージのリストはソースパッケージのリストです。

パッケージ名がわかれば、次にパッケージの情報を取得します。それには次の API を使います。パッケージ名のところは取得したいパッケージ名に置き換えます。

- [https://sources.debian.org/api/src/\(パッケージ名\)/](https://sources.debian.org/api/src/(パッケージ名)/)

たとえば、Groonga なら <https://sources.debian.org/api/src/groonga/> にアクセスするとパッケージの情報が取得できます。

パッケージの情報には、何をターゲットにリリースされたか、という情報が含まれています。そのため、sid のみのソースパッケージを抽出することができます。

ファイル情報は次の API で取得することができます。

- <https://sources.debian.org/api/src/groonga/latest/debian/watch/>

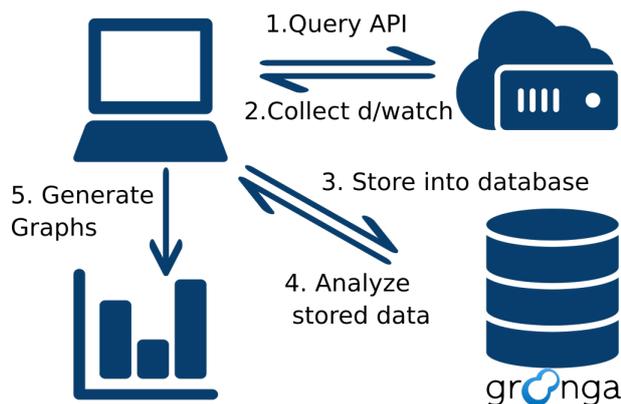
latest というキーワードを URL に指定すると、最新版のバージョンに置き換えてくれるという便利な機能があります。そのため、次の URL にアクセスすると debian/watch ファイルのメタ情報にアクセスできます。

- <https://sources.debian.org/api/src/groonga/8.0.5-1/debian/watch/>

メタ情報には raw_url というパラメータが含まれているので、実際のファイルの内容にアクセスするのに使えます。最終的に次の url にアクセスすると、ファイルの内容そのものを取得できます。

- <https://sources.debian.org/data/main/g/groonga/8.0.5-1/debian/watch>

ここまでの情報の取得方法を組み合わせれば、パッケージのリストをもとに、すべてのパッケージの debian/watch を芋づる式にかき集めることができます。手作業でやるのは大変なので、実際にはプログラムを組んで、収集してみました。API 経由であつめた情報を Groonga のデータベースに保存して分析できるようにしました。



7.6 debian/watch ファイルの傾向

前節でかきあつめた情報を分析してみたところ、いくつか興味深い結果が得られました。

debian/watch の使用状況に関する調査結果

watch file のサポート比率	パッケージ数	比率 (%)
watch file あり	23212	84
watch file なし	4448	16

sid のソースパッケージおよそ 27000 ほどのうち、およそ 84% にあたるパッケージで debian/watch に対応していました。それなりにパッケージングする場合には対応がすすんでいるようですね。

debian/watch の対応バージョンに関する調査結果

watch file のバージョン比率	パッケージ数	比率 (%)
version 3	18314	79
version 4	4439	19
version 2	396	2
version 1	63	0.3

debian/watch は複数のバージョンがあります。最新のバージョンは 4 です。もっともよく使われているのはひとつ前のバージョンであるバージョン 3 が圧倒的に多いこともわかりました。およそ 8 割がバージョン 3 で、残りのバージョンはほぼ 4 で、バージョン 2 以前は無視できる位に少ないです。

よく利用されているホスティングサイトに関する調査結果

ホスティングサイト比率	パッケージ数	比率 (%)
github.com	6432	28
metacpan.org	2946	13
sourceforge.net	1987	9
pypi.debian.net	1168	5
hackage.haskell.org	973	4
その他	9659	42

アップストリームのサイトはほぼ上位 5 サイトで過半数を占めていることがわかりました。上位 5 サイトは github.com、metacpan.org、sourceforge.net、pypi.debian.net、pакcage.haskell.org でした。SourceForge.net はいまだに根強く使われているようです。

よく利用されているオプションに関する調査結果

オプション使用率	パッケージ数	比率 (%)
filenamemangle	4134	34
dversionmangle	3921	32
uversionmangle	3695	31
pgpgsigurlmangle	1510	13
repacksuffix	1039	9
compression	489	4
repack	491	4
mode	249	2
downloadurlmangle	247	2
pgpmode	175	1
pasv	120	1
pagemangle	31	0.26
passive	30	0.25
decompress	18	0.15
comppnent	13	0.11
versionmangle	11	0.09
oversionmangle	10	0.08
date	9	0.07
dirversionmangle	5	0.04
gitmode	4	0.03
user-agent	3	0.02

ほぼ filenamemangle,dversionmangle,uversionmangle の 3 つしか使われていない状況であることがわかりました。

7.7 新しい記述方法の案

ある程度アップストリームのサイトも集中しており、また、debian/watch のオプションも使い方が限られていることから、よく使われるパターンのいくつかをサポートできる記法があれば十分ではないかと考えました。ホスティング先、プロジェクトとそのリポジトリといった情報があれば十分そうなので、たとえば、GitHub なら次のような具合です。

<https://github.com/kenhys/fcitx-implist> を例にして、新しい記述方法を比較してみましょう。

従来の debian/watch

```
version=4
opts=filenamemangle=s/.+\/v?(\\d\\S*)\\.tar\\.gz/fcitx-implist-\\$1\\.tar\\.gz/
https://github.com/kenhys/fcitx-implist/tags .*/v?(\\d\\S*)\\.tar\\.gz
```

新しい debian/watch

```
version=5
type=github.com,owner=kenhys,project=fcitx-implist
```

従来は正規表現を使うのが前提で、定義済みのマクロを使えば簡潔に書けるかも、という感じだったのを、正規表現を前提としない記述方法になっています。

7.8 新しい記述方法の検証

前項の記述方法のコンセプトの有効性を実証するために、uscan の改造を試み、同等の機能を有するかどうかを次のようにして検証しました。

- uscan を改造し、新しい記述方法をサポートする
- ソースパッケージをダウンロードする
- uscan で新リリースを検出できるように一つ前のリリースに debian/changelog を戻す
- 改造前の uscan と改造後の uscan でそれぞれ実行する
- uscan の出力を比較する

uscan の改造にあたっては、devscripts に含まれる uscan.pl で新しい記述方法をパースし、従来の変換ルールをエミュレートさせるようにすることで少ない修正で同等の機能を実現してみました。また、uscan の結果の比較には、dehs オプションを用いました。dehs オプションを指定すると、uscan は XML 形式でレポート結果を出力してくれます。その結果の差分をとることで修正版でも従来と同等の機能を実現できているかを確認できます。前項のサンプルとして掲載した debian/watch の例では、新しい記述でも同等の機能を実現できることを確認しました。

uscan への修正コミットは add-type-rule ブランチで確認できます。

<https://salsa.debian.org/kenhys-guest/devscripts/commits/add-type-rule>

ただし、最近 uscan のリファクタリングが進んだため、上記のコミットを最新の devscripts にはそのままあてられなくなっています。

7.9 まとめ

debian/watch の記述が複雑になりすぎないようにするにはどうすればいいか、新しい記述方法を検討し、その有効性を検証してみました。おおむね方向性はよさそうですが、記述の互換性の問題とその実現 (実装) 方法についてはまだまだ課題がありそうです。

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 <https://salsa.debian.org/tokyodebian-team/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 https://salsa.debian.org/tokyodebian-team/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 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.

Debian オープンユースロゴ 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 NO 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.

『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 2018/07~2018/11 まで Debian / Ubuntu ユーザーミーティング in 札幌 2018.07 の資料も収録しています。(関西は対象期間は LT 大会やディスカッション、もくもく会など実施) 内容は無保証、つつ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 2018 年冬号

2018 年 12 月 30 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)
