



Grand Unified Debian



銀河系唯一のDebian専門誌

東京エリア/関西D e b i a n勉強会



あんどきゅめんとでっど でびあん 2019 年冬号 2019 年 12 月 31 日 初版発行

全強勉強会レポート

目次

1	Introduction	2
2	Debian 10 buster リリース	3
3	Debian Updates	7
4	自作キーボード温泉に日帰り入浴してみた話	9
5	Debian GNU/kFreeBSD セットアップガイド 2019 年版	18
6	DebConf19 の気になるセミナーを調べてみる	26
7	OSC 2019 Hokkaido のイベント参加報告	29
8	東京エリア Debian 勉強会課題整理	34

1 Introduction

Debian.JP



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 境界で起こった出来事、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 10 buster リリース

杉本典充

Debian 10 buster リリースおめでとう！

Debian 10 (コードネーム : buster)

- 2019 年 7 月 6 日にリリース
- 最新版は 2019 年 11 月 16 日 にリリースした Debian 10.2

2.1 CPU アーキテクチャ

- amd64、i386
- arm64、armel、armhf
- mips64el、mipsel、mips
- ppc64el
- s390x

2.2 提供するソフトウェア

Linux kernel	4.19
GNOME	3.30
KDE Plasma	5.14
Cinnamon	3.8.8
LXDE	10
LXQt	0.14
MATE	1.20
Xfce	4.12
Chromium	78.0
Firefox ESR	68
Thunderbird	68
LibreOffice	6.1
GIMP	2.10.8
Inkscape	0.92.4
MariaDB	10.3
PostgreSQL	11
sqlite	3.27.2
	2.8.17

Emacs	26.1
Vim	8.1
OpenSSH	7.9p1
OpenSSL	1.1.1d
GnuPG	2.2.12
	1.4.23
Perl	5.28.1
Python	3.7.3
	2.7.16
Ruby	2.5.1
PHP	7.3
Go	1.11
OpenJDK	11
Rustc	1.34
GCC	8.3
binutils	31.1
glibc	2.28
LLVM	7.0.1
	6.0.1

2.3 Debian 10 の変更点

2.3.1 新機能：UEFI セキュアブート

- セキュアブートが有効な状態でもインストールと利用ができるようになった
- shim-signed、grub-efi-{amd64,ia32}-signed、buster の Linux カーネルパッケージをインストールすれば Debian 9 からアップグレードした場合でも利用可能
- DKMS が使用できないなど一部機能の制限あり

2.3.2 新機能：AppArmor

AppArmor がデフォルトで有効化

- Linux カーネルパッケージにおいて推奨 (Recommends) レベルの依存パッケージに apparmor が指定
- 多くのプロファイルは apparmor-profiles-extra をインストールすると利用可能

2.3.3 新機能：nftables

ネットワークフィルタリングの nftables への変更^{*1}

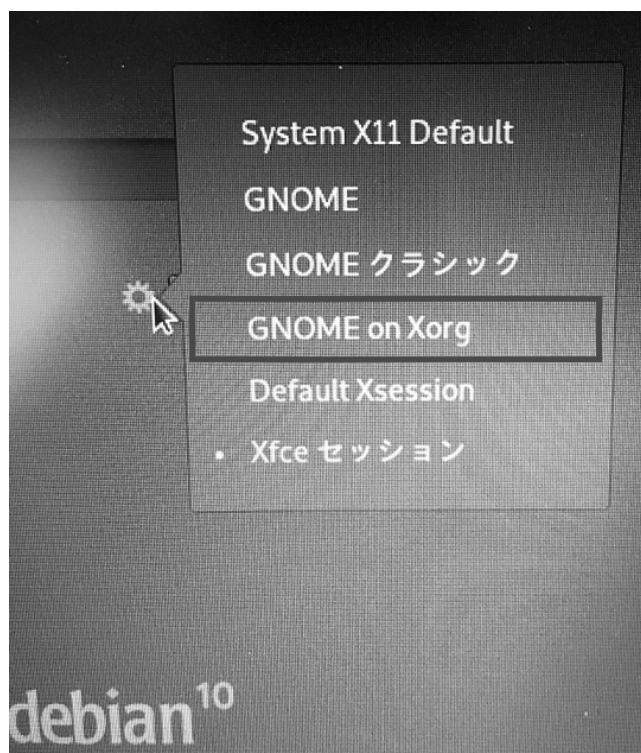
^{*1} RHEL 8 などの他のディストリビューションでも nftables への変更をしています

- iptables コマンドは nftables ベースの iptables-nft コマンドをデフォルトに変更
- 従来の x_tables ベース の iptables を使用するコマンドは iptables-legacy として提供

2.3.4 新機能：GNOME on Wayland

GNOME はデフォルトで Wayland で動作

- Xorg もデフォルトでインストールされる
- 「GNOME on Xorg」も選択可能
- 他のデスクトップ環境は Xorg で動作



2.3.5 新機能：/usr マージ

新規インストールは /usr マージした状態

- `/bin, sbin, lib` は シンボリックリンク

<code>/bin</code>	<code>/usr/bin</code>
<code>/sbin</code>	<code>/usr/sbin</code>
<code>/lib</code>	<code>/usr/lib</code>

- Debian 以外から提供されるソフトウェアを利用している場合は要注意

2.3.6 新機能：その他

- APT へのセキュリティ強化オプションの追加
- stable ポイントリリースに対する Unattended-upgrades の挙動
- Cryptsetup の on-disk LUKS2 への変更
- CUPS 2.2.10 でのドライバーレス印刷機能

- Allwinner A64 ベースのデバイスでの基本機能サポート
- など

2.3.7 動作の変更や制約

- glibc が新しくなり、Linux カーネルは 3.2 以上が必要
- Debian 9 から PostgreSQL をアップグレードした場合は DB の再インデックス化が必要
- Wayland で動作する GNOME の一部アプリケーションに不具合がある
 - 例 : synaptic、fcitx
 - Wayland で動かない場合は Xorg で動かしてみてください
- phpmyadmin、redmine、virtualbox などのパッケージは未収録
- python2.7 は非推奨
 - Debian 11 では削除する方向で作業中

2.3.8 その他の変更や制約

- OpenSSL のデフォルトが TLS 1.2 と セキュリティレベル 2
- ブラウザ、レンダリングエンジンは Firefox ESR、Chromium、webkit2gtk のみをセキュリティサポート
 - webkit や khtml エンジンを使った web ブラウザはセキュリティサポートされない
- Go 関連のパッケージは制限付きのセキュリティサポート
- evolution-ews は パッケージから削除
 - evolution で Exchange、Office365、Outlook にアクセスができなくなる

2.4 バグレポート

バグレポートをお願いします

- 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
- バグレポートの例 <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=903529>
- バグレポートの仕方（レポートは英語で送る必要あり）
 - <https://www.debian.org/Bugs/Reporting.ja.html>
- バグレポートの前にちょっと相談してみたい方は、日本語の Debian JP メーリングリストや、SNS で相談してみてください
 - <https://www.debian.or.jp/community/ml/openml.html>
 - Twitter: @debian_jp

3 Debian Updates

杉本典充

3.1 released timeline

Debian Updates

- 2019/04/27: Updated Debian 9.9 released
- 2019/07/06: Debian 10 released
- 2019/09/07: Updated Debian 10.1 released
- 2019/09/07: Updated Debian 9.10 released
- 2019/09/08: Updated Debian 9.11 released
- 2019/11/16: Updated Debian 10.2 released

3.2 Bits from the DPL

Debian Updates

Bits from the DPL

- 月に一度の Debian Project Leader である Sam さんのプロジェクトの進捗を報告
- 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
- 2018/07/02 <https://lists.debian.org/debian-devel-announce/2019/07/msg00000.html>
- 2018/08/12 <https://lists.debian.org/debian-devel-announce/2019/08/msg00001.html>
- 2019/09/18 <https://lists.debian.org/debian-devel-announce/2019/09/msg00001.html>
- 2019/10/29 <https://lists.debian.org/debian-devel-announce/2019/10/msg00002.html>

3.3 Removal of the mips

Debian Updates

2019/08/21: Removal of the mips architecture

- mips アーキテクチャ (32bit ビッグエンディアンの mips) を testing と unstable から削除
 - mips を使っている場合は mipsel または mips64el へ移行を推奨
- 次の Debian 11 では mips アーキテクチャはリリースされない
- Debian 9 および Debian 10 では mips アーキテクチャのサポートは継続
- サポート終了の理由
 - 仮想アドレス空間が 2GB までという技術的な制限
 - 開発する人や関心を持つ人の減少
- 参照: <https://lists.debian.org/debian-devel-announce/2019/08/msg00003.html>

3.4 Perl 5.30 transition

Debian Updates

2019/10/05: Perl 5.30 transition underway

- unstable において perl パッケージを 5.30 への切替を実施
- 【TIPS】
 - Debian においてプログラム言語やライブラリをバージョンアップすることを「トランジション」という
 - トランジションを行うと依存するパッケージをすべてビルドしなおす大規模な処理が行われる
- 参照 : <https://lists.debian.org/debian-devel-announce/2019/10/msg00000.html>

3.5 Python 2 removal

Debian Updates

2019/11/02: Python 2 removal in sid/bullseye: Progress and next steps

- python 2 系列は 2020 年 1 月 1 日にサポート終了を迎えるとアナウンスされている
- 次の Debian 11 では python 2.7 は削除する予定
- 進捗状況の報告、バグ報告のタグ管理方法について案内が出ている
- 参照 : <https://lists.debian.org/debian-devel-announce/2019/11/msg00000.html>

3.6 Init Systems and systemd

Debian Update

2019/11/17: General Resolution: Init Systems and systemd 2019/11/20: General Resolution: Init systems and systemd: new option

- 今後の Init System をどうするか投票する案内が出た
- https://www.debian.org/vote/2019/vote_002
- 選択肢は 5 つ
 - Init deversity is Important and NMUable
 - Systemd but we support exploring alternatives
 - Focus on systemd for init system and other facilities
 - Support non-systemd systems, without blocking progress
 - Init diversity is Required
- 参照 :
<https://lists.debian.org/debian-devel-announce/2019/11/msg00001.html>
<https://lists.debian.org/debian-devel-announce/2019/11/msg00002.html>

3.7 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」(隔月連載)
- 雑誌 シェルスクリプトマガジン USP 研究所発行

4 自作キーボード温泉に 日帰り入浴してみた話

小林 克希



4.1 はじめに

みなさん、キーボードは何をお使いでしょうか？この記事では、最近巷で流行っているという「自作キーボード」について、私が実際に作ってみたい、その後色々調べたりした内容について話そうかと思います。そもそも何故私がキーボードを自作してみようかと思ったか、ですが、要は肩凝りと腰痛がひどかったから……だったりします。とくにここ2年間くらい、仕事に腰が悲鳴をあげ、じっと席に座ってられないという状況に陥っており、フィットネスクラブにも通い、整骨院にも通い……としましたが、全然改善せず大変に悩んでおりました。

そんな中思い至ったのがキーボードでした。私は、5年ほど前に初めて Thinkpad を購入したのですが、トラックポイントがいたく気に入って、デスクトップ用のキーボードについても Thinkpad キーボードに乗り換えました。ご存知の方も多いと思うのですが、この Thinkpad キーボード、常にホームポジションから手をはなさずにいられるという点は大変にすばらしいのですが、トラックポイントって、結構操作に力が入るのですよね……。細かいカーソル操作をしている場合なんか特に。

ということで、一度他のキーボードを試してみるか……と思い至り、最初は市販の分割キーボードの Ultimate Hacking Keyboard^{*2} がトラックポイントも付けられるみたいだしよさそう、と思ったのですが、金額的にも、納期的にも、触ってもないのに手を出すのはどうかな……と尻込みし、結局自作することにしました。なにを言ってるんだこいつはと思うかもしれませんが、関西 Debian 勉強会メンバーの K さんの親指 Ctrl の布教が功を奏したとでも思ってください。

ところで、自作キーボードといっても、作成具合にレベルがあるようなのですが、一番基本的な自作方法は、キーボードキットを購入し、別途キースイッチやキーキャップも好みのものを購入していく形になるかと思います。わたしも、今のところはキットを購入して作成したので、今回はそのあたりの流れをご紹介できればと思います。「Debian 関係あるのか?」、ですって? ……………ほとんどありません…… のちほど、気持ちだけ出てきます……。

なお、この文章は関西 Debian 勉強会の 2019 年 11 月に発表した内容であるため、^{*3}スライドの方もご参照いただければと思います。

4.2 キーの数と配列

さて、実際にキーボードの自作をするという気になったはいいいものの、はたしてこういったものがいいのか困るかだと思います。そんな中でも、一番気になるのは配列ではないでしょうか。ということで、まずはどんな配列があるのかについて簡単に説明をしていこうかと思います。ところで、自分の追い求めた理想のキーボードのことを、この業界の人達は Endgame と呼ぶそうです。アベンジャーズ?

XX% キーボード

まず、キーの個数が重要な気がします。キーの個数については、フルキーの 104 キーに対して、そのキーボードのキーの数をパーセンテージであらわすのが一般的なようです。主流のものとしては、テンキー・ファンクションキーあたりを省いたものが 60%(HappyHackingKeyboard あたりがこのサイズになります。)、そこからさらにフルキーから数字の段を除いたものが 40% となるようです。

なお、逆に、100% を越えるようなキーボードもあり、たとえば左側にもテンキーがあったりするものがあるようです。どちら

^{*2} <https://ultimatehackingkeyboard.com/>

^{*3} <https://speakerdeck.com/rarewin/self-made-keyboard>

かという、ゲーム用な感じでしょうか。あと、全ての emoji が打てるキーボード^{*4} だとか 10% 切ってるようなイカれたキーボード^{*5}。もあるようです

キーの物理配列

キーの個数と同時に考えないといけないのが配列です。配列.....という QWERTY とか Dvorak とかの論理配列かと思いますが、自作キーボードでは、論理配列はファームウェアをいじれば自由自在であるため、この界限で配列といたら物理的な配列のことを指すような気がします。

ところで、物理的な配列で思い起こされるのは、市販されてるキーボードとしては、Microsoft などが出してる、いわゆる「エルゴノミクスキーボード」と呼ばれるような「G」と「H」の間あたりに隙間が空き、「G」や「H」よりも「A」や「;」が奥の方に配置されているものではないでしょうか。自作キーボード界ではもっと極端に、完全に分割している分割キーボードが流行っています(特に日本で、のようです)。なお、分割はしていますが、基本的には両者の間にシリアル通信のケーブルを這わすので、ちゃんと両者は連携できます。

また、分割しているだけではなく、キーのズレ方にもいくつか種類があります。まず、普段みかける、いわゆる「普通の」キーボードのように、行ごとに左右に少しずつずれている配置については「row-staggered」、もしくは「normal-staggered」と呼ばれます。対して、列ごと、すなわち「QAZ」は綺麗に縦に揃っているが、各列はずれている配列の事を「column-staggered」と呼びます。ぱっと見ると、「この配列は使えるんだろうか.....」と思うのですが、コンセプトとしては、指の長さに合わせて各列の位置を調整できるのが嬉しいようです。さらには、ひたすらコンパクトさを求め、完全に格子状に並べた直交配列とか Ortholinear とか呼ばれるものもあります。また、逆にキーボードのサイズが大きくなるのはお構いなしに、指の形状による配列を追い求めまくった TRON キーボードとかもあります(が、こちらのキットはみかけないかな.....)。

4.3 各パーツ

つぎに、自作キーボードにおいて必要となる各種パーツについて簡単に解説していきます。なお、2019 年 12 月現在では、実店舗のある自作キーボード専門のお店は、秋葉原にある遊舎工房さん^{*6}のみのようです。なお、関西はどうなっているかというと、日本橋のシリコンハウス^{*7}にて、最近になってキットや、Cherry と Gateron のキースイッチの販売が開始されたり、自作キーボード組み立てのワークショップが行なわれたりと、確実に風は吹いてきている!!.....ようです。あ、日本橋の千石電商^{*8}でも、Kailh と Cherry のキースイッチがあるようでした。

キースイッチ

キースイッチには色々と方式があったりしますが、実際にキーボードを自作しようとなると、流通や、各基板(PCB)の対応といった観点から

- Cherry MX
- Kailh Choc (通称、Kailh ロープロファイル)

の二種類のメカニカルキースイッチを使う事になると思います。前者は、ドイツ企業の CHERRY 社が出しているキースイッチで、特許が切れているため他のメーカーが互換品を出しており、たぶん一番種類が豊富なキースイッチかと思います。そのため、世間一般で販売されているような自作キーボードキットのほとんどの基板はこちらのスイッチに対応しているかと思います。手元にあるこのタイプのスイッチの写真を図 1 に置いてみます。互換品なので当然ですが、ぱっと見た感じではみんな同じ形をしています。打鍵具合や値段もかなりピンキリなので、色々見てると楽しいかと思います。

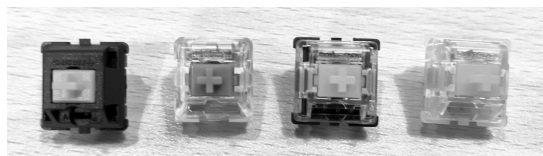


図 1 左から順に、Cherry MX(Red)、Kailh Speed(Copper)、Gateron MX(Red)、Gateron MX Silent(Red)

また、最近流行ってきているのが後者の Kailh ロープロファイルと呼ばれる、かなり薄い感じのキースイッチです。図 2 にあるのが、私が今使用している Helix というキーボードの左手側の部分なのですが、ずいぶんと薄くてコンパクトな感じになります。ただし、注意点として、CherryMX と Kailh ロープロファイルについては、基板に互換性がないため、基板が対応し

^{*4} <https://thenextweb.com/insider/2015/09/22/say-goodbye-to-words-the-emoji-keyboard-is-real/>

^{*5} https://www.reddit.com/r/MechanicalKeyboards/comments/dnrz8/science_isnt_about_why_its_about_why_not/

^{*6} <https://yushakobo.jp/>

^{*7} <http://silicon.kyohritsu.com/>

^{*8} https://www.sengoku.co.jp/shop_osaka.html

ていない場合は使うことができません。それでも、両方のスイッチに対応した設計になっている基板も存在するので、基板選定の際には、どのスイッチに対応しているのが確認が重要です。



図 2 Helix キーボード (左手)

で、打鍵感的なものについては、いわゆる「軸」というやつが関係していて、CherryMX については色によって表 1 のような特徴があります。余談ですが、2019 年 12 月現在、梅田ヨドバシのキーボード売り場には Cherry のキースwitchのテスターが置いてありますので、興味あれば直接さわってみてはいかがでしょうか。この間行ったときは、ずいぶんとわかりづらいところに引っこんでしまっていました。

また、互換スイッチでも、だいたい Cherry の同色の軸と同じような性能になってますが、たまに性能的に全然違うものや、Cherry には無い色 (Gateron の黄色いのか、Kailh の紫色だとか) とかもあったりするので、購入前には、実物を触るのが一番良いですね。それができなくても、スペックシートを見たり、どっかのブログを見たりするのが良いと思います。Kailh のロープロファイルについても、流通しているもので赤軸・白軸・茶軸と 3 種類あります。メーカーのサイト^{*9}を見てると他の軸もあるようなので、そのうち選択肢が広がるかもしれません。

表 1 Cherry MX の軸ごとの特徴

軸	クリック感	音	重さ	深さ
ピンク軸	なし	しずか	かるめ	
赤軸	なし	ふつう	かるめ	
茶軸	あり	ふつう	ふつう	
黒軸	なし	ふつう	おもめ	
青軸	あり	カチカチ	おもめ	
銀軸				浅め

参考: <https://www.diatec.co.jp/products/CHERRY/>

他にメジャーどころとして、ALPS というキーもあるみたいなのですが、現在は純正品は生産が停止されているそうです。根強い人気ようで、キットだけ販売されてたりするようですが……。また、ロープロファイルについては Cherry のロープロファイルもあったり、Kailh も Choc よりもさらに薄い X Switch と呼ばれるものがあってたりするようですが、遊舎工房さんとかでもまだ扱ってないので、いまのところ、イベントでその手の方のを触らせてもらうか、それらのキーを使用している市販キーボード (高級) を買うかしかなさそうです。なお、キースwitchについてですが、ピンソケットを付けて、後から付け換えできるようにすることもできます。基板が対応していたら、キースwitch専用のものを、していなくても、汎用のピンソケットをつかって対応している人もいます。

キーキャップ

次に重要なパーツとして、キーキャップがあります。まず、形 (プロファイル) について、DCS とか DSA とか色々あって、平たい部分の面積が違ったり、角度が違ったりと色々あるようです。が、私はまだこのあたり、くわしくないのであまりちゃんと説明できません……。キーキャップセットだけでも結構なお値段ですので!!

ただ、注意点としては、キーのサイズとしては普通のアルファベットのキーを 1U として、0.25 刻みで 1.25U、1.5U……といったぐあいのキーサイズが存在しています。少々ややこしいのが、キットによってキーのサイズが異なることです。たとえば、図 2 の Helix の場合、シフトキーやコントロールキー、全てが 1U になってるようなものもあれば、シフトキーやスペースキーは 2U 以上のサイズのキーであるようなものもあります。これの何がややこしいかというと、キーキャップとキーボードでキーのサイズが一致しないと、刻印が異なるキーを付けざるを得ない状況が発生するのです。たとえば、この後紹介しますが、私の利用している Mint60 というキーボードについては、スペースキーが 2 つに分割されたような配列なのですが、一緒に販売されていたキーキャップセットではそのサイズのスペースキーがなく、代わりにシフトキーを被せています。そして、シフトキーの位置には、テンキーの 0 が挿さってたりします。

^{*9} <http://www.kailh.com/en/Products/Ks/CS/>

マイコン (ProMicro)

元々は SparcFun 社が出しているマイコンボードで、Arduino の開発環境が使えるボードです。サイズが小さく、ピンもそこそこあり、かなり使い勝手の良いボードなのですが、定価だと 2,500 円くらいとちょっとお高いです。が、今では色々なクローンができて、それらが大変おやすく (500 円とか) 手に入るため、キーボード用ファームウェアの環境も整備されたことも手伝い、こちらのボードがこの界限では標準になっているようです。

ただ、このマイコンボード、いくつか問題もありまして、一番大きいのは USB のソケットが Micro-B で、そのソケットももげやすいというおまけ付きです。そのため、エポキシ接着剤で補強したり、ソケット強化版とか Type-C 版の ProMicro も販売されていたりします。また、そもそも無線キーボードじゃないと嫌だという人向けに BLE を積んだ ProMicro も販売されています。が、それらは結構な値段で販売されているため、微妙と言えば微妙……。それを嫌って、他のマイコンボードを使うキーボードもいくつかあります。

OLED/LED

また、キーボードと直接関係あるような無いようなのですが、自作キーボードキットには OLED のディスプレイや LED が付属するものが多いです。前者は、キーのレイヤー表示に使われたり、キットのロゴを表示させる等色々に使われています。LED に関しては完全に趣味ですね……。キーボードはカッコよく光るべきものらしいです。LED には、表面側を照らす Backlight 型と、底面側を照らす Underflow 型の二種類があります。

で、気をつけたいのが、LED がチップタイプのものを用いるキットです。このチップタイプのもの、小さい上に熱に弱いので、これがあるキットというだけではんだ付け難易度が跳ね上がります。特に Helix 系のキーボードは、本来 LED チップが想定している方法とは異なる方法で実装するため、ある程度はんだ付けに慣れてる人でないと難しいかもしれません。とは言え、光らすと図 3 のように、なかなか爽快感になるので、格好よさを求めるのであれば、是非挑戦していただきたいところです。



図 3 Helix キーボード (左手) を光らせたところ (カラーでないのが残念!!)

4.4 キットの選定・購入

ということで、ここまでの話を考慮した上でキットを選んでいくのですが、国内でキットを購入しようとした場合、おそらく遊舎工房さん、もしくは BOOTH^{*10}で購入する流れになるかと思います。他の手段ですと、キットを取り扱っている電子工作パーツ店をみつけたり、キーボード関連のミートアップ等の即売会に参加することになると思います。

キットによっては、ダイオードとして SMD タイプとよばれるすごく小さいものが付いてきたりします。私が組んだ Helix なんかはそうでした。こちらは、足が小さいためはんだ付け難易度は高いかと思います。ただ、実は Helix やその派生キーボードは、SMD タイプよりもはんだ付け難易度の低い足の付いた素子のタイプのダイオードも使えるようになっていますので、別途購入が必要ですが、たいした金額ではない (数百円程度) ので、欲しいキットがあってもはんだ付けに自信がなければ、そのような事ができないかを確認されても良いかもしれません。

また、やはり一番気になるのはキーの物理的な配列かと思います。キーの個数だとか、物理配列だとか、「そんな普通と違うようなキーボード打てる気がしない」とおっしゃる方もおられるかもしれません。実際、その通りでございまして、私のまわりで、キーが少な目で、column-staggered なキーボードを作成した二人は、「打つのにストレスが溜まる」と言い出して、そのキーボードをつかうのを諦めてしまったようです。ということで、各位におかれましては、いきなり奇抜なものとか Endgame を狙わずに、初めは普段使っているキーボードに近い形の自作キーボードに手を出す事を強くお勧めします。

^{*10} <https://booth.pm/ja>

4.5 キットの組み立て

キットを購入したら、当然組み立てが必要です。必須なものとしては「はんだ」「はんだごて」「はんだごて台」と「ドライバー」あたりでしょうか。「はんだ」は、無理して鉛フリーとかは止めといった方がよいと思います。また、「はんだごて」は温度の調整が可能なもの（比較的安めのももあります）、「はんだごて台」の汚れ落としは、水を使うのではなく、金色の金たわしみたいなやつ^{*11}の方が温度が下がらなくてオススメです。キーボードを作るのに必要なはんだ付けは、大抵そんなに難易度の高いものはないので、無理して高い物を買う必要はありませんが、この辺りの装備は、あまり安い物を買ったと後で後悔するかもしれません。

また、実際には「ラジオペンチ」「ニッパー」「ピンセット」もあった方が良いでしょう（素子タイプのダイオードならニッパーは必須）、「テスター」や「はんだ吸いとり器」もしくは「はんだ吸い取り線」は準備した方が良いでしょう。特にテスターについては、キーボードを作るとダイオードとキースイッチのはんだ付けでも 100 箇所以上となるので、大体どこかが接触不良だったりしますし、方向間違えたり、チップを壊したりとかもありますので、安めのものであった方が良いでしょう。

なお、キットですが、基本的には「基板」「ケース」「マイコンボード」に加えてスイッチやケーブルのコネクタなどの小物あたりがセットになって販売されており、「キースイッチ」「キーキャップ」およびケーブル類は別売りになっている事が多いです。そこは、自分で好きに選んでくれという話なのかと思います。また、キットによってはケースと PCB が別売りになっていることがあります。これは、ケースがアクリル以外にもステンレス製のものがあるキットもありますし、遊舎工房さんが、メジャーなキットについてはケースの色を選んで作ってくれるサービスをはじめたから、というのもありそうです。LED とか OLED が別売りであるキットもありますので、キット購入の際は、付けたいものが揃っているか確認するのをおすすめします。

Mint60 の場合

私が最初に作った Mint60 の場合、キーキャップやキースイッチもセットで販売してくれていますので、私の場合は素直にそれらコミコミのセットで注文しました。届いて開封したときの写真を図 4 に掲載してみました。



図 4 Mint60 開封の儀

これを、図 5 のように、リセットスイッチや通信用のケーブルのジャックや、ProMicro、ダイオード、キースイッチをはんだ付けしてだけです。なお、一番右の図で PCB にキースイッチだけ生えてますが、実はこれは失敗例で、間にアクリルのカバーを挟んで実装するのが正解です。気をつけましょう。間違えると、泣きながらはんだ吸い取り器を Amazon でポチり、キースイッチ全と ProMicro を鳴咽しながら剥ぎ取るという不毛な作業が必要になります……。

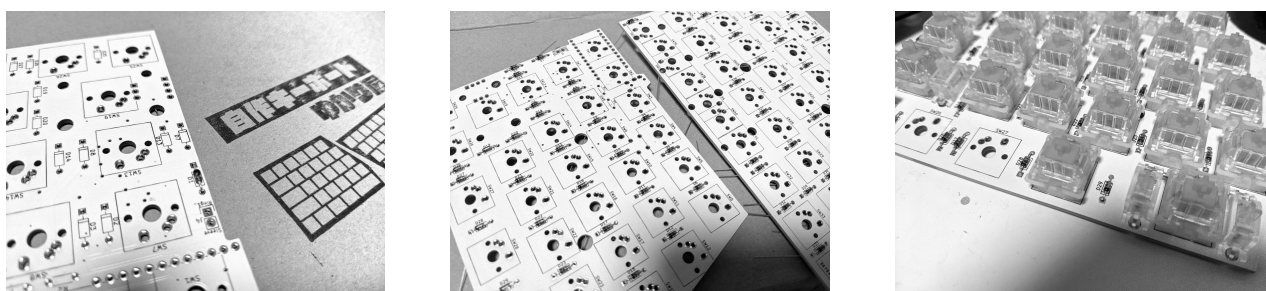


図 5 Mint60 組み立て

完成した（左半分）が図 6 になります。キーキャップの項でも書きましたが、キーキャップの刻印と齟齬の生じるキーが出てきています。本来スペースのところのシフトキーとか、本来シフトキーのところにテンキーの 0 キーだとか。また、ESC キーは、デフォルトのキーマップだと左下で刻印通りなのですが、セットで販売されていたキーキャップは、ESC キーは手前側に傾斜のある

^{*11} 会社の人は「ゴールデンアフロ」と呼んでいました。

タイプだったので、よく見ると隣の Ctrl キーと盛り上がり方が異なっていたりします。このあたりは、割り切るなり、なにか他の良い感じのキーキャップを持ってくるなど工夫が必要そうです。

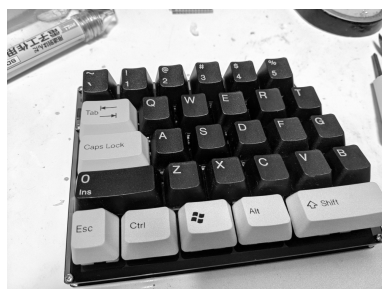


图 6 Mint60 完成品 (左半分)

4.6 ファームウェアのカスタマイズ

物理的にキーボードを作ったとしても、その後キーの配列を自分の好みに合わせていくという終わりの無い旅がはじまっていきます。配列をいじるにはマイコンのファームを変更していくことになります。キーボードのファームウェアについて、大体のキットが、`qmk_firmware`^{*12} という、ファームを焼くところまで `make` コマンド一発でできてしまう環境を使っています。

さて、ここでもうやく Debian の影が見えますが、この qmk_firmware を使用するためには、Debian の場合は以下のパッケージのインストールが必要です。

```
% sudo apt-get install gcc unzip wget zip gcc-avr binutils-avr avr-libc dfu-programmer dfu-util \
> gcc-arm-none-eabi binutils-arm-none-eabi libnewlib-arm-none-eabi
```

この状態で、まずはキーボードのキー配列を換えてみます。といっても、実に簡単でして、環境の `keyboards/(キーボード名)/keymaps/(キーマップ名)/keymap.c` を覗いてみると、以下のようなコードが記述されています。

```
[O] = LAYOUT( \
KC_GRV, KC_1, KC_2, KC_3, KC_4, KC_5, KC_6, KC_7, KC_8, KC_9, KC_0, KC_MINS, KC_EQL, KC_BSPC, \
KC_TAB, KC_Q, KC_W, KC_E, KC_R, KC_T, KC_Y, KC_U, KC_I, KC_O, KC_P, KC_LBRC, KC_RBRC, KC_BSLS, \
KC_CAPS, KC_A, KC_S, KC_D, KC_F, KC_G, KC_H, KC_J, KC_K, KC_L, KC_SCNL, KC_QUOT, KC_ENT, \
KC_LSFT, KC_Z, KC_X, KC_C, KC_V, KC_B, KC_N, KC_M, KC_COMM, KC_DOT, KC_SLSH, KC_RSFT, KC_UP, MO(1), \
KC_ESC, KC_LCTL, KC_LGUI, KC_LALT, MO(1), KC_SPC, KC_ENT, LALT(KC_GRV), KC_LEFT, KC_DOWN, KC_RIGHT \
).
```

ええ、もうおわかりですね、上記の配列*13を良い感じに書き換えるだけです。Debian さわってるような人であれば、テキストファイルをさわるなんてなんの抵抗もないのではないのでしょうか。

さて、よく見ると上記の配列にはファンクションキーの定義がありません。もちろんそれらはキーの組み合わせで打つことになります。上記の配列の場合、MO(1)となっているキーがありますが、これを押しながら他のキーを押すと、上記の配列以外のキー入力としてPC側へ送信できます。なお、それらのコードも以下のような配列を変更するだけです。

```
[1] = LAYOUT( \
    KC_ESC, KC_F1, KC_F2, KC_F3, KC_F4, KC_F5, KC_F6, KC_F7, KC_F8, KC_F9, KC_F10, KC_F11, KC_F12, KC_DEL, \
    RGB_TOG, RGB_BST, RGB_HUI, RGB_SAI, RGB_VAI, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, \
    XXXXXXXX, RGB_MOD, RGB_HUD, RGB_SAD, RGB_VAD, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, \
    -----, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, \
    XXXXXXXX, -----, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, XXXXXXXX, KC_PGUP, -----, \
    )
```

この後、以下のコマンドを叩き、キーボードのリセットスイッチを押せばファームウェアの更新が実行されます。

```
% make (キーボード名):(キーマップ名):avrdude
```

また、このレイヤーの切り替えも、キーを押しながらだけでなく、カナキーよろしくトグル動作といった具合にもできます。その場合、現在のモードは、`qmk_firmware` に用意された仕組みで OLED に表示させること等ができます。これによって、QWERTY と Dvorak を切り替えてつかったりもできるわけです。

もちろん、単純なキーではなく、ショートカットや文字列入力を 1 キーに割り合えることもできます。たとえば、Twitter 廃人らしく、“#debian #kansaid.debian” を 1 キーで入力できるようにしてみます。先程のキー配列のあったファイルに以下のよう

*¹² https://github.com/qmk/qmk_firmware

*13 LAYOUT がマクロ関数になってるので、配列に見えないかもしれませんが。

な enum 値やマクロを追加し、

```
enum macro_keycodes {
    KC_SAMPLEMACRO,
    KC_HASH_DEBIAN, /* 追加 */
};

//Macros
#define M_SAMPLE M(KC_SAMPLEMACRO)
#define M_HASH_DEB M(KC_HASH_DEBIAN) /* 追加 */
```

さらに配列を書き換え……、

```
const uint16_t PROGMEM keymaps[][MATRIX_ROWS][MATRIX_COLS] = {

    /* 中略 */
    KC_HASH_DEBIAN, KC_LCTL,KC_LGUI,KC_LALT, MO(1), KC_SPC, KC_ENT, LALT(KC_GRV), KC_LEFT,KC_DOWN,KC_RGHT \
    /* 後略 */
```

最後に、キー入力関数の case 文を追加すると……

```
bool process_record_user(uint16_t keycode, keyrecord_t *record) {
    switch (keycode) {

        /* 中略 */

        case KC_HASH_DEBIAN:
            if (record->event.pressed)
                SEND_STRING("#debian #kansaidebian");
            break;

        /* 後略 */
```

1 ボタンで debian 向けハッシュタグを打ててしまいます。この機能を使うと、他にも例えばキーを短押ししたときは普通のキーとして、長押ししたらレイヤー切り替えのキーとして動作する、といったような事もできます。他にも、LED は OLED の制御等も色々と触れますが、ProMicro だとファームサイズはそこまで大きくできませんので、ご注意ください。

4.7 それでも満足できないアナタへ

ここまで読んでいただいてわかるように、本当に自分の Endgame を目指すのであれば、結局それに合ったキットがなければどうしようもありません。ところで、最近はプリント基板こと PCB は気軽に作れるようになってきている……らしいです。実際、10cmx10cm であれば、送料を考えなければ (お) 5\$とかで作れてしまうようです。ということで、Endgame のために基板から自作しましょう (錯乱)。私もまだ志半ばですので、是非みなさんも一緒に!!

基板設計ってどうやるの、と不安になるかもしれません。しかしそこはご安心ください、PCB の設計用のツールはフリーの KiCad と呼ばれるソフトがあり、ちゃんと Debian パッケージもございます (申し訳程度の Debian 要素その 2)。

```
% apt search '^kicad$'
ソート中... 完了
全文検索... 完了
kicad/unstable,now 5.1.5+dfsg1-1 amd64 [インストール済み]
  電子回路および PCB 設計ソフトウェア
% apt install kicad
```

KiCad を入れたら、まずは回路図から作っています。その際、キーボード用のパーツ (キースイッチとか ProMicro) のライブラリが必要になりますが、文献 [1] にて、著者の方が KiCad 用のライブラリを公開してくれています^{*14}。この文献、この記事なんかよりも詳細な KiCad での設計方法が書かれていますので、興味があれば是非お買い求めください。ライブラリを適当な場所に clone したら、メニューの「シンボルライブラリーを管理」から登録できます。例として、図 7 に、私がテストで作りかけている 4x4 のキーボードの回路図を掲載しておきます。左上に ProMicro が、右上にキーマトリックス回路が見えてると思います。

次に、フットプリント (基板上のはんだ付け部分) を回路と関連付けていくのですが、その際にもフットプリントのライブラリが必要となります。そちらのライブラリについても、[1] の著者の方が先程と同様の場所に公開してくれています。作業としては、図 9 のような感じで、回路図の各シンボルについて、どのフットプリントを適用するのかという事を関連付けていくことになります。

フットプリントの関連付けが終わると、次はレイアウトになります。パーツが干渉しないように注意して……やるのですが、私はこの作業の途中で力尽きております……。画面としては、図 9 のような感じになります。

その後の流れとしては、

1. エッジカット線を引く
2. スペーサー用の穴をあける

^{*14} <https://github.com/foostan/kbd>

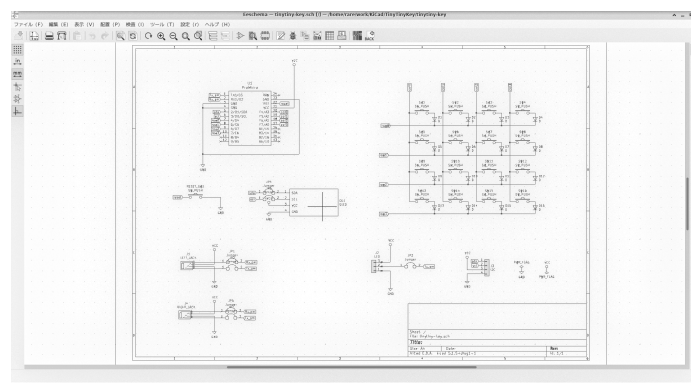


図 7 KiCad の画面 (回路図エディタ)

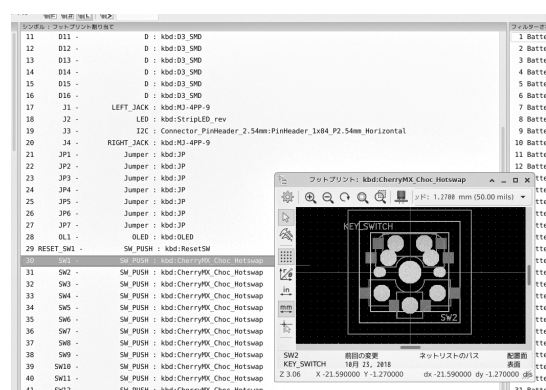


図 8 KiCad の画面 (フットプリントの関連付け)

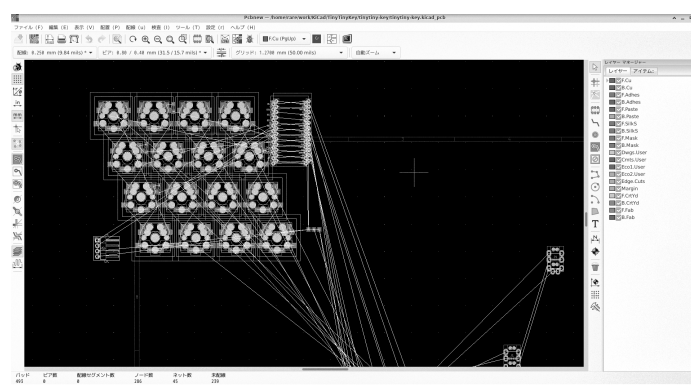


図 9 KiCad の画面 (レイアウト.....の途中で力尽きているところ)

3. 配線を実施する (一番大変らしい)

4. 必要に応じてケースを作成する

- アクリル板を加工するのが主流
- 最近は 3D プリンタ勢が増えているみたい.....

といったような感じになるようです。PCB 自体は、設計ができればプリント基板の実装サービスをやっている場所に発注するようです。seed^{*15}や ELECROW^{*16} といったところがメジャーなようです。海外なので送料はそこそこかかってしまいますが、10cmx10cm のサイズであれば、相当安く作ってくれるようです。

なお、今回は試しにと基板をまっさらなところから作成していますが、実は回路図データを公開しているキットは結構あります。先程紹介した Helix がその一つで、回路図が GitHub 上に公開されています^{*17}。そのため、この Helix の PCB をベースにして作成されたキットも数多くあります。なお、KiCad については、文献 [1], [2] あたりが参考になるかと思います。

^{*15} https://www.seeedstudio.com/fusion_pcb.html

^{*16} <https://www.elecrow.com/pcb-manufacturing.html>

^{*17} <https://github.com/MakotoKurauchi/helix>

4.8 まとめ

ということで、最後の基板からの自作については少々尻すぼみな感じになってしまいましたが、キットから作るという選択肢を取れば、結構簡単に自作できてしまうのでした。私は、分割式のキーボードを使うようになって、はや2ヶ月とか経ちますが、腰痛に関しては冗談抜きで本当に改善しました。以前は、一時間も座っていたら座っていられないくらい辛くなっていた腰痛が、もちろんまったく痛くならないわけではないのですが、座っていられないような事はなくなりました。逆に、Fitbit に課される「1 時間 250 歩」が全然クリアできなくなって困っております。

Debian 勉強会で、この内容についてしゃべったところ、自己紹介でのキーボード遍歴で大変に盛り上がりました。おそらく、今回 Debian 本を読まれている方につきましても、キーボードに関しては色々と言語することもあるかと思います。Debian をハックするには、まずは健康から!! どうでしょう、自作キーボード!! 自作キーボードについては、「Self-Made Keyboards in Japan」という Discord のサーバー [3] に結構な情報が集まっています。また、Discord はちょっと.....という方には、Youtube などで情報を集めるのも良いかと思います。「ほぼ週刊キーボードニュース」[4] あたりはおすすめです。本記事が、みなさまの健康的なハックライフに繋がると幸いです。

4.9 参考文献

- [1] foostan, “自作キーボード設計入門 Endgame への第一歩”, <https://booth.pm/ja/items/1044084>
- [2] 小坂貴美男, “KiCad 5.0 / 5.1 入門実習テキスト『KiCad Basics for 5.x』”, <https://booth.pm/ja/items/941963>
- [3] @Biacco42, “Self-Made Keyboards in Japan”, <https://discordapp.com/invite/NM7XtDW>
- [4] @Biacco42 and @Pekaso, “ほぼ週刊キーボードニュース”, https://www.youtube.com/channel/UCyU1PAGvw_suAyI4w1jHmag

5 Debian GNU/kFreeBSD セットアップガイド 2019 年版

杉本典充



5.1 はじめに

Debian Project では Debian GNU/Linux という Linux ディストリビューションを開発しており多くの開発者及び利用者がいます。Debian Project ではユニバーサルなオペレーティングシステムを提供する考え方をもっており、その理念に則り FreeBSD カーネルで動作する Debian GNU/kFreeBSD という別の OS^{*18} があります。

Debian GNU/kFreeBSD はその特異さゆえに Debian GNU/Linux と異なる点が多くあります。今記事では Debian GNU/kFreeBSD を触れるにあたり、どのようにセットアップを行うとよいか説明します。

5.2 Debian Ports と Debian GNU/kFreeBSD

5.2.1 Debian Ports とは

Debian Ports^{*19}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です。

FreeBSD カーネルで動作する 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）。

5.2.2 Debian GNU/kFreeBSD のリリースの歴史

Debian GNU/kFreeBSD のリリースの歴史は以下となっています。

- 2011 年 2 月 6 日、Debian 6 (squeeze) おいてテクノロジープレビューとして stable 版をリリース。ftp.debian.org にてパッケージの配布を開始。
- 2013 年 5 月 4 日、Debian 7 (wheezy) において stable 版をリリース。
- 2014 年 9 月、Debian 8 (jessie) でリリースするアーキテクチャを選定する過程で、kFreeBSD は開発や保守に関わる人が少なく stable の維持と新規開発の両立ができていない状況が見られることを指摘される^{*20}。その指摘を改善できなかったため、Debian 8 (jessie) では stable 版のリリースは見送られる^{*21}。
- Debian 9 (stretch) においても stable 版のリリースはされない状況が続く^{*22}。
- 2018 年 5 月 31 日、Debian 7 (wheezy) の LTS が終了。これにて Debian GNU/kFreeBSD の stable 版はサポートを終了。
- Debian 10 (buster) のリリースに向けて ftp.debian.org の整理を実施。
- 2019 年 5 月 25 日、Debian GNU/kFreeBSD は Debian Ports (<https://www.ports.debian.org/>) へ移転。これにより ftp.debian.org におけるパッケージの配布を終了。

^{*18} Linux カーネルでなく FreeBSD カーネルでもない Debian GNU/Hurd というものもあります。

^{*19} <https://www.ports.debian.org/>

^{*20} <https://lists.debian.org/debian-devel-announce/2014/09/msg00002.html>

^{*21} 過去バージョンの利用者が困るであろうということで jessie-kfreebsd 版という裏バージョンが存在していた。

^{*22} stretch-kfreebsd 版はリリースされませんでした。

5.2.3 Porterbox

Debian Project では多くのアーキテクチャを開発しているため、移植作業用にサーバを借りる「porterbox」という仕組みがあります^{*23*24}。

bugreport にはあるアーキテクチャのみで発生する不具合も報告されますが、パッケージをメンテナンスしている人がそのアーキテクチャの環境を持っていない場合があります。その場合は環境を自分で構築するか、porterbox を利用してバグの修正作業を行います。

なお、Debian GNU/kFreeBSD では kfreebsd-amd64 の porterbox ”lemon.debian.net” を運用しています^{*25*26}。

5.2.4 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。

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。Debian Ports の kFreeBSD では kfreebsd-image-10.3 が利用できます。experimental には kfreebsd-image-11 があります^{*27}。

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。

freebsd-utils パッケージ

freebsd-utils パッケージは、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。

freebsd-net-tools パッケージ

freebsd-net-tools パッケージは、FreeBSD 向けのネットワーク操作のコマンドを含んだパッケージです。arp、ifconfig、netstat、route コマンドなどが入っています。

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}
```

freebsd-ppp パッケージ

freebsd-ppp パッケージは、ダイヤルアップする ”/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 パッケージでダイヤルアップしますが、kfreebsd ではそれらは使えないため注意が必要です。

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

^{*23} <https://wiki.debian.org/PorterBoxHowToUse>

^{*24} 実際に借りた事例の報告「Debian の移植作業用のインフラを借りるには」<https://tokyodebian-team.pages.debian.net/pdf2016/debianmeetingresume201603.pdf>

^{*25} <https://db.debian.org/machines.cgi?host=lemon>

^{*26} ”New GNU/kFreeBSD Porterbox” <https://lists.debian.org/debian-devel/2019/02/msg00313.html>

^{*27} FreeBSD の最新版は 2018 年 12 月 11 日にリリースした FreeBSD-12.0 です。

パッケージです。

/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。

5.3 Debian GNU/kFreeBSD のインストール

5.3.1 インストールする PC

今回は ThinkPad X220 という Intel の Sandy Bridge 世代のノート PC にインストールをしてみます。このノート PC の有線 LAN と無線 LAN のチップは Intel 製であり FreeBSD カーネルの標準でドライバが提供されているため認識させやすいです。

5.3.2 インストールイメージの入手

開発者向けの Debian Installer は通常 <https://www.debian.org/devel/debian-installer/> で配布しています。しかし、Debian GNU/kFreeBSD は Debian Ports へ移行したため、<https://www.debian.org/devel/debian-installer/> にある kFreeBSD 用インストーラは動作しなくなっています。

代わりに非公式版の Debian Installer の ISO イメージが用意されており、以下の ISO ファイルを利用してインストールします。

通常は kfreebsd-amd64 版^{*28}を使うとよいでしょう。kfreebsd-i386 版^{*29}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ので注意してください。

ISO ファイルを CD/DVD メディアへ焼いてインストールディスクを作成し、インストールディスクから PC を起動すると Debian Installer が起動します。

5.3.3 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。

5.3.4 パーティション構成とファイルシステム

Debian GNU/kFreeBSD を MBR 方式でブートする環境へインストールする場合は 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^{*30}（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）。kFreeBSD を占有する環境の場合は以下のパーティション構成でよいでしょう（以下では "/dev/ada0p1" のファイルシステムに UFS を選択しています）。

```
# fdisk -l /dev/ada0

Disk /dev/ada0: 238.5 GiB, 256060514304 bytes, 500118192 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: 0x88f37f78

Device      Boot      Start          End      Sectors      Size Id Type
/dev/ada0p1                2048    273436671    273434624    130.4G a5 FreeBSD
/dev/ada0p2    273436672    281249791     7813120       3.7G 82 Linux swap / Solaris
```

ファイルシステムは通常の利用する分には UFS を選択することが多いです。他には ZFS というスナップショット機能をもち多数のストレージを扱えるファイルシステムを選択することができます。

5.3.5 パッケージをダウンロードするミラーサーバの指定

Debian Installer が利用するミラーサーバは <http://deb.debian.org/debian/> がデフォルトになっています。しかし、kfreebsd-amd64 のパッケージは ftp.ports.debian.org にて配布しているためデフォルト値は利用できません。ひとまずミラーは利用しない選択をして処理を進めて "Standard system utilities" のみをインストールします^{*31}。

^{*28} http://jenkins.kfreebsd.eu/jenkins/view/cd/job/debian-cd_sid_kfreebsd-amd64/ws/build/debian-unofficial-kfreebsd-amd64-NETINST-1.iso

^{*29} http://jenkins.kfreebsd.eu/jenkins/view/cd/job/debian-cd_sid_kfreebsd-i386/ws/build/debian-unofficial-kfreebsd-i386-NETINST-1.iso

^{*30} もし Debian GNU/Linux や Windows とデュアルブートしたい場合は、MBR 方式では基本パーティションの作成は 4 つまでという制約を考慮したパーティション構成にする必要があります。

^{*31} <http://ftp.ports.debian.org/debian-ports/> を指定すればよさそうなものなのですが、入力してもパッケージが見つからないエラーが起こり、先に進めませんでした。

5.4 Debian GNU/kFreeBSD のセットアップ

5.4.1 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 向けの場合は "em0"、realtek の場合は "re0")。設定ファイルは Debian GNU/Linux と同じ "/etc/network/interfaces" を参照しますが、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 では利用できず注意が必要です。

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
```

なお、ネットワークインタフェースの状態確認は昔ながらの "ifconfig" コマンドを利用してください (最近の linux と違い ip コマンドはありません)。

5.4.2 apt の設定

インストール時にミラーを指定しなかったため、/etc/apt/sources.list ファイルは中身がない状態になっています。/etc/apt/sources.list は以下を設定します^{*32}。

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

deb-src http://ftp.se.debian.org/debian/ sid main contrib non-free
deb http://ftp.ports.debian.org/debian-ports sid main
deb http://ftp.ports.debian.org/debian-ports unreleased main
deb-src http://ftp.ports.debian.org/debian-ports unreleased main
deb http://ftp.ports.debian.org/debian-ports experimental main
```

上記の apt の設定を行い apt-get コマンドを実行すると、GPG キーエラーが発生して先に進めません。手動で keyring をインストールします。

```
# wget http://ftp.jp.debian.org/debian/pool/main/d/debian-ports-archive-keyring/debian-ports-archive-keyring_2018.12.27_all.deb
# dpkg -i debian-ports-archive-keyring_2018.12.27_all.deb
```

これで apt-get が動作するようになります。他の keyring も更新しておきましょう。

```
# apt-get update
# apt-get install debian-keyring debian-archive-keyring
```

5.4.3 locale の設定

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化けるため C.UTF-8 の方がデバッグしやすいかもしれません)。

```
# dpkg-reconfigure locales

-> ja_JP.UTF-8 を選択する
```

5.4.4 sshd のインストール

openssh-server パッケージを利用できます。以下コマンドでインストールできます。

```
# apt-get install openssh-server
```

^{*32} <https://lists.debian.org/debian-bsd/2019/08/msg00005.html>

5.4.5 無線 LAN

無線 LAN のハードウェアの設定

ThinkPad X220 に搭載している無線 LAN チップは「Intel Centrino advanced-N 6205」です。

無線 LAN を利用するためにまずはメーカーが配布する無線 LAN 用 firmware をインストールします。ただ、`/etc/apt/sources.list` に `non-free` のバイナリパッケージのダウンロード先を指定をしていないため、パッケージを直接ダウンロードしてインストールします。reboot して firmware を認識すると、`dmesg` に出力されていたファームウェアがロードできないというエラーがなくなります。

```
# wget http://ftp.se.debian.org/debian/pool/non-free/f/firmware-nonfree/firmware-iwlwifi_20190717-2_all.deb
# dpkg -i firmware-iwlwifi_20190717-2_all.deb
# reboot
```

無線 LAN の通信を制御するデーモンである `wpa_supplicant` パッケージ をインストールします。

```
# apt-get install wpa_supplicant
# wpa_supplicant -v
wpa_supplicant v2.4
Copyright (c) 2003-2015, Jouni Malinen <j@w1.fi> and contributors
```

無線 LAN のネットワーク設定

FreeBSD は物理無線 LAN インタフェースと論理無線 LAN インタフェースに分かれています。

ThinkPad X220 に搭載している Intel 製無線 LAN チップの場合は `iwn0` という物理無線 LAN インタフェースがあります。

```
# ifconfig iwn0

iwn0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 2290
    ether yy:yy:yy:yy:yy:yy
    media: IEEE 802.11 Wireless Ethernet autoselect (autoselect)
    status: no carrier
    nd6 options=23<PERFORMNUD,ACCEPT_RTADV,AUTO_LINKLOCAL>
```

以下のコマンドで論理無線 LAN インタフェースを生成します。`wlan0` というインタフェースが生成されました。

```
# ifconfig wlan create wlandev iwn0
wlan: Ethernet address: yy:yy:yy:yy:yy:yy
wlan0

# ifconfig wlan0
wlan0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether yy:yy:yy:yy:yy:yy
    ssid " channel 1 (2412 MHz 11b)
    country US authmode OPEN privacy OFF txpower 14 bmiss 10 scanvalid 60
    bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 1 wme
    bintval 0
    media: IEEE 802.11 Wireless Ethernet autoselect (autoselect)
    status: no carrier
    nd6 options=23<PERFORMNUD,ACCEPT_RTADV,AUTO_LINKLOCAL>
```

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。

```
$ wpa_passphrase apname1 appassword > wpa_apname1.conf
$ cat wpa_apname1.conf
network={
    ssid='apname1'
    #psk='appassword'
    psk=e9fdbc43eba09b6342df30f14275625c8494e534799a82d6639b6124434ea627
}
```

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 zz:zz:zz:zz:zz:zz (SSID='apname1' freq=2427 MHz)
wlan0: Associated with zz:zz:zz:zz:zz:zz
wlan0: WPA: Key negotiation completed with zz:zz:zz:zz:zz:zz [PTK=CCMP GTK=CCMP]
wlan0: CTRL-EVENT-CONNECTED - Connection to zz:zz:zz:zz:zz:zz completed [id=0 id_str=]
wlan0: WPA: Group rekeying completed with zz:zz:zz:zz:zz:zz [GTK=CCMP]
```

```
# dhclient wlan0

# /sbin/ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
ether yy:yy:yy:yy:yy:yy
inet6 fe80::a288:b4ff:fe18:b514%wlan0 prefixlen 64 scopeid 0x6
inet 192.168.1.5 netmask 0xfffff00 broadcast 192.168.1.255
ssid apname1 channel 4 (2427 MHz 11g) bssid zz:zz:zz:zz:zz:zz
country US authmode WPA2/802.11i privacy ON deftxkey UNDEF
AES-CCM 3:128-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/54Mbps mode 11g
status: associated
nd6 options=23<PERFORMNUD,ACCEPT_RTADV,AUTO_LINKLOCAL>
```

無線 LAN 接続のトラブル

wpa_supplicant 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。^{*33}

5.4.6 音の再生

FreeBSD は OSS（Open Sound System）という仕組みで音声出力をします（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 FreeBSD kernel に static link されています）。

以下のコマンドを実行すると、コンソール上で MP3 形式の音声ファイルを再生できます。

```
# apt-get install mpg123
# mpg123 ${mp3 ファイルのパス}
```

5.4.7 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。

電源に接続時は 2501 MHz で動作しています。

```
# sysctl dev.cpu.0.freq
dev.cpu.0.freq: 2501
```

電源に接続しておらずバッテリー動作時は 800 MHz で動作しています。

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

バッテリー残量を取得するには acpicnf コマンドを実行します。

```
# /usr/sbin/acpicnf -i 0

Design capacity:      57240 mWh
Last full capacity:   51530 mWh
Technology:           secondary (rechargeable)
Design voltage:       10800 mV
Capacity (warn):      2576 mWh
Capacity (low):       200 mWh
Low/warn granularity: 1 mWh
Warn/full granularity: 1 mWh
Model number:         45N1172
Serial number:        641
Type:                 LION
OEM info:             SANYO
State:                high
Remaining capacity:    99%
Remaining time:        unknown
Present rate:         0 mW
Present voltage:       12154 mV
```

^{*33} 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。筆者の自宅で利用している NEC 製 Wifi ルーター Aterm に接続するにはこれを指定しないとうまくつながりませんでした。

サスペンドとハイパーネートについては未確認です。

5.4.8 KMS の有効化

KMS (kernel mode settings) を有効にするとコンソール画面の解像度が上がります。i915.ko が KMS 非対応で、i915kms.ko が KMS 対応のカーネルモジュールです。これらのカーネルモジュールを利用するときはどちらかひとつのみロードします。

以下コマンドで KMS を有効にできます。

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

再起動後も自動で kernel module をロードするには以下のように設定します。

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

5.4.9 X Window System

X Window System 用ビデオドライバ

ThinkPad X220 は Intel のオンボード GPU を搭載しています。そのため、ビデオドライバは xserver-xorg-video-intel を利用します。インストールするには以下のコマンドを実行します。

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

今の kFreeBSD で安定動作させるには Intel のドライバの動作モードを古い UXA モードに変更するとよいようです。

```
# vi /etc/X11/xorg.conf.d/50-intel.conf
Section "Device"
    Identifier "Card0"
    Driver      "intel"
    Option      "AccelMethod" "uxa"
EndSection
```

ウィンドウマネージャ

twm をインストールすることが可能です。それ以外の icewm、xfce4、lxde などはパッケージの依存関係が解決できない状態のためインストールできません。グラフィカルログインマネージャの xdm、lightdm もパッケージの依存関係が解決できない状態のためインストールできません。

ターミナルエミュレーターも必要になるため、xterm、eterm をインストールしておきます。

```
# apt-get install twm xterm eterm
```

コンソール環境において root ユーザで startx コマンドを実行すると twm を起動できます。

```
# startx
```

twm の画面上で左クリックをするとメニューが表示されますので、eterm を起動すれば CLI の操作ができます^{*34}。

5.5 Debian の開発ツールのセットアップ

5.5.1 インストールできるパッケージ

以下のパッケージは、/etc/apt/sources.list に設定したリポジトリからバイナリパッケージをインストールできます。

- make / bmake
- gcc-9 / g++-9
- clang-7
- dpkg-dev
- devscripts
- build-essential

^{*34} xterm は "open ttydev: No such file or directory" というエラーが出て起動できませんでした。

- debhelper
- debootstrap
- subversion
- vim

5.5.2 インストールができない状態になっているパッケージ

以下のパッケージは ftp.ports.debian.org のあるパッケージ群で依存関係が壊れているためインストールできないパッケージです。

- clang-8
- git
- emacs / emacs-nox

5.6 その他の機能の紹介

5.6.1 コンテナ環境 : Jail

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コンテナを操作するコマンド群をインストールすることができます。コンテナ環境の作成は debootstrap、コンテナ環境の操作は jail、jls、jexec などが利用できます。

```
# debootstrap --no-check-gpg sid ./jail_demo_1 http://ftp.ports.debian.org/debian-ports/
```

以下コマンドで chroot コマンドのように jail 環境に入れます^{*35}。

```
# jail -c path=./jail_demo_1 command=/bin/bash
```

jail 環境を起動した後は、jls コマンドで jid を取得し、jexec で jail 環境の中で任意のコマンドを実行できます。

```
# jls
JID  IP Address      Hostname          Path
1    -                /root/jail/jail_1

# jexec 1 cat /etc/debian_version
```

5.6.2 Linux エミュレーション

debootstrap で Debian 6 squeeze の linux-i386 の コンテナ環境を作成しようとしたところエラーが発生しました^{*36}。現在は動かないのかもしれませんが。

```
# debootstrap --no-check-gpg --arch=i386 squeeze ./linux_demo_1 http://archive.debian.org/debian/
ELF binary type "0" not known.
E: Unable to execute target architecture
```

5.6.3 完全仮想化環境

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。

5.7 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。bugreport で kfreebsd 環境で動かないと報告がある場合や kfreebsd を開発してみたい方は参考にしてみてください。

^{*35} hostname を指定するオプション、ipv4 アドレスを指定するオプションは省略しています。

^{*36} kfreebsd-image-10.3 の linux.ko は linux-2.6.32 の 32bit ABI のエミュレーションだったと記憶しています。

6 DebConf19 の気になるセミナーを調べてみる

杉本典充



6.1 はじめに

DebConf19 が 2019 年 7 月 21 日から 28 日にブラジルのクリチバ市で開催されました。

<https://debconf19.debconf.org/>

多くのセミナー発表があり、現在の Debian Project のトレンドをつかむことができます。このセミナーはビデオ公開しているセミナーも多いため、ビデオを見てみてどんな内容だったのか調べてみました。

6.2 DebConf19 のセミナービデオ

6.2.1 セミナースケジュールとビデオの場所

DebConf19 のセミナースケジュールは以下の URL で公開しています。

<https://debconf19.debconf.org/schedule/>

この URL のページ内にある各セミナーのタイトルをクリックするセミナーの概要のページが表示され、セミナーによってはスライドとビデオが見れるようになっています。

6.2.2 気になるセミナー

私が個人的に気になるセミナーは以下です。

- i'm (a bit) sick of maintaining piuparts.debian.org (mostly) alone, please help (20min)
- APT 2.0 and other news (20min)
- Using Debian Containers for Cross-Development (45min)
- Debian LTS, the good, the bad and the better (45min)
- Who's afraid of Spectre and Meltdown? (45min)
- Escaping the Surveillance Blackhole with Free Mobile Computing (45min)
- Bootstrappable Debian BoF (45min)
- debian/copyright BoF (45 min)
- Bits from the DPL (45min)

6.3 ビデオを見てみたセミナー

6.3.1 APT 2.0 and other news

- Ubuntu 14.04 の ESM が始まった
 - public dist/、private pool/ が要件。認証処理が必要
 - "never" pin という仕組みを考えた
- 2018 年に frontend lock を導入
 - 主に python-apt が抱える不具合を修正
 - Ubuntu error トラッカーの情報では unattended-upgrades による crash の報告が減った
- アップデート中にシャットダウンした場合の問題への対応

- apt 1.8.1 以降では、systemd へ shutdown しないように制御できるようになった
 - systemctl や GUI ツールの実行中は shutdown しないように防ぐことができる
 - sudo reboot や似たようなコマンドはまだ対応していない
- buster へ dist-upgrade するときにエラーが出たのは申し訳なかった
 - エラー内容
 - * E: Repository 'foo' changed its 'Suite' value from 'testing' to 'stable'
 - * E: Repository 'bar' changed its 'Codename' value from 'buster' to 'bullseye'
 - 何がやりたかったのか
 - * Security: Prevent people from giving you stable when you ask for stable-security
 - * Pinning: Prevent broken pinning situations
 - どうすればよかったのかの案を考えてみたので、みなさんと議論してみたいです
- dpkg の news
 - Trigger loops might finally be over
 - Rootless ビルドできるようになった
 - start-stop-daemon は新しい systemd の readiness protocol に対応
 - CPAN に Dpkg perl モジュールが入りました
 - クロスコンパイルサポートの改善
- APT 2.0
 - できるかどうかわからないこと
 - * “purge”、“installed”、“config-files”などでパターンマッチング指定を採り入れたい
 - * pinning もパターンマッチング指定を採り入れたい
 - * Pin and upgrade by source package ?
 - Mandatory InRelease
 - * “InRelease” ファイルを見るようにする (“Release” や “Release.gpg” ファイルはもう参照しない)
 - * まだ決まっていないこと
 - ・署名なしリポジトリにも対応するよう配慮すべきか ?
 - ・gpg 署名ありリポジトリから gpg 署名なしリポジトリへフォールバックしたときにアクシデントが起きたりしないか ?

6.3.2 i'm (a bit) sick of maintaining piuparts.debian.org (mostly) alone, please help

- 2009 年から piuparts.debian.org をメンテナンスしている Holger Levsen さんの発表
- piuparts の説明
 - パッケージのインストール、アップグレード、削除のテストスイートである
 - 現在 59 スイートとそのスイートの組み合わせのテストを実行している
 - piuparts のメンテナンスは実質 2 名でやっている
 - 悲しいことにバグ対応は Andreas Backmann さんがすべてやっている
 - コードのほとんどは、Andreas Backmann さんが書いた
 - Herbert Fortes さんの help には希望が持てる
 - bullseye にはまだ対応していない
 - buster では stable2sid が壊れているとバグ報告がある
- src:piuparts について
 - 99.9% の処理は /usr/sbin/piuparts と piupartslib の 5 ファイルが占めている
 - しかし src:piuparts パッケージはファイル数が 358 と多い
 - テストは 5 件しかない
- 複数のアーキテクチャへの対応
 - 現在は amd64 のみ稼働中
 - 複数のアーキテクチャに対応するコードを piuparts-master に入っている
 - サーバリソースを欲しいと RT ticket に書いて DSA に依頼するもサーバをもらえていない
 - モチベーションが下がっている
- buster の状況
 - piuparts-1.0.0 が入っている
 - 1.0.1 の開発はうまくいかなかった
 - 多くの piuparts ユーザには関係ない修正だった
 - モチベーションは依然下がったままである
- コントリビュートできること

- 21 個のバグが残っている
- python2、asciidoc の環境を新しくするべく Herbert Fortes さんが作業中
- テスト結果をよりよくまとめる
- web 画面のデザイン再考（スイートは 59 あるが、1 つのスイートしか表示できない）

6.3.3 Using Debian Containers for Cross-Development

- 組み込みデバイス向けのアプリケーション開発をするためのクロス開発環境の話
- 実演しながら説明するスタイル
- クロス開発環境がなぜ必要なのかの実演
 - arm-linux-gnueabi-hf-gcc でクロスコンパイルすれば ARMv7 なバイナリが作れる
 - ARMv7 ボードで GPIO の LED をチカチカさせるプログラムを例にクロスコンパイルしてみる
 - apt-get install package:armhf とアーキテクチャを指定してライブラリをインストールしコンパイル
 - ただ、ボード PC にそのままバイナリをコピーしてもライブラリが足りず動かない。ライブラリの依存関係解決は大変で「Dependency hell」の状態になる
- 異なる CPU アーキテクチャ向けのクロス開発するにはいくつか方法があることの説明
 - yocto project
 - chroot
 - docker 今回はこのやり方をデモ
- amd64 の PC 上で "arm32v7/debian:buster-slim" な docker image を作成し docker push
- ボード PC 上で docker pull し、docker run するデモを実施
- 質問では、ボード PC 側では disk や RAM をたくさん使うと思うがどれくらい使うのかというリソースの観点で質問があった

6.4 おわりに

DebConf19 のビデオを見て勉強してみました。セミナーのビデオはまだまだあるので、時間を見つけて見てみようと思います。

7 OSC 2019 Hokkaido のイベント参加報告

杉本典充

7.1 本レポートについて

北海道で Debian / Ubuntu ユーザーミートアップ in 札幌 2019.05、OSC 2019 Hokkaido のイベントを開催しました。本レポートは、北海道におけるイベント開催のレポートになります。

7.2 Debian / Ubuntu ユーザーミートアップ in 札幌 2019.05

7.2.1 イベント概要

- イベント名
 - Debian / Ubuntu ユーザーミートアップ in 札幌 2019.05
- イベントの web サイト
 - <https://debianjp.connpass.com/event/126637/>
- 日時
 - 2019-05-31 (金) 19:00-21:00
- 場所
 - 株式会社インフィニットループ様 会議室 (サッポロファクトリー 1 条館 3 階)

7.2.2 イベントの来場者

- 合計 3 名
 - 開催側
 - * 杉本さん、吉野さん
 - 参加者
 - * 北野大地@JI8GRX さん
 - ・ Debian 古参ユーザで、北海道 Linux ユーザーズクラブ (DoLUC)^{*37}のメンバ
 - ・ 2017 年、2018 年のミートアップの参加実績あり

7.2.3 セミナー

- 予定していたセミナーはキャンセル (参加者と話し合い、キャンセルとした)
- 北海道の Debian ユーザや北海道のコミュニティの現状を共有することにした
- キャンセルとなったセミナー
 - golang の ”net/http/fcgi” を使って REST API アプリを作る (発表者 : dictoss さん)
 - Debian と セキュアブート (発表者 : yy-y-ja.jp さん)

^{*37} 現在、DoLUC は活動を停止中。

7.2.4 情報交換した内容、出てきた意見

北海道の IT 勉強会の現状

- LOCAL がまとめた北海道における勉強会一覧の資料
 - <https://www.local.or.jp/wp-content/uploads/2019/05/Handbook-2019-19052903.pdf>
- 上記資料の勉強会はどのような勉強会であるか調べた
 - イベント系の団体
 - * オープンソースカンファレンス北海道実行委員会
 - * 一般社団法人 LOCAL
 - ・ OSC 北海道のイベント実務の大部分は LOCAL がやっている
 - * ET ロボコン北海道地区実行委員会
 - ・ 北海道の高専は高専ロボコン全国大会の強豪校
 - * U-16 プログラミングコンテスト
 - 学生が主な参加者の勉強会や団体
 - * LOCAL 学生部
 - ・ 属している学生の学校は北海道各地に散らばっている
 - ・ 【疑問】どうやって交流や話をしているのか?^{*38}
 - * FuraIT (富良野)・ゆるい勉強会 (旭川)
 - ・ 富良野と旭川の高校生・高専生が主な参加者に見える
 - ・ 参加者の一部の人は両方のイベントに参加している (生活圏内が近いと思われる)
 - * 子ども向け
 - ・ TEAM IchigoJam ほっかいどう
 - ・ CoderDojo 札幌 × 札幌東 × 恵庭

北海道の工学系の大学や高校

- 北海道大学
 - OSC 北海道で例年参加者を見かける
- 室蘭工業大学
 - OSC 北海道で例年参加者を見かける
- 北見工業大学
- 公立はこだて未来大学
 - OSC 北海道で例年参加者を見かける
- 公立千歳科学技術大学
 - OSC 北海道で例年参加者を見かける
- 札幌市内の私大の工学部系
- 国立高専 (旭川、函館、釧路、苫小牧)
 - OSC 北海道で例年参加者を見かける
- 富良野緑峰高校

企業

- アットマークテクノさん
 - Armadillo というボードをつくっており、OS は Debian を採用

今回の OSC 北海道の参加予定セミナーの申し込み数の傾向

- Microsoft のセミナーは人気が高い
- Python のセミナーは人気が高い
- VR ネット、AI ネットは人気が高い
- プログラミング教育系のセッションの人気が高い
- 勉強会そのものの体験談、カンファレンスの参加体験談に人が集まる

^{*38} 後述の OSC 2019 Hokkaido で学生に実際にヒアリングを行った。

どうやって大学や高校、学生と接触していくとよいか

- 【悩み】先生や学生に接触できればよいが、どうやって接触すればよいかわからない

7.3 オープンソースカンファレンス 2019 Hokkaido

7.3.1 イベント概要

- イベント名
 - － オープンソースカンファレンス 2019 Hokkaido
- イベントの web サイト
 - － <https://www.ospn.jp/osc2019-do/>
- 日時
 - － 2019-06-01 (土) 10:00-18:00
 - * DebianJP では、2 日目のコミュニティデイのみ参加
 - * ブース展示、セミナーを開催
- 場所
 - － 札幌コンベンションセンター (地下鉄東西線 東札幌駅近く)

7.3.2 イベントの来場者

- 2 日間合計で約 720 名程度 (主催者発表値)^{*39*40}
 - － 5/31 (金) 約 80 名 (ビジネスデイ)
 - － 6/01 (土) 約 640 名 (コミュニティデイ)
 - － 2 日間合計 約 720 名
- 学生の参加者比率が高い
 - － 高校生、高専生、大学生が参加している
 - － 交流した学生で学校が判明しているところ
 - * 公立千歳科学技術大学
 - * 公立はこだて未来大学
 - * 室蘭工業大学
 - * 北海道大学
 - * 苫小牧高専
 - * 旭川高専
 - * 旭川工業高校

7.3.3 セミナー

- 公開プログラム
 - － <https://www.ospn.jp/osc2019-do/modules/eventtrsv/?id=2&noform=1>
 - － 2019/6/1 (土) 11:00-11:45
- 発表内容
 - － タイトル「Debian Updates」
 - － 発表者は杉本さん
 - － スライド資料^{*41}
- 聴講者
 - － 6 名

7.3.4 ブース展示

- 2019/6/1 (土) のみ展示
 - － 吉野さん、杉本さんで対応
- 出展内容
 - － Debian GNU/Linux マシンのデモ展示

^{*39} 出典：メーリングリスト osc-member [osc:7983] [北海道] アンケート集計結果

^{*40} 過去データ一覧 <https://www.ospn.jp/visitors/>

^{*41} <https://tokyodebian-team.pages.debian.net/pdf2019/debianmeetingresume201906-osc2019do-presentation.pdf>

-

図 10 展示ブースの様子 (OSC 2019 Hokkaido)

- プース来訪者
 - ー 合計 約 61 名
 - ー 属性
 - * 男性・大人 30 名
 - * 女性・大人 5 名
 - * 男性・学生子供 25 名
 - * 女性・学生子供 1 名
- 物販
 - ー 本：あんどきゅめんとつどでびあん^{*42}
 - * 3 冊販売
 - ー Debian T シャツ
 - * 4 着販売

7.3.5 ブース来訪者との情報交換

- 利用 OS のアンケート
 - Ubuntu 10
 - Debian 7
 - Raspbian 4
 - ChromeOS 1
 - EV3 Debian 1
 - CentOS 1

^{*42} <https://tokyodebian-team.pages.debian.net/undocumenteddebian.html>

7.3.6 イベントの参加者や出展者との交流

- LOCAL 学生部
 - － メンバは北海道各地に散らばっている
 - － 主に slack でコミュニケーションをとっている
 - － Face to Face で会うのは OSC とそれ以外に年に 1 回程度
 - － 「LOCAL Students 情報ボーイズの寄稿ノート」という本を出している^{*43*44*45}
- 深町先生
 - － 公立千歳科学技術大学の先生
 - － FML の開発者

7.4 今後の Debian 開発者候補の方への呼びかけ方のディスカッション

7.4.1 観点

会いに行く・会いに来る

- 東京圏におけるイベントと地方におけるイベントの傾向の違い
- 高校生・高専生・専門学校生・大学生におけるイベントや勉強会の参加の壁は何か（時間、金銭、距離など）
- 学生のいる場所へ呼んでもらう方法
- 先生とどう知り合い、どのような情報交換をしていくとよいか

Debian 勉強会における情報提供の在り方

- 取り上げる話題をどう選んでいくか
 - － 時事ネタ、流行りネタ、利用事例
 - － インストール
 - － アプリケーション、ミドルウェアの利用
 - － 開発環境
 - － Debian の仕組み解説
- Debian 勉強会の資料をどのようなメディアで提供するとよいか
- Debian 勉強会のセミナーのビデオ配信（ライブまたは録画）の是非

インターネットにおける情報提供

- メーリングリスト
- ブログ記事
- web サイト

イベントやセミナーにおける情報提供

- セミナーのテーマに取り上げる話題の選定
 - － OSC では「Debian Update」という半年間を振り返る話をする事が多い。見直すべきか？
- 魅力あるブース展示にするための検討
- イベントの参加者属性を意識する
 - － OSC は入門者向けのイベントのため、セミナーは入門レベルの難易度に設定している
 - － より高度な知識を求める人はどのようなイベントに集まるか？また、そのイベントに Debian 勉強会はどう関わるか？

^{*43} <https://techbookfest.org/event/tbf04/circle/14720002>

^{*44} 原稿は tex で書いている模様。 https://github.com/hyoiutu/techbookfest_localstudents_2017

^{*45} 2 冊目にあたる「LOCAL Students 情報ボーイズの寄稿ノート 2.0」を当日販売していた。

8 東京エリア Debian 勉強会課題整理

第 175 回東京エリア Debian 勉強会参加者

東京エリア Debian 勉強会参加者でディスカッション

8.1 Debian 勉強会や Debian 界限と OSS の関わる人とのかかわり

東京と地方のイベントの開催傾向

- 東京はユーザが非常に多い- OSC はユーザ向けイベントの傾向あり
- 地方はユーザも開発者も集まる

8.2 Debian を使うきっかけ

Debian を触るきっかけ

- プログラミングコンテストに参加した

Debian 勉強会に参加したきっかけ

- twitter を見て知った (henrich さん)
- 有名な人と話がしたい、聴きたい

8.3 世界での開発者同士の交流状況

DebConf18 で聞いたこと

- 東京は毎月 FaceToFace で集まっていると話したところ、海外の DD からは羨ましいという意見があった
- 海外の DD の人口密度が低いいため、ビデオ会議やメールでやりとりするしかない環境の人は多い

8.4 Debian 開発者の育成

パッケージ作成ハンズオンセミナーをやってみたらどうか

- dh の使い方
- Debian Policy の理解講座

Debian 開発者になると何がうれしいのかを他の人たちは知らないのではないか

- Debian 開発者の権限
 - ー 自分が普段使っているソフトをパッケージにアップロードできる
 - ー porterbox が使える (移植のため)
- Debian 開発者のおまけな特典
 - ー <https://wiki.debian.org/MemberBenefits>
 - ー 例
 - * LWN 無料で読める
 - * Steam の「portal」というゲームが一部無料
 - * gandi.net でドメインが安くとれる
 - * 就職の役に立つ?

8.5 Debian 勉強会とイベント参加の在り方

参加の壁（時間、金銭、距離、心情の問題）

- 開催回数が多すぎて初めて参加する人から見ると心理的ハードルが高い模様
- 遠くからの参加者を DebianJP で支援できないか
 - － 過去に支援していたことがあった気がする
 - － イベントで話す人に対して金銭支援してきてもらう
- ライフステージによって勉強会やイベントに来れなくなる
 - － 仕方がない面がある
 - － 若い人に対してリーチする必要あり

学生への呼び込み

- 学生のいるところに呼んでもらって Debian 勉強会を開催する
 - － 呼んでもらうにも大学生とのつながりが必要
 - － どうやるとつながることができるかのいい案が浮かばない
- Summer Of Code を DebianJP や日本でできないのか
 - － 成果を OSC などのイベントで発表してもらうこともできそう

オンラインで勉強会に参加できるようにする

- Debian 勉強会を配信してみたらどうか
 - － どれくらい人が見てくれるかはやってみないとわからない
 - － 初めて参加する人は質問しにくい気もする
 - － wifi 経由で主催者の画面をインターネットに共有しておいて、セミナー配信する形で始めてみてはどうか

debian 勉強会で取り上げるネタ

- 深い話の解説話- 例
 - － init の起動の処理を読み解く
 - － AWS や GCP でのブート処理は PC とどう違うのか
 - － debian は中までソースコードを読めばわかる
 - － 例のような深い話はなかなか解説もないため、やってみるといいかもしれない
- ネタの回し方
 - － OSS の記事が載っている雑誌
 - － 雑誌と同じ回し方を勉強会ですると内容が被るのよくないかもしれない
 - － debian には debian 開発の入門の流れがあるように思える
- ハンズオン- パッケージング道場

勉強会資料の公開の仕方

- 現状は PDF の資料を公開している
- google の検索でも PDF の記事はヒットする
- HTML で出せればいいが、ひとまず PDF の現状でもよいのではないか
- まずは情報を出していくことが重要

web の露出

- 英語で検索したほうが記事のヒットが多い
- debian で検索しても出てこないが、ubuntu の記事が検索すると出てくる- ask ubuntu で検索がヒットすることが多い
- Arch Linux の wiki の記事の情報量はすごい
- kernel などのビルドまわりは Gentoo Wiki が強い

メディア

- Software Design の記事執筆（やまねさん）- 大変がんばっていると思う

イベント参加

- ODC
 - － 2018 年にも開催されている
 - － 出たことがないため雰囲気は不明
 - － コミュニティもセミナーは出せるため出してみたらどうか
- Developers Summit
 - － 2018 年は成果を見せる場になっているように感じた
 - － 開発者のガチ発表という感じでなくなっている

8.6 CJK 開発ネタ

日本語環境

- 日本語入力環境をどう維持するか
 - － DebianJP でも維持する人/できる人が限られている
 - － 他のディストリビューションも人手不足の問題を抱えている
 - － 新規で作業する人/作業できる人が入ってこない状況がある
 - － 人手不足の解消案としてディストリビューション間での連携も考えた方がいいかもしれない
- mozc - 比較的新しい IM だが開発が止まっている
- authy
 - － FSIJ が引き受けている状況
 - － こう開発したいというリストは出ている
 - － 辞書は職人芸によりスコアリングされたものになっており難解
- skk - 日本で使っている人はいたはず
- 辞書データ
 - － AI 処理で利用するため整備の必要あり
 - － 辞書のライセンスの考え方
 - － 辞書づくりの作業自体がフリーであるべき派
 - － 商用版のみでも辞書があれば問題ないとする派

海外の IM 事情

- 中国語- 入力方法は 2 種類
 - － 部首
 - － 発音 (こちらが多い)
- IM ソフト
 - － <https://rime.im/> 個人で開発したことから始まる
 - － fcitx (UIM、ibus と似た機能のソフト)

8.7 リモート会議を行うツール

mumble

- <https://packages.debian.org/ja/stretch/mumble-server>
- VoIP サーバのソフトで、音声のみ
- ライセンスは DFSG 準拠
- 会話するサーバとクライアントソフト

WebRTC を使うソフト

- DFSG 準拠のソフトはあるのか

nageru

- <https://packages.debian.org/sid/video/nageru>
- 利用実績はあるか

discord

- 画面の共有がしやすい
- インフラ勉強会でこれを使って開催しているらしい

slack もビデオもあったと思う

- やりづらい

youtube

- コメントが下に出てくるため見にくい、主催者に伝わりづらい

Mini Hangouts

- 質疑応答のログがチャットを切ると消える

spread-webrtc

- <https://github.com/strukturag/spreed-webrtc>

8.8 ディスカッションを終えての感想

今後 Debian 勉強会をどうしていくか振り返る時間をここ数年とっていなかったの、今回はよい機会だった

本資料のライセンスについて

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

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

ソースコードについて

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

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

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

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

Preamble

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

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

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

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

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

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

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

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

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

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

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

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

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

2. You may modify your copy or copies of the Program or any portion

of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

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

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

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

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

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

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

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

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

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

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

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

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

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

5. You are not required to accept this License, since you have not

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

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

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, 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.

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

——『あんどきゅめんとっど 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 2019/07～2019/11 まで内容は無保証、つっこみなどがあれば勉強会にて。



あんどきゅめんとっど でびあん 2019 年冬号

2019 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）