



Grand Unified Debian



銀河系唯一のDebian専門誌

東京エリア/関西Debian勉強会



あんどきゅめんでっど でびあん 2024 年夏号 2024 年 8 月 12 日 初版発行

今年勉強の目標

目次

1	Introduction	2
2	RISC-V ボード VisionFive2 に Debian を入れてみた話	3
3	Debian を ConoHa VPS (v3) にインストールしてみた	10
4	xz-utils のバックドア問題の情報交換	13
5	Debian における 64bit time.t 移行について	16
6	python の仮想環境管理 venv と pipx	19
7	2023 年を振り返って	24
8	2024 年の目標	27

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 界隈で起こった出来事、などなど) について話し合う会です。

目的として次の三つを考えています。

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

それでは、楽しい一時をお楽しみ下さい。

2 RISC-V ボード VisionFive2 に Debian を入れてみた話

Norimitsu SUGIMOTO (杉本 典充)



2.1 自己紹介

- Norimitsu SUGIMOTO (杉本 典充)
- dictoss@live.jp
- Twitter: @dictoss
- Debian を使い始めたのは 3.1 sarge が testing の頃
- 仕事はソフトウェア開発者をやってます
- python と Django の組み合わせで使うことが多いです

2.2 RISC-V とは

- 2010 年頃にカリフォルニア大学バークレー校で設計された RISC 型 CPU の ISA (命令セット)
- オープンソースライセンスで提供されている*1
 - Creative Commons Attribution 4.0 International Public License
- 歴史的に RISC-I、II、III、IV、V とバージョンアップしてきた、というものではない
 - RISC-V はこれまでの経緯と課題を考慮した結果、スクラッチで開発したものらしい

2.3 Debian と RISC-V

- Debian Wiki - RISC-V <https://wiki.debian.org/RISC-V>
- Debian Ports <https://www.ports.debian.org/>
- 時系列
 - 2018-05-24 Debian Ports に riscv64 が追加
 - 2023-07-23 Debian Ports の riscv64 が unstable に移動*2

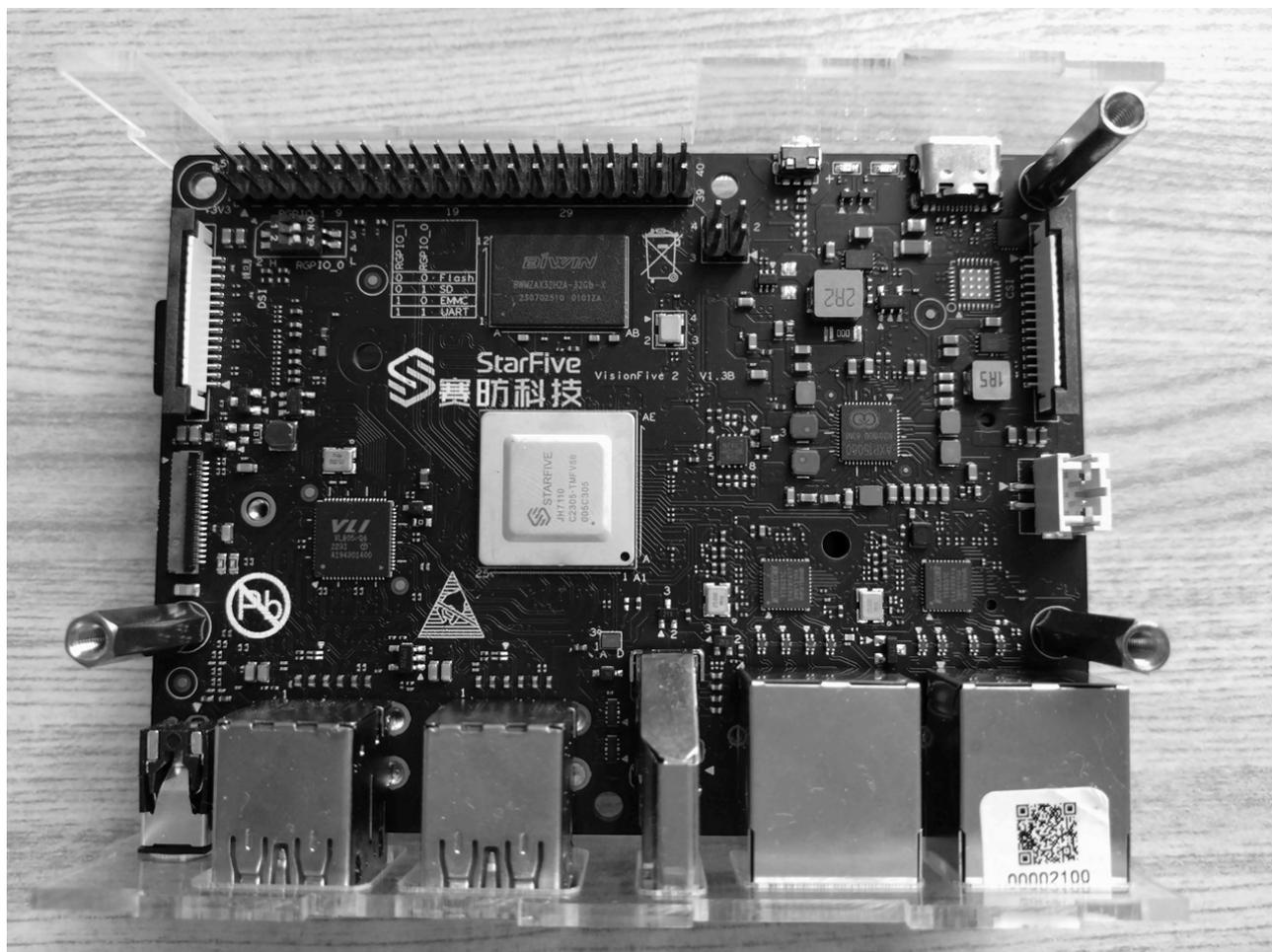
2.4 RISC-V ボード PC 「VisionFive2」の紹介

- 中国 StarFive 社が製造する RISC-V CPU を搭載するシングルボードコンピュータ (SBC)

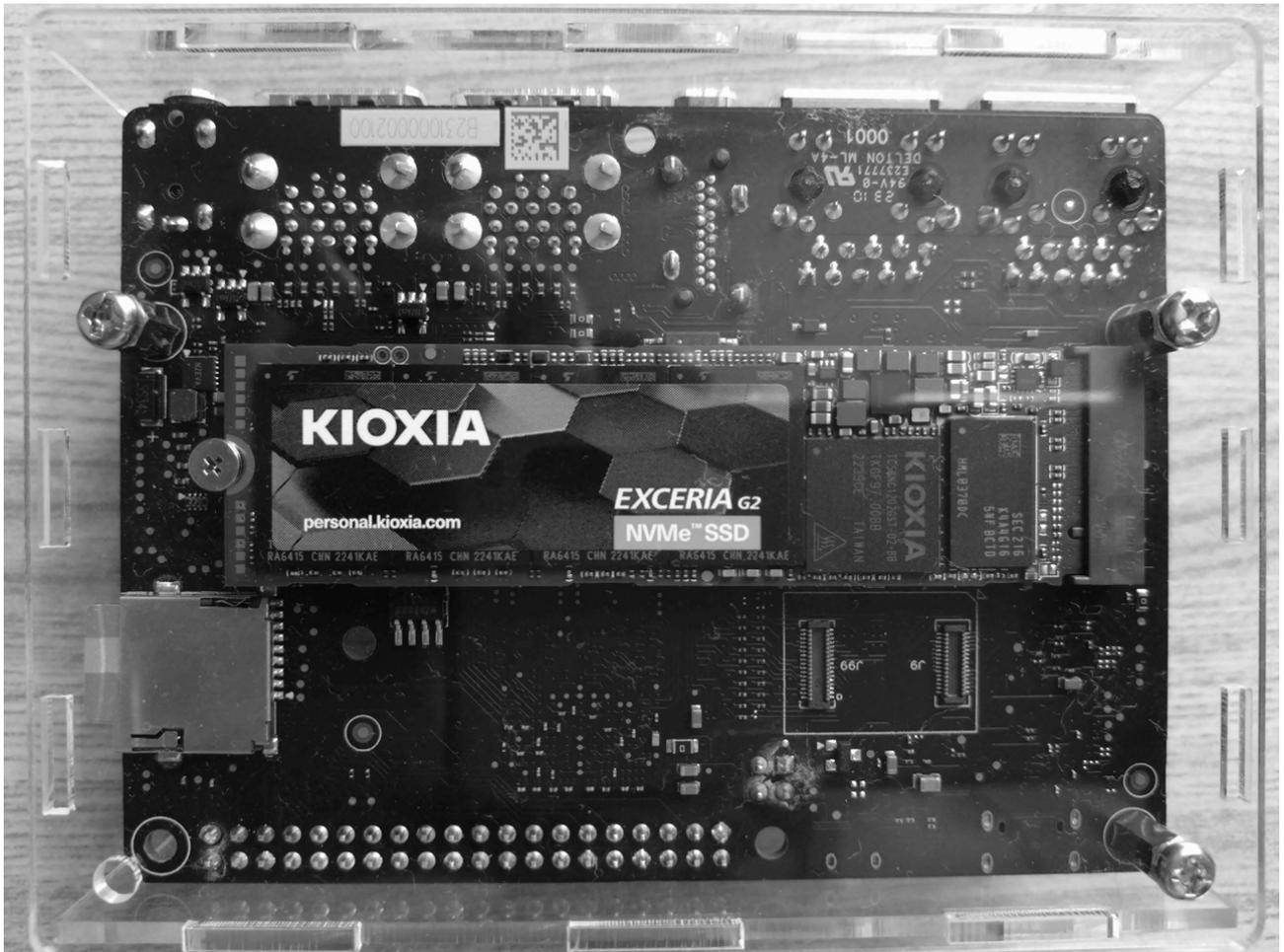
*1 <https://github.com/riscv/riscv-isa-manual>

*2 <https://lists.debian.org/debian-riscv/2023/07/msg00053.html>

- <https://www.starfivetechnology.com/en/site/boards>
- kickstarter.com のクラウドファンディングで寄付を募って製造・販売された*3
 - 2022-09-21 に寄付が締め切られ、早い人では 2022 年 12 月に入手していた人がいる模様
- ドキュメント https://doc-en.rvspace.org/Doc_Center/visionfive_2.html
- クイックスタートガイド https://doc-en.rvspace.org/VisionFive2/PDF/VisionFive2_QSG.pdf



*3 <https://www.kickstarter.com/projects/starfive/visionfive-2?lang=ja>



- CPU : StarFive JH7110 64bit SoC RV64GC, up to 1.5GHz クアッドコア
- GPU : IMG BXE-4-32 MC1 3D GPU 600 MHz (端子は HDMI 2.0、MIPI-DSI)
- メモリ : LPDDR4 SDRAM, up to 2,800 Mbps (2GB/4GB/8GB モデルの 3 種類)
- ストレージ : MicroSDHC/SDXC、eMMC、NVMe SSD (PCIe-2.0 x1)
- ネットワーク : RJ-45 GbE 2 ポート
- 電源 : USB PD 30W (USB Type-C 端子)、GPIO 5V、PoE (要オプション)
- その他 : GPIO 端子 (コンソール等)、USB 3.0 x 4 ポート

- SDK <https://github.com/starfive-tech/VisionFive2>
 - u-boot-spl.bin.normal.out : SPL
 - visionfive2_fw_payload.img : U-Boot イメージ
 - sdcard.img : SPL と U-Boot の更新用起動イメージ (buildroot)
- Debian イメージ https://doc-en.rvspace.org/VisionFive2/Quick_Start_Guide/VisionFive2_QSG/flashing_with_mac_linux.html

- Purchase Links <https://forum.rvspace.org/t/how-to-purchase-visionfive-2/665>
 - 「Japan」と書いてあるリンクをクリックすると Amazon.co.jp の販売ページへジャンプする
 - 販売元は WayPonDEV-jp
 - 本体 4GB (無線 LAN ドングルなし)、アクリルケースを購入
 - 自分は 2023-10-25 に購入し、2023-10-31 に届いた
 - 梱包を開けて製品を確認するとシールが 2 枚重ねて貼ってあった。上のシールをはがすと下のシールに「Early Bird」の記載あり

2.5 VisionFive2 のファームウェアの更新方法

2.5.1 SPL、U-boot の更新方法

アップデート方法は 2 種類ある

- Updating SPL and U-Boot of Flash^{*4}
 - tftpboot コマンド (SD カードなしで実行可能)
 - flashcp コマンド (VF2_v2.5.0 以降の buildroot 環境で実行可能)
- 今回入手したボードは初期ファームウェアの模様で、2 段階で更新した
 - SD カードなしでブートし、tftpboot コマンドで v2.11.5 へ更新
 - v2.11.5 の sdcard.img でブートし、flashcp で v3.7.5 へ更新^{*5}

2.5.2 シリアルコンソールへの接続

PC とシリアルコンソールで接続する

- GPIO からシリアルコンソールを取るピンアサイン
 - https://doc-en.rvspace.org/VisionFive2/Quick_Start_Guide/VisionFive2_SDK_QSG/recovering_bootloader%20-%20vf2.html
- pin-6 GND、pin-8 TX、pin-10 RX
- Raspberry Pi で使っている USB シリアル変換ケーブルが使えます

2.5.3 TFTP サーバの準備

LAN 内に TFTP サーバを準備

```
# apt-get update
# apt-get install xinetd tftpd
# mkdir /var/lib/tftpboot
# chmod 777 /var/lib/tftpboot
# vi /etc/xinetd.d/tftpd
service tftp
{
    disable = no
    log_type = File /var/log/tftpd
    log_on_success = HOST
    log_on_failure = HOST
    socket_type = dgram
    protocol = udp
    wait = yes
    user = root
    server = /usr/sbin/in.tftpd
    server_args = -s /var/lib/tftpboot
    per_source = 11
    cps = 100 2
    flags = IPv4
}
# systemctl restart xinetd
```

2.5.4 TFTP 経由の SPL、U-boot の更新

microSD カードは差さずに起動する

- 起動前の準備
 - LAN ケーブルを差す (2 ポートの内、自分は外側を使用)
 - GPIO に USB シリアル変換ケーブルを差し、PC と接続
- 電源 ON

^{*4} https://doc-en.rvspace.org/VisionFive2/Quick_Start_Guide/VisionFive2_SDK_QSG/spl_new.html

^{*5} 2023-12-14 時点の最新は v3.9.3。

IP アドレスを設定し、LAN へ接続

```
StarFive # setenv ipaddr 192.168.1.230
StarFive # setenv serverip 192.168.1.1
StarFive # setenv gatewayip 192.168.1.1
```

`${serverip}` は `setenv` で指定した値が展開されるため、そのまま実行すること

```
StarFive # ping ${serverip}
host 192.168.1.1 is alive
```

SPI Flash を初期化

```
StarFive # sf probe
SF: Detected gd25lq128 with page size 256 Bytes,
erase size 4 KiB, total 16 MiB
```

SPL をダウンロードし、更新

```
StarFive # tftpboot 0xa0000000 ${serverip}:u-boot-spl.bin.normal.out
StarFive # sf update 0xa0000000 0x0 $filesize
```

U-Boot をダウンロードし、更新

```
StarFive # tftpboot 0xa0000000 ${serverip}:visionfive2_fw_payload.img
StarFive # sf update 0xa0000000 0x100000 $filesize
```

リセットします

```
StarFive # reset
```

2.5.5 Buildroot 環境での起動

microSD カードへ `sdcard.img` を `dd` して起動イメージを作成

- https://github.com/starfive-tech/VisionFive2/releases/tag/VF2_v2.11.5

```
# dd if=sdcard.img of=/dev/sdb bs=1M status=progress
```

microSD カードを VisionFive2 にセットして起動すると「Buildroot」という linux 環境が起動する

```
Welcome to Buildroot
buildroot login: root
Password: starfive

# version
Linux version 5.15.0 (jenkins@soft05) (riscv64-buildroot-linux-gnu-gcc.
br_real(Buildroot VF2_v2.11.5) 10.3.0, GNU ld (GNU Binutils) 2.36.1)
#1 SMP Fri Mar 24 01:42:57 CST 2023
VF2_v2.11.5
```

2.5.6 flashcp を使った SPL、U-boot の更新

v3 系以上のバージョンの SPL、U-boot へ更新する

VisionFive2 の IP アドレスを確認しておく

```
# ip a
```

Buildroot 環境は `busybox` 環境で `scp` のサーバ側にはなれないため、別サーバでダウンロードしておいた SPL、U-boot ファイルを `scp` で取得する

```
LAN 内のサーバ
# wget https://github.com/starfive-tech/VisionFive2/releases/download/
VF2_v3.7.5/visionfive2_fw_payload.img
# wget https://github.com/starfive-tech/VisionFive2/releases/download/
VF2_v3.7.5/u-boot-spl.bin.normal.out
```

```
VisionFive2
# scp username@192.168.xxx.xxx:visionfive2_fw_payload.img /root/
# scp username@192.168.xxx.xxx:u-boot-spl.bin.normal.out /root/
```

インストールするデバイス名を確認

```
# cat /proc/mtd
dev:      size  erasesize  name
mtd0: 00040000 00001000 "spl"
mtd1: 00300000 00001000 "uboot"
mtd2: 00100000 00001000 "data"
```

flashcp コマンドで更新

```
VisionFive2
# flashcp -v u-boot-spl.bin.normal.out /dev/mtd0
# flashcp -v visionfive2_fw_payload.img /dev/mtd1
# reboot
```

2.6 VisionFive2 へ Debian をインストールする

2.6.1 VisionFive2 の Debian イメージの準備

microSD カードへ Debian のイメージを dd して起動イメージを作成

- 公式が Debian イメージを配布している*6
- イメージは、sd / emmc / nvme がある
- 2023/08 の sd イメージを使ってみました
starfive-jh7110-202308-SD-minimal-desktop.img.bz2

```
# bzip2 -d starfive-jh7110-202308-SD-minimal-desktop.img.bz2
# dd if=starfive-jh7110-202308-SD-minimal-desktop.img \
of=/dev/sdx bs=1M status=progress
```

microSD カードを VisionFive2 に差して起動すると、Debian unstable が起動する

2.6.2 VisionFive2 の Debian イメージの起動

ログインは root ユーザ、パスワード「starfive」で行える

kernel は linux-5.15

```
root@starfive:~# uname -a
Linux starfive 5.15.0-starfive #1 SMP Wed Aug 23 11:18:20 CST
2023 riscv64 GNU/Linux
```

apt-line が debian ports になっているため、unstable に書き換えて apt で更新する

```
# vi /etc/apt/sources.list
deb https://snapshot.debian.org/archive/debian-ports/20221225T084846Z
unstable main

deb http://deb.debian.org/debian/ unstable main non-free-firmware
deb-src http://deb.debian.org/debian/ unstable main non-free-firmware

# apt-get update
# apt-get dist-upgrade
# reboot
```

2.7 VisionFive2 の NVMe SSD について

2.7.1 VisionFive2 の NVMe SSD の性能

- XiOXIA EXCERIA G2 500GB を差した
– Read 2,100 MB/s

*6 https://doc-en.rvspace.org/VisionFive2/Quick_Start_Guide/VisionFive2_QSG/flashing_with_mac_linux.html

– Write 1,700 MB/s

- microSD カードの debian イメージで起動すると、NVMe SSD を認識している (/dev/nvme0n1)

/dev/nvme0n1 を fdisk し、mkfs.ext4 でフォーマットしてベンチマーク

```
# mkdir /mnt/ssd
# mount /dev/nvme0n1p1 /mnt/ssd
# dd if=/dev/zero of=/mnt/ssd/bench.dat bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 6.30584 s, 170 MB/s
```

NVMe SSD を root ディスクにするのは試せていません

2.8 まとめ

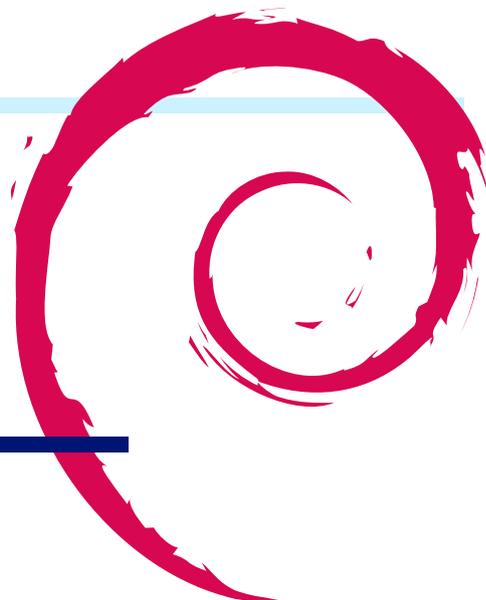
- クイックスタートガイドを読むとかなり部分が理解できるため、読みましょう
- SPL、U-boot の更新は Buildroot のイメージで起動して行うのがよい
- debian イメージで起動するところまでできれば、あとは PC と同じ
- NVMe SSD の性能は microSD カードの性能と比べればかなり性能はよいが、PC レベルの性能は出ない

2.9 参考資料

- Ubuntu Weekly Recipe - 第 752 回 RISC-V のシングルボードコンピューターである VisionFive 2 を使ってみる
<https://gihyo.jp/admin/serial/01/ubuntu-recipe/0752>
- VisionFive 2 Datasheet https://doc-en.rvspace.org/Doc_Center/visionfive_2.html
- github.com - VisionFive2 <https://github.com/starfive-tech/VisionFive2>

3 Debian を ConoHa VPS (v3) にインストールしてみた

Kentaro HAYASHI (林健太郎)



本記事は Debian 勉強会 (2024 年 1 月開催) にて公開した資料を冊子向けに編纂・収録したものです。なお、2024 年 5 月 23 の ConoHa のプレスリリースにて言及があったように、ConoHa VPS (v3) では Debian が標準テンプレートに含まれるようになりました。本記事で扱っているようなカスタム ISO を作成する手順は現在では不要になっています。

*7

3.1 ConoHa VPS (v3) とは

GMO グループ傘下のホスティングサービスです。VPS の時間課金での利用もできるのが特徴のひとつです。Debian だけでなく、BSD 系も含めて幅広い OS が標準テンプレートとして提供されています。

ただ、インフラの刷新により、従来のインフラである v2 では Debian が標準で含まれていましたが、新しいインフラである v3 では Debian が除外されてしまいました。Debian を使いたいならあえて v2 のインフラでインスタンスを立ち上げる必要があります。^{*8}

3.2 ConoHa VPS (v3) でカスタムインストールする

前述の状況ではありませんが、ConoHa VPS (v3) ではカスタムインストールするという手段が提供されています。「API で VPS に ISO イメージを挿入する」ためのドキュメントというのがまさにそれです。^{*9} これを利用すれば、任意の ISO を利用してインストールできます。

例えば、次のような手順で ISO をマウントし、インストール手順を進められます。

- インスタンスを作成後停止しておく
- トークンを作成 (API アクセスに必要)
- イメージ ID を作成 (あとで ISO をアップロードするときに使う)
- ISO イメージ (bookworm netinst .iso) をアップロード
- 停止済みインスタンスに ISO をマウント
- レスキューモードで ISO イメージからインストーラーを起動
- コンソール URI を発行して操作

^{*7} https://www.conoha.jp/vps/news/?btn_id=vps-top--news_vps-news&ap=2015053467

^{*8} Debian にこだわらず Ubuntu の標準イメージを選択するという手もありますが、Ubuntu の場合 512MB メモリのインスタンスを立ち上げることができません。最低スペックがメモリ 1GB からとなるので月額費用が変わってきます。

^{*9} https://doc.conoha.jp/api-vps3/api-mount_iso_image-v3/

3.3 Debian 公式 ISO イメージを利用した場合の課題

任意の ISO を利用して、インストーラーを起動すれば Debian をインストールできます。しかし、公式 ISO を利用してインストールした場合、次のような課題があります。

- セキュアなインスタンスの初期構築が手間
 - ConoHa VPS で提供されている標準イメージのように、あらかじめ `.ssh/authorized_keys` を仕込めない
- ConoHa 標準の VNC コンソールがちょっと不便
 - テキスト送信や特殊キー送信経由ではパイプや `_` などの記号が打てず欠落する
 - vim で直接いじるにも操作感が通常と異なる

後者はともかく前者はパスワード認証したくない人にとってはやや致命的です。これについてはあらかじめカスタマイズした ISO を利用すれば解決できます。

3.4 カスタム ISO イメージを作成する方法

カスタム ISO イメージを作成する方法はいくつかあります。FAI(Fully Automatic Installer) とか Simple-CDD とか `preseed.cfg` などです。

今回は過去の Debian 勉強会 (2016 年 10 月の Debian 勉強会資料参照 ^{*10}) でも紹介されていた `preseed.cfg` によるカスタム ISO イメージ作成方法を紹介します。

`preseed.cfg` というのは、Debian においてインストールを自動化する方法のひとつです。`preseed.cfg` を参照させる方法はいくつかあります。

- `initrd.gz` に仕込む (CD イメージ再生成必要)
- 起動時パラメーターで `preseed.cfg` を参照させる
 - `file=...` で CD イメージに含めた `preseed.cfg` を参照する (CD イメージ再生成必要)
 - `url=` でネットワーク上の `preseed.cfg` を参照させる

このなかでは、VNC コンソールへの切り替えに時間がかかっている最中に誤爆ないように `preseed.cfg` を CD イメージに含めるのがよいでしょう。

`preseed.cfg` を CD イメージに含める場合、インストーラー自体にも若干手を加える必要があります。具体的には次のような変更が必要です。

- `isolinux/menu.cfg`
 - 搭載メモリが少ないインスタンス (例えば 512MB のメモリ) では GUI インストールできないので、Graphical Install を最初から削除する (`gtk.cfg` をコメントアウト)
- `isolinux/txt.cfg`
 - テキストモードをデフォルトにする
 - `file=/cdrom/preseed/preseed.cfg` を参照させる
 - 言語やロケール、キーマップを指定する

なお、後述の説明では Debian のインストール CD の内容を `custom-iso` ディレクトリにコピーしてコンテンツを編集するものとします。`preseed.cfg` は `preseed/preseed.cfg` としてファイルを配置するものとします。

^{*10} <https://tokyodebian-team.pages.debian.net/pdf2016/debianmeetingresume201610-presentation-sugimoto.pdf>

3.4.1 isolinux/menu.cfg の変更

```
menu title Debian GNU/Linux installer menu (BIOS mode)
include stdmenu.cfg
#include gtk.cfg
include txt.cfg
menu begin advanced
...
```

上記のように、グラフィカルインストールの選択肢をコメントアウトして無効化します。

3.4.2 isolinux/txt.cfg の変更

```
label install
menu label ^Install
menu default
kernel /install.amd/vmlinuz
append language=en country=US keymap=us \
file=/cdrom/preseed/preseed.cfg vga=788 initrd=/install.amd/initrd.gz --- quiet
```

file=で preseed.cfg を指定し、テキストモードを既定 (menu default) にしておきます。

3.4.3 preseed.cfg で公開鍵を仕込む

```
# Setup public key
d-i preseed/late_command string \
cp /cdrom/preseed/sshd_config.d.local /target/etc/ssh/sshd_config.d/local.conf ; \
mkdir -p /target/home/debian/.ssh ; \
cp /cdrom/preseed/authorized_keys /target/home/debian/.ssh/ ; \
in-target chown debian:debian -R /home/debian/.ssh ; \
in-target chmod 400 /home/debian/.ssh/authorized_keys
```

公開鍵を使って ssh できるようにするため、あらかじめ authorized_keys を仕込んでおきます。既定のユーザーは debian にしてある前提です。必要に応じて適宜読み替えてください。preseed/late_command でファイルを配置するなどもろもろ実施する^{*11}のがポイントです。

3.4.4 ISO イメージを再作成する

```
(cd custom-iso; \
find -follow -type f ! -name md5sum.txt -print0 | xargs -0 md5sum > md5sum.txt)
sudo genisoimage -quiet -r -J -b isolinux/isolinux.bin \
-c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table \
-o netinst.iso custom-iso
```

CD に追加でファイルを仕込むので、チェックサムを更新して再作成します。あとは、このようにして作成したカスタム ISO(netinst.iso) をマウントすれば自動的にインストールを走らせることができます。

3.5 まとめ

- ConoHa VPS (v3) ではカスタムインストールができます
- preseed.cfg を活用すると初期構築の時点から自由にカスタマイズできます

今回は、ConoHa VPS (v3) でカスタム ISO を用いたインストール方法を説明しました。Debian が標準イメージとして提供されるようになったのでその必要性は薄れましたが、preseed.cfg を利用した自動インストールの知見はどこかで役に立つかもしれません。

^{*11} in-target を指定することで、chroot した状態でコマンドを実行できます。

4 xz-utils のバックドア問題の情報交換

2024 年 4 月の参加者



4.1 参加者

参加者は以下の 8 名でした。(敬称略)

- dictoss
- kenhys
- knok
- su_do
- yy-y-ja-jp
- znz
- sanadan
- YukiharuYABUKI

4.2 参考情報

- XZ Utils に悪意のあるコードが挿入された問題 (CVE-2024-3094) について
 - <https://www.jpCERT.or.jp/newsflash/2024040101.html>
 - Debian の情報
 - * <https://security-tracker.debian.org/tracker/CVE-2024-3094>
 - * <https://lists.debian.org/debian-security-announce/2024/msg00057.html>
- xz-utils パッケージ (Debian Package Tracking)
 - <https://packages.qa.debian.org/x/xz-utils.html>
 - 2024-04-20 現在 (stable は bookworm)
 - * stable 5.4.1-0.2 (影響なし)
 - * testing 5.6.1+really5.4.5-1 (影響あり、バージョンを巻き戻し)
 - * unstable 5.6.1+really5.4.5-1 (影響あり、バージョンを巻き戻し)
 - * Ubuntu 5.6.1+really5.4.5-1 (影響あり、バージョンを巻き戻し)
- 開発中の Ubuntu 24.04 LTS にも影響があり、ベータ版のリリースが 1 週間遅れた
 - <https://gihyo.jp/admin/clip/01/ubuntu-topics/202404/05>
- xz backdoor (2024-03-29)
 - <https://lists.debian.org/debian-devel/2024/03/msg00333.html>

- 2024-02-28 に unstable で really5.4.5 へ巻き戻し
 - * <https://tracker.debian.org/news/1515519/accepted-xz-utils-561really545-1-source-into-unstable/>
- XZ Utils にバックドア攻撃が行われるまでのタイムラインまとめ (2024-04-03 の記事)
 - <https://gigazine.net/news/20240403-timeline-of-xz-open-source-attack/>
 - Debian unstable に問題のバージョンが混入したのは、2024-02-26
 - * 混入直前:<https://snapshot.debian.org/archive/debian/20240225T151757Z/pool/main/x/xz-utils/>
 - * 5.6.0 混入直後:<https://snapshot.debian.org/archive/debian/20240226T213049Z/pool/main/x/xz-utils/>
 - * really 版リリース直後:<https://snapshot.debian.org/archive/debian/20240328T211728Z/pool/main/x/xz-utils/>
- 分析
 - <https://research.swtch.com/xz-script> 仕込まれたスクリプトの解析
 - <https://boehs.org/node/everything-i-know-about-the-xz-backdoor>
 - * JiaT75 が xz 以前にも bsdtar に悪意あるコードを PR していた

4.3 参加者の意見

- どういう脆弱性だったのか？
 - xz-utils ライブラリのパッケージが脆弱性を生み出しているが、このコードは ssh のログイン認証を迂回するような細工を行うものだった
 - 問題となったコードは m4 マクロだった
 - * 読める人はあまりいない気がする
- debian.org の unstable で今回戻したバージョンは、疑いがある人のコミットが 1 つもないところまで戻している
 - このメンテナーは信用できるのか？という感じで見られている
 - https://joeyh.name/blog/entry/reflections_on_distrusting_xz/
 - * joeyh さんがそうすべきとした理由
- debian のバグレポートでもソーシャルハッキングのような感じで、バグ入りのバージョンを入れてほしいと圧力がかかっていた
 - <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=1067708>
 - debian も攻撃されたと言えると思う
- MiniDebConf ベルリンでこの脆弱性の話の枠がある模様
 - <https://salsa.debian.org/ftp-team/xz-2024-incident/-/issues/12>
- 今回の問題は人とコードの問題をわけて議論する必要があると思う
- 人の話
 - コミット権はどのような理由で誰に与えるかの問題
 - * コミット権の与え方がゆるい人を作らない、というのはいいがその見極めは難しい気がする
 - * debian では公的機関が発行した証明書で本人認証をした上で Debian Developer を任命することになっている
 - ソーシャルハッキング
 - * 事前に対応するのは難しい
 - * 事後対応になりそうなので、どのように対処するか事前にある程度決めておく
- コードの話

- 今回は tarball にマルウェアのコードが入り込んだが、git にはマルウェアのコードはなかった
 - * debian.org では tarball と git の中身をクロスチェックするツールをつくってはどうかという意見あり
- Chain of Responsibility という考え方
- コードを作る人 (upstream) と送り届ける人 (distributor) に分けて考えたほうがいい
 - * debian は distributor
- コードのレビューで対応する案もあるが、この対応をボランティアでやりきるのはかなり辛い
 - * 実際見抜けるのか、という問題もありそう
- 今回の脆弱性は m4 マクロで埋め込んだものであるが、これは技術的に古く面倒見れる人が少ない
 - * autoconf、automake はビルドシステム職人が行う領域な気がする
 - * 新しいビルドシステム形式に置き換えを進めるのがいいのか？
 - * 辛いビルドシステムとは何？
 - ・ なにかあるかと言われると困る。どれも辛い気がする
 - ・ スクリプト言語などのビルドしないソースコードが主流になればよい未来が来る？
 - * 標準技術は時代とともに変わる
 - ・ https://www.explainxkcd.com/wiki/index.php/927:_Standards
- 多様な環境を提供するのが Debian のアイデンティティだと思う
- 対応は個人でできるレベルを超えているため、組織的に関与していかないと対応は難しいと思う

5 Debian における 64bit time_t 移行について

Kentaro HAYASHI (林健太郎)



本記事は Debian 勉強会 (2024 年 4 月開催) にて公開した資料を冊子向けに編纂・収録したものです。現在、Debian における 64bit time_t 対応について主要なパッケージに関してほぼ完了し、収束へと向かっています。

*12

5.1 64bit time_t 移行はなぜ必要なのか

昨今の Linux ディストリビューションの動向として、32bit アーキテクチャのサポートは廃止ないしは縮小の流れがあります。Debian では armel, armhf, i386 がサポートされている 32bit アーキテクチャとして残っています。これらアーキテクチャでは 32bit time_t のままでは (実機がそのころまで稼働しているかというはおいておくと) 2038 年問題が不可避となっています。扱える時刻の範囲が signed 32bit で扱える範囲 (~ 2147483647) の限界を迎えてしまうためです。

この 2038 年問題の X デーは 2038 年 1 月 19 日です。日本時間だとちょうどお昼ごろに崩壊がやってきます。

```
Time.at(2147483647)
=> 2038-01-19 12:14:07 +0900
```

かつて、時刻の扱いにまつわる問題として、2000 年問題がありました。2000 年問題とは、西暦下 2 桁処理でリソースを節約する工夫がかえって問題を引き起こすものでした。上記前提で実装されたアプリケーションによる障害発生が騒がれましたが、事前の対策が進められたことから大混乱は回避された印象があります。一方の 2038 年問題は 32bit アーキテクチャにおいて、業務アプリケーションのレイヤーではなくシステム全体で障害発生しうるより深刻な問題です。^{*13}

5.2 Debian における 2038 年問題対策の構成要素

Debian において、対策が必要な構成要素は多岐に渡ります。カーネル、glibc、ファイルシステム、アプリケーションやライブラリーなどです。これらで 2038 年問題を踏むとどうなるでしょうか。

```
$ sudo date --set="2038-01-19 12:14:00"
Tue Jan 19 12:14:00 UTC 2038
$ sudo ls
sudo: unable to open /etc/sudo.conf: Value too large for defined data type
sudo: unable to determine tty: Value too large for defined data type
sudo: unable to get time of day: Value too large for defined data type
sudo: error initializing audit plugin sudoers_audit
```

^{*12} <https://slide.rabbit-shocker.org/authors/kenhys/tokyodebian-timet-transition-202404/>

^{*13} 業務アプリケーションレベルでも深刻なのにそれを動かすベースのシステムから死ぬというわかりやすいでしょうか

例えば、対策のなされていないシステムにおいて X デーに日付を更新してみると、シェルでの操作が不能に陥ります。

ただし、昨今では 2038 年問題はある程度対策が進んでいます。カーネル (kernel 5.6 以降対応済み) や glibc (2.34 以降対応済み)、ファイルシステム (ext4 など対応済み) などでは対処済みです。^{*14}したがって、問題になるのはアプリケーションやライブラリーのレイヤーとなります。

5.3 Debian における 64bit time_t transition

Debian においては 2038 年問題は 64bit time_t transition として段階を踏んで対策が進められています。詳細については 64bit time_t に関する Debian Wiki のページ^{*15}を参照するとよいでしょう。

具体的には次のような対策が進められました。

- transition が必要な ABI の変更対象をリストアップする^{*16}
 - 当初少なくとも 4900 ソースパッケージが影響を受けると考えられていた
- dpkg-buildflag が `-D_FILE_OFFSET_BITS=64` と `-D_TIME_BITS=64` を既定で有効にする (i386 系を除く)
 - dpkg 1.22.5 以降で有効となりました
- NMU でライブラリー名変更を実施
 - 進行中は依存関係を満たせずインストールできない、意図せず削除される等が発生しました^{*17}
- 依存ライブラリーを大量にリビルドしなおす

移行を一気に押し進めた 2024 年 2 月頃は依存関係が激しく壊れひどい状況でしたが、4 月頃には概ね収束へと向かいました。

5.3.1 NMU でライブラリー名変更を実施

移行の過程ではライブラリー名の変更が実施されました。これには 64bit time_t に対応しているものを区別する意味合いがあります。

NMU(Non Maintainer Upload) とよばれる手続きにより、次のようにして依存関係の変更が行われました。例えば、`debian/control` を以下のように変更します。

```
-Package: libfoo0
+Package: libfoo0t64
+Provides: ${t64:Provides}
+Replaces: libfoo0
+Breaks: libfoo0 (<< ${source:Version})
```

上記のようにライブラリーのパッケージ名のサフィックスに `t64` が付与されました。`t64:Provides` は、`debhelper 13.14` で実装されているマクロです。^{*18}従来の 64bit time_t に対応していないパッケージから円滑なアップデートができるようにしています。

5.4 64bit time_t transition の進捗状況

Ubuntu では、24.04 LTS で早々と `armhf` 2038 年問題へのサポートをアナウンスしています。

Debian においては、2038 年問題はバグレポートに `time-t` タグが付与されトラッキングされています。^{*19} これをみると、まだバグが残っており、完了したとは言えない状況であることがわかります。とはいえ主要なパッケージの

^{*14} ext2 は対応していないので、`/boot` に ext2 を使っているケースなどは注意が必要かもしれません

^{*15} <https://wiki.debian.org/ReleaseGoals/64bit-time>

^{*16} <https://lists.debian.org/debian-devel/2023/05/msg00168.html>

^{*17} 特定の時期に限定されますが、パッケージが削除されているせいなどが理由で `testing` を使っているひとはアップデートでシステムを壊しやすくなっていたりもしました。

^{*18} `t64:Provides` の詳細については `dh-makeshlibs` を参照するとよい。

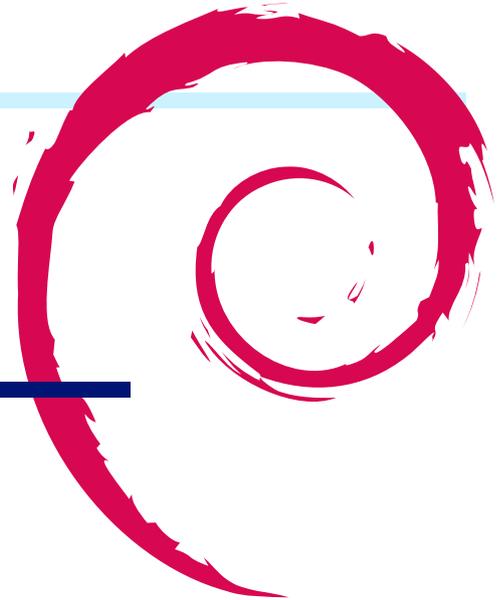
^{*19} <https://bugs.debian.org/cgi-bin/pkgreport.cgi?users=debian-arm@lists.debian.org;tag=time-t>

対応が済んでいるので、実用上は問題となることは少ないでしょう。

5.5 まとめ

Debian における 64bit `time_t` 移行について紹介しました。もしかしたら移行の過程で 32bit アーキテクチャだけ特定パッケージがサポートをやめてしまったり、機能が落とされている場合があるかもしれません。

そのような場合を見つけたら、bugs.debian.org にぜひバグ報告してみてください。



6 python の仮想環境管理 venv と pipx

Norimitsu SUGIMOTO (杉本 典充)

6.1 自己紹介

- Norimitsu SUGIMOTO (杉本 典充)
- dictoss@live.jp
- Twitter: @dictoss
- Debian を使い始めたのは 3.1 sarge が testing の頃
- 仕事はソフトウェア開発者をやってます
- python と Django の組み合わせで使うことが多いです

6.2 これまでの python パッケージのインストール

- これまでは特に気にせずに以下コマンドを実行

```
# pip3 install xyz
```

- Debian 12 bookworm で実行した場合

```
# cat /etc/debian_version
12.5

# pip3 install xyz
error: externally-managed-environment

× This environment is externally managed
|--> To install Python packages system-wide, try apt install
python3-xyz, where xyz is the package you are trying to
install.
(以下略)
```

- エラーになって困りませんでしたか？

6.3 Debian 12 bookworm における python の変更

6.3.1 PEP-668 と externally-managed (1)

- Debian 12 bookworm リリースノート^{*20}
 - Debian が提供する python3 インタプリタパッケージ (python3.11 および pypy3) は、PEP-668 に従っ

^{*20} <https://www.debian.org/releases/bookworm/amd64/release-notes/ch-information.ja.html#python3-pep-668>

- て externally-managed とマークされるようになりました。
- Debian で提供されるバージョンの python3-pip はこれに伴って、`-break-system-packages` オプションが指定されない限り、Debian の python インタプリタ環境へ手作業でのパッケージインストールを拒否します。
- 詳細については `/usr/share/doc/python3.11/README.venv` を参照してください。
- よくわからない言葉が出てきます
 - PEP-668、externally-managed
 - python3-pip、`-break-system-packages` オプション
- PEP-668*²¹
 - Marking Python base environments as “ externally managed ”
 - python を実行する環境より外部の環境で python が管理される、という意味
 - 外部の環境 = Debian における deb パッケージによる管理
- externally-managed として管理されてることを示すファイル
 - `/usr/lib/python3.11/EXTERNALLY-MANAGED`
 - 中身は ”pip3 install xyz” を実行したときのエラーメッセージとほぼ同じ
 - この状態で「pip3 install xyz」を実行するとエラーで止まる
- Debian ではパッケージ化されていない python パッケージをインストールするには？
 - 【非推奨】「pip3 install xyz `-break-system-packages`」で強引に実行
 - 【推奨】python の仮想環境を作成して、その仮想環境の中へ python パッケージをインストールする
 - * venv を使う (python ライブラリ向け)
 - * pipx を使う (python アプリケーション向け)

6.4 python の仮想環境 venv

6.4.1 venv モジュール

- venv — 仮想環境の作成
 - <https://docs.python.org/ja/3.11/library/venv.html>
- 2012/9/29 に リリースした python3.3 から標準機能として利用できる仮想環境の作成手段
- 以下コマンドで仮想環境を作成できる

```
$ mkdir -p myvenv
$ cd myvenv
$ python3 -m venv .
$ ls
bin include lib lib64 pyvenv.cfg
$ cat pyvenv.cfg
home = /usr/bin
include-system-site-packages = false
version = 3.11.2
executable = /usr/bin/python3.11
command = /usr/bin/python3 -m venv /home/dictoss/myvenv
```

6.4.2 venv を使った仮想環境のパッケージ管理 (1)

`bin/activate` を実行すると仮想環境へ入る

仮想環境の中の `bin` ディレクトリが `PATH` 環境変数へ追加され、`python` コマンドと `pip` コマンドは仮想環境の中のコマンドを優先して実行するよう切り替える

*²¹ <https://peps.python.org/pep-0668/>

```
$ ls bin
Activate.ps1  activate.csh  pip  pip3.11  python3
activate      activate.fish  pip3  python   python3.11

$ source bin/activate
(myvenv) $ which python
/home/dictoss/myvenv/bin/python
(myvenv) $ which pip
/home/dictoss/myvenv/bin/pip
```

deactivate を実行すると仮想環境を抜ける

```
(myvenv) $ deactivate
$
```

6.4.3 venv を使った仮想環境のパッケージ管理 (2)

pip コマンドで python ライブラリをインストールしたとき、仮想環境の中の lib ディレクトリへインストールされる

```
(myvenv) $ pip install python-dateutil
(myvenv) $ pip freeze
python-dateutil==2.9.0.post0
six==1.16.0

$ ls lib/python3.11/site-packages/
__pycache__          pkg_resources
_distutils_hack      python_dateutil-2.9.0.post0.dist-info
dateutil             setuptools
distutils-precedence.pth  setuptools-66.1.1.dist-info
pip                  six-1.16.0.dist-info
pip-23.0.1.dist-info  six.py
```

python-dateutil ライブラリを仮想環境内で利用できている

```
(myvenv) $ python
>>> import dateutil.parser
>>> dateutil.parser.parse('Sat, 18 May 2024 14:15:16 +0900')
datetime.datetime(2024, 5, 18, 14, 15, 16, tzinfo=tzoffset(None, 32400))
```

6.5 アプリケーション向け仮想環境管理 pipx

6.5.1 pipx とは

- pipx は python のアプリケーション単位で仮想環境を管理するコマンド
 - venv を使った仮想環境は基本的にアプリケーションごとに用意すると良いのだが、多いと管理がづらい
 - その仮想環境の管理を任せる仕組みが pipx
- pypi で提供されており^{*22}、debian では pipx パッケージで提供している

```
$ sudo apt-get install pipx
$ pipx --version
1.1.0
```

6.5.2 pipx の使い方 (パッケージ単体の場合)

pipx のインストール後は、pipx ensurepath を実行する

```
$ pipx ensurepath
Success! Added /home/dictoss/.local/bin to the PATH environment variable.

Consider adding shell completions for pipx. Run 'pipx completions' for
instructions.

You will need to open a new terminal or re-login for the PATH changes
to take effect.

Otherwise pipx is ready to go!
```

`${HOME}/.local/bin` が PATH に追加される

^{*22} <https://pypi.org/project/pipx/>

```

$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:
/home/dictoss/.local/bin

$ grep -i pipx ~/.bashrc

# Created by `pipx` on 2024-05-15 13:51:28
export PATH="$PATH:/home/dictoss/.local/bin"

```

6.5.3 pipx の使い方 (パッケージ単体の場合)

例: python アプリケーションの pelican をインストールする

```

$ pipx install pelican
installed package pelican 4.9.1, installed using Python 3.11.2
These apps are now globally available
- pelican
- pelican-import
- pelican-plugins
- pelican-quickstart
- pelican-themes
done!

```

pelican コマンドが `/${HOME}/.local/bin/pelican` に追加され、実体は `venv` で作成した仮想環境の中のインストールした pelican コマンドのシンボリックリンクになっている

```

$ which pelican
/home/dictoss/.local/bin/pelican

$ ls -l /home/dictoss/.local/bin/pelican
lrwxrwxrwx 1 dictoss dictoss 53  5月 15 23:04
/home/dictoss/.local/bin/pelican
-> /home/dictoss/.local/pipx/venvs/pelican/bin/pelican

```

6.5.4 pipx の使い方 (requirements.txt を使う場合)

既存で動いている python アプリケーションがあり、それは requirements.txt でバージョンが指定されている場合

```

$ git clone https://github.com/dictoss/xddc.git
$ cd xddc
$ cat requirements.txt
pelican==4.8.0
Jinja2==3.1.3

```

requirements.txt に書かれている `pelican==4.8.0` がインストールされる

```

$ pipx install pelican --pip-args "-r requirements.txt"
installed package pelican 4.8.0, installed using Python 3.11.2
These apps are now globally available
- pelican
- pelican-import
- pelican-plugins
- pelican-quickstart
- pelican-themes
done!

```

6.6 これまで通りのやり方をしたいあなたに

6.6.1 pip3 の `-break-system-packages` オプション

- python3-pip パッケージをインストールするとこれまで通り pip3 コマンドを利用できる
- pip3 コマンドに `-break-system-packages` オプションを付与して実行すれば、これまで通り python パッケージは一応インストールできる
 - `pip3 install xyz -break-system-packages`
 - `pip3 uninstall xyz -break-system-packages`
- pip3 コマンドで `-break-system-packages` オプションを指定する方法を続けてもよさそうな状況
 - 1 台のホスト環境の中で python アプリケーションを 1 つしか動かさない場合
 - 例) docker、k8s、lxc、chroot、KVM 上の仮想マシンなどで Debian 環境自体を分離して実行している

6.7 まとめ

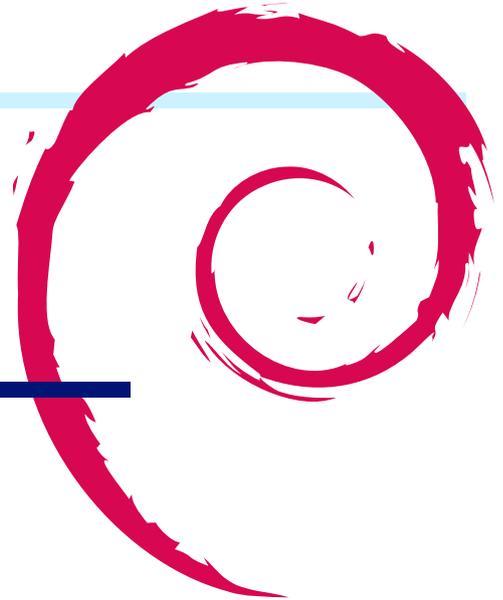
- Debian 12 bookworm で python アプリケーションを使う場合に覚えておくとい仕様変更の「externally-managed」
- python の標準機能の仮想環境を作成する venv
- python アプリケーションごとに仮想環境を作成・管理する手間を減らせる pipx コマンド
- これまで通りのスタイルを通すなら python3-pip の `-break-system-packages` という奥の手

6.8 参考資料

- pipx - pypi <https://pypi.org/project/pipx/>
- PEP 668 <https://peps.python.org/pep-0668/>
- venv モジュール <https://docs.python.org/ja/3.11/library/venv.html>

7 2023 年を振り返って

2023 年 12 月の参加者



7.1 参加者

参加者は以下の 7 名でした。(敬称略)

- dictoss
- yy_y-ja.jp
- NOKUBI Takatsugu (knok)
- Hiroyuki Yamamoto (yama1066)
- Kazuhiro NISHIYAMA (znz)
- koedoyoshida
- okao5963

7.2 参加者の意見

7.2.1 dictoss

- やったこと
 - Debian 勉強会を 1 年間続けることができ、7 回発表できた (2 月、3 月、4 月、5 月、8 月、11 月、12 月)
 - OSC などのイベントで新しい人や若い人を呼び込む工夫に悩んでいる
 - * 他の団体に聞くと珍しい物を展示すると寄ってくる、とのこと
 - * OSC などの展示ブースの拡充
 - ・ RISC-V ボードは入手した
 - ・ 他に珍しい物を探してみる
- やれるとよかったこと
 - パッケージの ITP はできなかった (とはいえ、あまり自分でアップロードしたいパッケージがない)
 - @tokyodebian での X(旧 Twitter) の投稿ができておらず、情報発信頻度に難あり
 - wayland 環境はあまり触っていないため、そろそろ慣れたほうがいいのか? (Xfce4 が Wayland に対応するのはいつですか?)
- 気になった話題
 - LoRaWAN の情報を調べていると armbian 搭載の機器を見つけた
 - * https://docs.google.com/spreadsheets/d/10BC823F2FWwrs8_VRHSLjkgpV2xF7Ao0bspXv09K3wWo/edit#gid=0
 - * LPS8v2

* LG01v2

7.2.2 knok

● やったこと

- Debian 特有のことがあんまり作業できていない
 - * アップグレード時の問題・対応メモ <https://github.com/knok/debian-stretch-upgrade-notes/wiki/bullseye%E2%80%90to%E2%80%90bookworm>
 - * mailman3 は BTS でやりとりした <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=1041496>
- Emacs-28.2 の native compilation 関連で起きる問題
 - * 非同期外部プロセス周りがどうも鬼門の模様
 - * ex: mew <https://groups.google.com/g/mew-ja/c/oN924UHMJd4?pli=1>
 - * 似た問題が Wanderlust でもある
 - ・ native compilation を無効にするしか現状対応策はなさそう
 - ・ notmuch の移行を考えている <https://wiki.archlinux.jp/index.php/Notmuch>

● やれるとよかったこと

- Debian 30 周年「Debian Day 2023」 <https://wiki.debian.org/DebianDay/2023>
- bookworm リリースパーティー
- 森若さんの systemd 本が気になる <https://gihyo.jp/book/2024/978-4-297-13893-6>
- crossgrade をちゃんと見ていきたい <https://wiki.debian.org/CrossGrading>

● 気になった話題

- bulleseye tor support 終了 <https://lists.debian.org/debian-lts/2023/11/msg00019.html>
- ext4 のバグで Debian 12.4 リリースに支障が出た <https://micronews.debian.org/2023/1702270964.html>
- EU の法律に関する声明 vote https://www.debian.org/vote/2023/vote_002

7.2.3 koedoyoshida

● やったこと

- 夏、冬の Debian 本の編集
- 別イベントの主催を通じて、海外発表者への VISA 発給方法などは分かったこと

● こういうことをできれば良かった

- Minidebconf および Debconf の準備

7.2.4 yy-y-ja-jp

● やったこと

- OSC 出展

● こういうことをできれよかった

- 学生など若い方へのアプローチ: 出張勉強会?

7.2.5 yama1066

● やれるとよかったこと

- bookworm のリリースパーティーには出たかった

7.2.6 znz

- やれるとよかったこと
 - Debian 10 や 11 のままの環境が残っているため更新したい

7.2.7 okao

- お聞きしたい事
 - 昔はメーリングリストで linux、debian について質問できていたのですが、最近はメーリングリストが機能しているのか、質問しても良いものなのか分かりません
 - * 最近、linux や deibian の技術的な質問は、どちらにする感じなのでしょうか？
 - * <https://www.debian.or.jp/community/ml/> こちらの Debian 日本語メーリングリストは今も生きています

7.2.8 各回の参加人数とセミナー

2023 年に実施した Debian 勉強会の参加人数^{*23}とセミナー内容は以下の通りです。

表 1 2023 年の勉強会の参加人数とセミナー

開催日	参加人数	セミナータイトル
2023-01-21	9 人	「Debian パッケージング アップロード編」 BoF「2023 年の目標」
2023-02-18	8 人	「DDTP 及び DDTSS の紹介」 ハンズオン「DDTSS を使ってパッケージ説明文を翻訳してみよう」
2023-03-11	17 人	「Debian Updates (OSC 2023 Online/Spring 出張勉強会)」
2023-04-01	87 人	「OSC 2023 Tokyo/Spring 出張展示」
2023-04-15	12 人	「Debian で PHP が動く Web サーバを構築してみる (apache2 / nginx)」 「Debian パッケージング Open Build Service 編」
2023-05-20	6 人	「レポート ”Analysis of the Debian Developer Survey (2022) ” を読んでみて」
2023-06-17	32 人	「Debian Updates (OSC 2023 Online/Hokkaido 出張勉強会)」
2023-06-24	48 人	「OSC 2023 Hokkadio 出張展示」
2023-07-15	9 人	「Debian Installer の日本語フォントをどうにかする話」
2023-08-19	10 人	「Live Build を使ったカスタムフレーバの作成」 「2038 年問題と 32 bitOS の 64 bit time_t への移行について」
2023-09-16	10 人	「DebConf23 インド現地レポート」
2023-10-21	37 人	「OSC 2023 Tokyo/Fall 出張展示」
2023-11-18	8 人	「apt の JSON Hook プロトコルを使ってみた」 「自宅サーバを nftables で守ってみる」
2023-12-16	7 人	「RISC-V ボード VisionFive2 に Debian を入れてみた話」 BoF「2023 年を振り返って」

^{*23} <https://salsa.debian.org/tokyodebian-team/tokyodebian-team.pages.debian.net/-/blob/master/event-history.txt>

- 「GTK 4 support? #173 」 <https://github.com/uim/uim/issues/173>
 - なにか手伝えることはあるだろうか (技術的な勉強をいろいろする必要がある)
- 2025/1 にはおそらく trixie のフリーズが開始されるため、なにか準備が必要か？
- XDDC(OS 横断の日本語系開発コミュニティ) の活動
- isc-dhcp の代わりを探しておく <https://www.debian.org/releases/stable/amd64/release-notes/ch-information.ja.html#deprecated-components>

8.3.2 kenhys

- GTK2 からの置き換えを進めることができるとよい
 - libgtk2.0-0: is deprecated and entirely unmaintained upstream
 - <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=947713>
- <https://fabre.debian.net> のメンテナンス
- 理想のキースイッチとキーキャップの組み合わせを見つける (UHK v2 を昨年 Debian プロジェクトの支援で購入したのでちゃんと使いこなす)

8.3.3 NOKUBI Takatsugu (knok)

- debugpt の概要や使った例の紹介
- LLM のファインチューニングで日本語処理に利用可能なコーパスの作成やタスク実施
- 32bit 64bit crossgrading の評価・実施 <https://wiki.debian.org/CrossGrading>
- anthy、uim、sj3 デバッグ環境の確立
- elisp でいまいちうまく動かなくなった wanderlust をどうにかする or 別 MUA 移行 (on tty)
- FTBFS DoubleBuild の修正 <https://wiki.debian.org/qa.debian.org/FTBFS/DoubleBuild?action=show&redirect=qa.debian.org/%2FFTBFS%2FSourceAfterBuild>

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。

```
$ sudo apt update
$ sudo apt install git
$ 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 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 オープンソースライセンス

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.

『あんどきゅめんでっど 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 2023/11 ~ 2024/05 まで。

内容は無保証、つっこみなどがあれば勉強会にて。



あんどきゅめんでっど でびあん 2024 年夏号

2024 年 8 月 12 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)

<https://debianjp.connpass.com>

<https://tokyodebian-team.pages.debian.net>

<https://wiki.debian.org/KansaiDebianMeeting>

