



# *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 2023 年夏号 2023 年 8 月 13 日 初版発行



# 今強勉のアービト

---

---

## 目次

1	Introduction	2
2	Blender と ffmpeg で動画編集	3
3	Debian で KVM を使う (CLI で呼吸している人向け)	11
4	Debian パッケージング repack 編そのソフトウェア、そのまま配布して大丈夫だっけ	20
5	BoF 相談・情報共有・もくもく会	25

---

# 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 Blender と ffmpeg で動画編集

Yosuke OTSUKI

YouTuber になったり、Udemy や講義などで動画を作成する機会が増えてきていると思います。OSS のみを利用して、動画を作成する手順が日本語であまりないと感じこの記事を書きました。なお、ターミナルの中に生息している人 Unix 系の OS に慣れている人を対象としています。

本文章では、最新の debian 11 で利用できる blender 2.83.5+dfsg-5+deb11u1 と ffmpeg 7:4.3.4-0+deb11u1 を前提としています。debian パッケージになっている blender とアップストリームの blender では機能に差異がある可能性が高く、debian のバージョンでは機能制限があることを確認しています。例えば、debian 版では rotate 機能がありませんでした。

なお、本文章では blender のスクリプト機能は使用していません。また、動画編集機能の説明になります。3D モデリングを作成し、それを映像作品として出力する方法は説明しません。

### 2.1 ffmpeg の基本ルール

基本的に以下だけ覚えておけば良いと思います。あとは、適時検索をするとよいです。

#### 2.1.1 ffplay で確認して、ffmpeg で動画を出力する

動画や音声の簡易プレイヤーとして使用できます。なお、ここでは入力ファイルの形式は、動画が mp4 と音声ファイルが mp3 形式と仮定しています。

```
ffplay -i ${movie}.mp4
```

```
ffplay -i ${sound}.mp3
```

本来はフィルタなどを適応してリアルタイムで確認する用途です。下記コマンドを使用すると、読み込んだ動画が反転して再生されます。ffmpeg に同じオプションを渡す事で、動画として出力できます。

```
ffplay -i ${input} -vf "transpose=1" ${output}.mp4
```

#### 2.1.2 video と audio filter

コマンドラインオプションの -vf と -af のあとに、対応するフィルタのオプションを羅列します。-vf と -af はコマンドライン中に一度しか宣言できません。なお、他の方法もあります。

```
ffmpeg -vf ${動画フィルタのオプション} -af ${音声フィルタのオプション} ${output}.mp4
```

## 2.2 動画作成の流れ

1. 台本を書く
2. 絵コンテを作成
3. 動画を撮影
4. 絵コンテをベースに、ナレーションを録音
5. 必要ならば、動画を各種フィルタで補正
6. blender で音声と動画を編集
7. ffmpeg で合成

## 2.3 台本を書く

お好きなエディターで書いてください。アウトラインぐらいは書いた方が良いです。

## 2.4 絵コンテを作る

適当なツールがなかったので、紙媒体やりました

## 2.5 ナレーションを録る

マイクから音声を入力し、録音します。マイクによっては、highpass と lowpass の値を調整する必要があります。各種ノイズフィルターも利用できるのですが、以下の highpass と lowpass フィルタだけで十分でした。2000 円ぐらいのヘッドセットで録音しましたが、聞くには問題ない品質だと思います。フィルタの設定はハードウェア毎に探る必要があると思います。携帯付属と 100 均のマイク付きイヤホンでも試してみましたが、音質はかなりきびしいです。必要に応じて、ノイズキャンラなども試してみましたがあまり改善はしませんでした。

```
ffmpeg -f pulse -i default -af highpass=f=4000,lowpass=f=100 ${output}.mp3
```

stackoverflow などでは、highpass=200,lowpass=3000 の記述が多いですが、わたしのハードウェアでは上記が有効でした

```
* オプション
-f: format
-i: 入力指定。この場合はデフォルトのサウンドデバイスになる
-af: audio filter
```

デフォルトのサウンド入力となっているデバイスは、arecool や pacmd など確認できます

## 2.6 動画を撮影する

スマホで撮影する。撮影したら、PC にインポートしてください。droidcam なんていうアプリもあるので、スマホをカメラとして使用することもできます。わたしは、いつもインポートしています。

## 2.7 必要ならば、動画を各種フィルタで補正する

### 2.7.1 動画から音声を取り除く

一部の動画では、ナレーションを被せるため動画の邪魔な音声を取り除きます。以下のコマンドラインで完全に取り除きました。

```
$ ffmpeg -i ${input} -vcodec copy -an ${output}.mp4
```

```
* オプション
-i: input
-vcodect: ビデオコーデック
-an: output (省略可能)
```

## 2.7.2 動画を回転させる

スマホで撮影したところ、横向きで録画したのに縦向きで表示されていたり、その逆に保存されていた動画が多数ありました。それを修正します。

```
ffmpeg -i ${input} -vf "transpose=1" ${output}.mp4
```

```
* オプション
-i: input
-vf: ビデオフィルタ

* transpose のオプション
- transpose: 0,1,2,3 のいずれかの整数で回転方向を指定
  - 0: 90 °時計周り + 縦軸で反転
  - 1: 90 °半時計周り
  - 2: 90 °時計周り
  - 3: 90 °時計周り + 縦軸で反転
```

## 2.7.3 録音した音声を結合 (concat) する

GUIのある動画編集ソフトウェアで結合するのが一般的ですが、ffmpeg でやりました。blender では複数のファイルに対して同一の処理ができなかったためです。ただし、方法では動画間の遷移効果なども ffmpeg で行う必要があります。このような編集を行う場合は、GUIの方が効率的かもしれません。動画を結合する場合でも、同様の方法で行えます。

```
$ cat file_list.txt
file './movie1.mp3'
file './movie2.mp3'
file './movie3.mp3'
$ ffmpeg -f concat -safe 0 -i file_list.txt -c copy res.mp3
```

## 2.8 Screen Capture

スクリーンのキャプチャもできます。

```
ffmpeg -video_size 1024x768 -framerate 24 -f x11grab -i default output.mp4
```

音声もキャプチャしたい場合は以下になります。なお、結構ホワイトノイズが入ります。なぜか high,low フィルタを同時に使用すると無音になりました。一度ノイズありの状態キャプチャーをしてから、フィルタを適応すると良いでしょう。

```
ffmpeg -video_size 1024x768 -framerate 24 -f x11grab -f pulse -ac 2 -i default output.mp4
```

## 2.9 BGM を準備する

copy light free のBGMが公開されているので、それを使用しています。自作できるならば、それを使用してください。

## 2.10 blender で音声と動画を合成する

まずは、blender をインストールします。

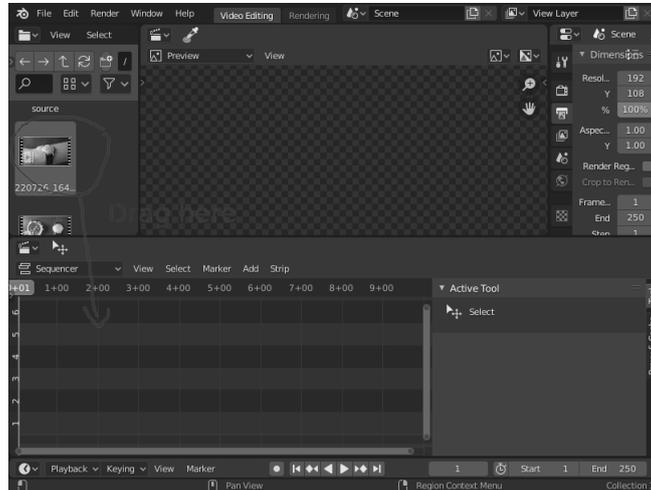
```
# apt-get install blender
```

blender を起動すると、スタートアップメニューが表示されるので [Video Editing] を選択してください。使用して

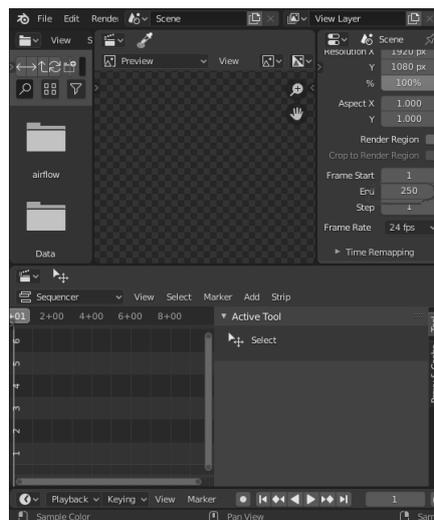
いる画像は LANG=en\_US.UTF-8 の環境でキャプチャしています。

### 2.10.1 動画、音声と画像ファイルをインポートする

左上のカラムのアイコンから、対象のファイルをシーケンサにドラッグ。もしくは、メニューバーの [file/import] ([ファイル/インポート]) から対象のファイルをインポートします。対応している画像、動画と音声ファイルならばインポートできます。



デフォルトでは最終のフレームが 250 になっています。このままだと、動画が途中で終了したり、余分な黒画面が表示されてしまうので適当な値を入力して変更してください。



### 2.10.2 タイミングを合わせる

マウスでシーケンサのカラーバーをドラッグすると、対象が表示されるタイミングが調節できます。カラーバーの端をクリックすると延長/短縮できるモードに切り替ります。

また、シーケンサー上で上下にドラッグする事で、チャンネルを変更する事ができます。動画、画像、ナレーションや BGM など、それぞれのチャンネルに分けておいた方が整理できます。

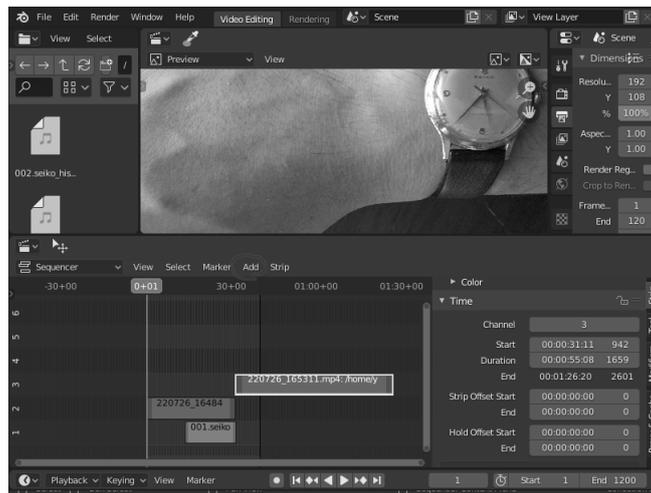


しかし、これよりも開始と終了のフレーム数を指定した方が良いと思います。シーケンサで対象をクリックして選択すると、対象のパラメタが右下のメニューに表示されます。ここで Start もしくは End に希望するフレーム数を入力してください。



### 2.10.3 動画に文字を入れる

シーケンサの上、中段のメニューにある [add/text] を選択すると、シーケンサに text オブジェクトが生成されます。テキストの替りに、ロゴなどを入れたい場合は画像としてインポートできます。gimp で画像を作成、または加工してください



text オブジェクトを選択して、パラメタを編集します。Effect Strip (赤枠) に表示したい文字列を入力し、Style (黄枠) で文字のサイズと色を調節します。そして、Layout (青枠) で文字の位置を調節しましょう。

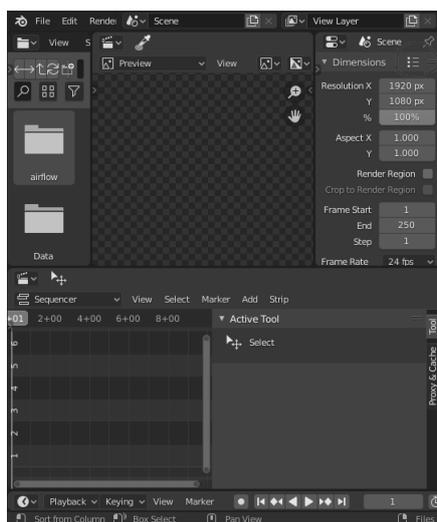


#### 2.10.4 編集した動画を出力する

debian 版の bledner は、動画と音声を合成して 1 つの動画ファイルとして出力することができません。<sup>\*1</sup>そのため、動画ファイルの出力と音声ファイルの出力を別々に行い、ffmpeg で合成します。

まずは出力する動画のフォーマットを決めます。デフォルトで full HD の 24 fps になっているので YouTube などにアップロードする場合は変更する必要はないと思います。変更する場合は、dimension の横のアイコンをクリックすると、任意のサイズを選択できます。

\*1 アップストリーム版は未確認

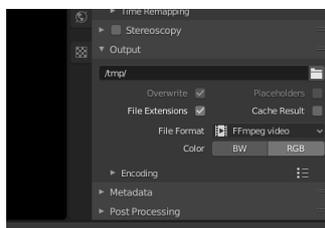


また、少しスクロールしていくと [Frame Rate] という入力があります。業界のデファクトスタンダードとして、YouTube が 24fps, 映画が 30fps, 特殊用途の動画が 60fps らしいです。

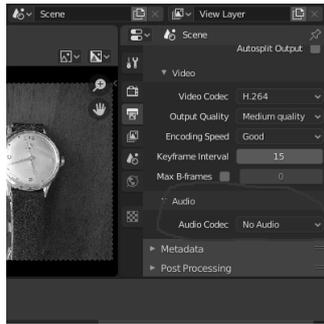
次に、編集した動画をファイル出力します。右上段のメニューに、[Output] というファイルパスがあります。それが動画ファイルの出力先です。デフォルトでは /tmp になっています。



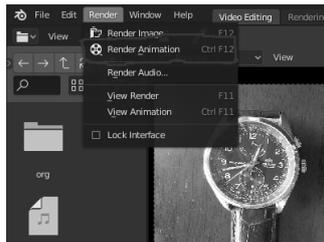
また、File Format も PNG になっているので動画ファイル形式に変更しましょう。ffmpeg video 形式が良いと思います。



blender の動画編集機能では、動画をレンダリングした時、デフォルトでは音声は同時に出力されないようになっています。動画のレンダリングに音声を追加するには、右側のメニューの [Output] タブの一番下にある [Audio] タブを展開し、Audio Codec を有効にしてください。



動画のレンダリングはトップのメニューバーから [Render/Render Animation] を押下してください。これで、動画ファイルが指定したパスに出力されます。



## 2.11 番外編

### 2.11.1 その他のツール

OBS: たぶん知名度は No.1。ストリーミングから、動画の録画までできる。しかし、動画の編集機能はない(はず)。Podcast や雑談などを録画するには良い。debian 11 ではパッケージ化済

Natron: 動画編集用のソフトウェア、Blender の代りに使用できる。Adobe の After Effect と同じような、node + edge で編集する仕様らしい。この仕様がハイエンドには多いとか...(未確認) プロ用の動画編集ソフトウェアを使用した経験がなかったので、使用しなかった。慣れている人には、学習コストが低いかもしれない。debian 11 ではパッケージ化はされていない。アップストリームからバイナリをインストールするのが早い。動画のエンコードはかなり遅かったのでローカルでビルドしても良いかも

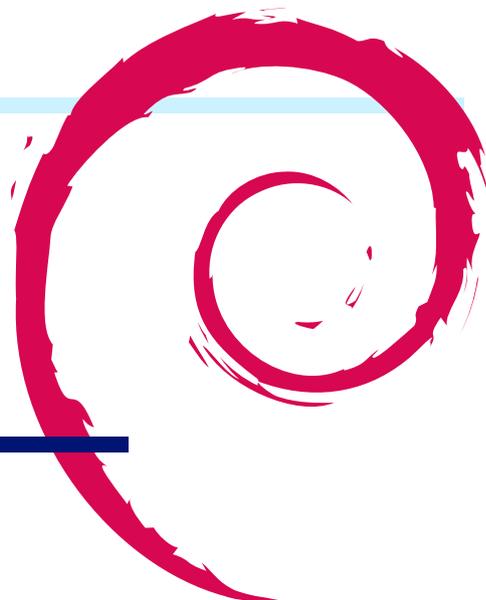
Openshot: こちらも動画編集用のソフトウェア Blender を代替できる。debian 11 ではパッケージ化済触った感じでは、blender よりも学習コストは低そう

DaVinci Resolve: これだけ、OSS ではない。プロ用のハイエンドだが無料版がある。Win, Mac と Linux のバイナリが公式から入手可能。インストールしてみたが Nvidia の GPU が必須で起動ができなかった

OpenJTalk: OSS ゆっくり動画っぽいものを作成する場合は、OpenJTalk を使用する方法もあります

## 3 Debian で KVM を使う (CLI で呼吸している人向け)

杉本 典充



### 3.1 はじめに

自宅で使っている PC の入れ替えに伴い、Debian で KVM サーバを作りました。最近の Debian 勉強会では KVM の話題はないため、まとめてみました。今回は KVM をコマンドラインで操作する CLI ツールに焦点を当ててみます。

なお、動作環境については OS は Debian GNU/Linux 11 bullseye amd64、CPU は Intel Core i5-6500T とします。

### 3.2 KVM とは

KVM (for Kernel-based Virtual Machine) とは、「仮想化拡張機能 (Intel VT または AMD-V) を含む x86 ハードウェア上の Linux 用の完全仮想化ソリューション」です<sup>\*2</sup>。公式の Web サイトには x86 とだけ書いてありますが、Debian Wiki の KVM の説明ページ (<https://wiki.debian.org/KVM>) には、ARM 系 CPU も KVM が使えると記載があります。

コンピュータの仮想化技術は表 1 のように分類され、KVM は「完全仮想化 (ハイパーバイザ型)」に該当します<sup>\*3</sup>。

### 3.3 KVM のインストール

#### 3.3.1 KVM の動作条件確認

KVM を利用するには仮想化機能に対応した CPU が必要です。Intel 製 CPU であれば Intel-VT、AMD 製 CPU であれば AMD-V、ARM64 系 CPU であれば Arm Virtualization Extensions<sup>\*4</sup> と呼ばれている機能を搭載しており、かつマザーボードの設定で有効にしている必要があります。

Intel 製 CPU または AMD 製 CPU であれば、

```
$ grep -E 'vmx|svm' /proc/cpuinfo
```

というコマンドで仮想化機能が有効になっているか調べることができます (Intel 製 CPU の場合は vmx、AMD 製 CPU の場合は svm の文字列が出現します)。

<sup>\*2</sup> [https://www.linux-kvm.org/page/Main\\_Page](https://www.linux-kvm.org/page/Main_Page)

<sup>\*3</sup> 引用元: 東京エリア Debian 勉強会 2013 年 4 月 杉本典充「debootstrap を有効活用してみよう」8.1 仮想化技術について を元に一部加筆

<sup>\*4</sup> <https://developer.arm.com/documentation/ddi0406/c/System-Level-Architecture/The-System-Level-Programmers--Model/The-Virtualization-Extensions>

表 1 仮想化技術の分類

仮想化技術	実装例	仮想化環境の特徴
完全仮想化 (エミュレーション型)	QEMU VirtualBox	既存の OS を無修正のままゲスト環境として動作させる。ホスト OS で動作する仮想化アプリケーションがエミュレーションする。
完全仮想化 (ハイパーバイザ型)	KVM	既存の OS を無修正のままゲスト環境として動作させる。CPU 等の仮想化機能を使うことでホスト環境におけるオーバーヘッドを極力減らしている。
準仮想化	Xen	ホスト環境とやりとりする API を利用できるように修正が入った OS をゲスト OS として動作させる方式 (= 既存の OS そのままでは動かない)
コンテナ型仮想化	OpenVZ LXC FreeBSD jail docker(筆者加筆)	ホスト環境とゲスト環境は同一カーネルで動作しつつ、ホスト環境から分離したゲスト環境を提供する。

```
$ grep "model name" /proc/cpuinfo | head -n 1
model name      : Intel(R) Core(TM) i5-6500T CPU @ 2.50GHz

$ grep -E 'vmx|svm' /proc/cpuinfo | head -n 1
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi
mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good
nopl xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg
fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm
abm 3dnowprefetch cpuid_fault epb invpcid_single pti tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsbase
tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm mpx rdseed adx smap clflushopt intel_pt xsaveopt xsavec
xgetbv1 xsaves dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp
```

### 3.3.2 KVM のカーネルモジュールの読み込み

KVM を利用するには linux カーネルのカーネルモジュールを読み込む必要があります。Intel 製 CPU の場合は "kvm\_intel.ko"、AMD 製 CPU の場合は "kvm\_amd.ko" という名前になっています。

modprobe コマンドを実行してカーネルモジュールを読み込みます。以下は Intel 製 CPU のコマンド例となります。

```
# modprobe kvm_intel
```

lsmod コマンドで "kvm\_intel" または "kvm\_amd" モジュールが表示されていれば必要なカーネルモジュールを読み込んでおり、KVM の動作条件を満たしています。

```
# lsmod | grep kvm
kvm_intel      331776  0
kvm            937984  1 kvm_intel
irqbypass     16384   1 kvm
```

### 3.3.3 QEMU のインストール

KVM 上で仮想マシンを実行する QEMU をインストールするには以下コマンドを実行します。今回は CLI 操作を目的とするため、GUI 関連のパッケージをインストールしないよう "-no-install-recommends" オプションを指定しています\*5。

```
# apt install --no-install-recommends qemu-system libvirt-clients libvirt-daemon-system
```

\*5 <https://wiki.debian.org/KVM#Installation>

KVM 上に仮想マシンをコマンドラインでインストールする `virt-install` コマンドを含む `virtinst` パッケージをインストールします。`libosinfo-bin` パッケージには `virt-install` コマンドの `"-os-variant"` オプションに指定できる値を調べることができる `osinfo-query` コマンドが入っています\*6。

```
# apt install virtinst libosinfo-bin
```

上記コマンドをインストールすると `libvirt` グループが作成されます。以降の手順で仮想マシンを操作するコマンドを紹介しますが、そのとき `libvirt` グループ に所属しているユーザが仮想マシンを操作できます。自分のユーザを `libvirt` グループに追加しておきます。

```
# adduser <youruser> libvirt
```

### 3.3.4 KVM 上で動作する仮想マシンのネットワークの構成

KVM 上の仮想マシンが接続するネットワーク構成は、NAT と Bridge の 2 種類があります。

#### NAT 構成

NAT 構成は、ホストマシンの中に仮想マシン同士が通信する専用の NAT ネットワークを作る構成をいい、KVM 上の仮想マシン用ネットワークのデフォルト設定になっています。NAT 構成の場合、仮想マシン用ネットワークが KVM のホストマシンの中で完結するため、開発用 PC やネットワークが頻繁に切り替わるノート PC では NAT 構成の利用が向いています。

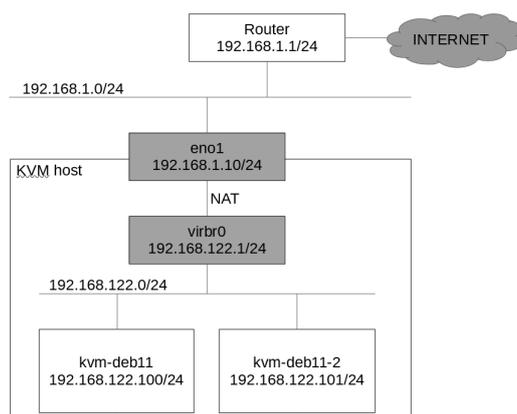


図 1 KVM ホストマシンと仮想マシンのネットワーク構成 (NAT 構成)

以下コマンドを実行すると、仮想マシン向けのネットワークとして `virbr0` インタフェースが起動します。デフォルトでは、`192.168.122.0/24` のネットワークが作成され、仮想マシンから見たデフォルトゲートウェイは `192.168.122.1/24` となります。

```
$ virsh --connect=qemu:///system net-start default
ネットワーク default が起動されました

$ ip addr show dev virbr0
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:db:b9:5b brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
```

この状態のままでは、PC の起動時に `virbr0` インタフェースは自動起動しません。もし、PC の起動時に自動で `virbr0` インタフェースを起動するようにしたい場合は以下コマンドを実行してください。

\*6 \$ `osinfo-query os` を実行すると一覧を表示します。

```
$ virsh --connect=qemu:///system net-autostart default
```

## Bridge 構成

Bridge 構成は、ホストマシンが接続するネットワークセグメントに仮想マシンも接続する構成のことをいいます。Bridge 構成の場合は仮想マシンの IP アドレスにホストマシンと同じセグメントの IP アドレスを割り当てるため、KVM のホストマシンとは別のホストからでも仮想マシンへ直接通信できるメリットがあります。

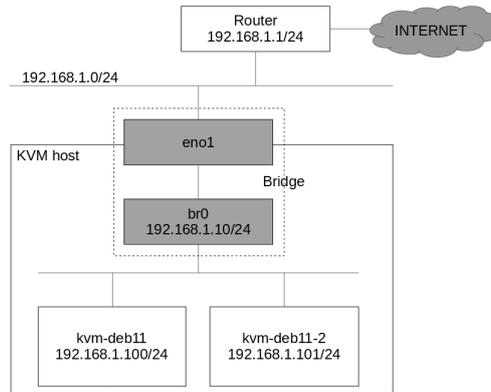


図 2 KVM ホストマシンと仮想マシンのネットワーク構成 (Bridge 構成)

Bridge 構成にするには以下コマンドで追加のパッケージをインストールします。

```
# apt install bridge-utils
```

次に有線 LAN の設定を Bridge 構成に変更します\*7。

変更前の有線 LAN で固定の IP アドレスに設定している設定ファイルは以下です。

```
$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eno1
iface eno1 inet static
address 192.168.1.10
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 192.168.1.1
bridge_ports eno1
bridge_stp off

# This is an autoconfigured IPv6 interface
iface eno1 inet6 auto
```

以下のようにネットワークの設定を変更して Bridge のインタフェースに IP アドレスを設定し、有線 LAN のインタフェースへ Bridge するよう紐つけます。

\*7 ネットワークの設定変更をするため、ssh 経由で操作している場合に設定ミスがあるとネットワークを再起動した後につながらなくなる危険性があります。できるだけホストマシンのコンソールへログインできる状況で作業する方がよいです。

```

$ sudo vi /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eno1
iface eno1 inet manual

auto br0
iface br0 inet static
address 192.168.1.10
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 192.168.1.1
bridge_ports eno1
bridge_stp off

# This is an autoconfigured IPv6 interface
iface eno1 inet6 auto

```

設定変更を反映するため、ネットワークを再起動します。

```
$ sudo systemctl restart networking
```

ホストマシンから `wget` や `apt` を実行して、ネットワークに接続できることを確認してみてください。

## 3.4 KVM 上の仮想マシンを操作するコマンド

### 3.4.1 qemu-img コマンド

`qemu-img` コマンドは、KVM 上で動作する仮想マシンの仮想ディスクを作成、変更するコマンドです。

仮想ディスクを作成するには `create` サブコマンドで作成できます。以下の例は、KVM で利用できる `qcow2` 形式の 10 GB の仮想ディスクを作成します。`qcow2` 形式の仮想ディスクは実際にデータを書き込んだ分だけファイルサイズが増えていくため（スパースファイル、またはスパースディスクといいます）、作成直後のファイルサイズはとて小さいです。

```

# qemu-img create -f qcow2 /var/lib/libvirt/images/kvm-deb11.img 10g
Formatting '/var/lib/libvirt/images/kvm-deb11.img', fmt=qcow2 cluster_size=65536 extended_l2=off
compression_type=zlib size=10737418240 lazy_refcounts=off refcount_bits=16

# ls -lh /var/lib/libvirt/images/kvm-deb11.img
-rw-r--r-- 1 root root 193K 10月 12 23:11 /var/lib/libvirt/images/kvm-deb11.img

# file /var/lib/libvirt/images/kvm-deb11.img
kvm-deb11.img: QEMU QCOW2 Image (v3), 10737418240 bytes

```

`qemu-img` コマンドのオプションには他の仮想化ソフトが採用している仮想ディスクの形式と相互変換する `convert` サブコマンドもあります。以下の例は、VirtualBox 向けの VDI 形式から KVM 向けの `qcow2` 形式へ変換するコマンドです。

```
$ qemu-img convert -f vdi disk-virtualbox.vdi -O qcow2 disk-kvm.img
```

### 3.4.2 virt-install コマンド

仮想マシンの OS インストールをシリアル接続したコンソールで操作して行う場合

`virt-install` コマンドを使うと、KVM 上に仮想マシンをインストールすることができます。コマンドライン環境で仮想マシンをインストールする場合は、起動した仮想マシンのコンソールへシリアル接続するようパラメータを指定します（`-graphics none`、`-console` オプション、`-extra-args` オプション）。以下のように `virt-install` コマンドを実行すると、起動した仮想マシンのコンソールで Debian インストーラの処理が始まり、キーボード入力 で仮想マシンへ Debian のインストールを行うことができます。

```
$ sudo virt-install \
--name kvm-deb11 \
--disk path=/var/lib/libvirt/images/kvm-deb11.img,format=qcow2,bus=virtio \
--vcpus 1 \
--ram 1024 \
--os-type linux \
--os-variant generic \
--network bridge=br0,model=virtio \
--location 'http://deb.debian.org/debian/dists/bullseye/main/installer-amd64/' \
--graphics none \
--console pty,target_type=serial \
--extra-args 'console=ttyS0,115200n8 serial'
```

### 仮想マシンの OS インストールを preseed 機能を使って自動インストールする場合

Debian インストーラには設定ファイルにしたがって OS を自動インストールする preseed<sup>\*8</sup> という機能があります。virt-install コマンドのパラメータに preseed の設定ファイルを指定することで仮想マシンへ Debian を自動インストールでき、検証環境の作成や大量に仮想マシンをインストールする場合に便利です。

Debian 11 bullseye 向けの preseed 設定ファイルのサンプルは以下 URL にあります。

- <https://www.debian.org/releases/bullseye/example-preseed.txt>

preseed の設定ファイルを作成し<sup>\*9</sup>、/var/lib/libvirt/images/preseed.cfg としてホストマシンに置きます。そして virt-install コマンドのパラメータに ”-initrd-inject” を指定すると仮想マシンの / 直下に ”-initrd-inject” で指定したファイルがコピーされます。すると、仮想マシン上で起動した Debian インストーラの処理が /preseed.cfg を検出し、シリアル接続した仮想マシンのコンソール上で Debian のインストールが自動で進んでいきます<sup>\*10</sup>。

```
$ sudo qemu-img create -f qcow2 /var/lib/libvirt/images/kvm-deb11-2.img 10g

$ sudo virt-install \
--name kvm-deb11-2 \
--disk path=/var/lib/libvirt/images/kvm-deb11-2.img,format=qcow2,bus=virtio \
--vcpus 1 \
--ram 1024 \
--os-type linux \
--os-variant generic \
--network bridge=br0,model=virtio \
--location 'http://deb.debian.org/debian/dists/bullseye/main/installer-amd64/' \
--initrd-inject=/var/lib/libvirt/images/preseed.cfg \
--graphics none \
--console pty,target_type=serial \
--extra-args 'console=ttyS0,115200n8 serial'
```

### 3.4.3 virsh コマンド

#### virsh とは

virsh コマンドは Debian では libvirt-clients パッケージで提供されており、仮想マシンを制御する様々なハイパーバイザーに対応するコマンドラインのフロントエンドツールです。Linux では KVM や Xen のハイパーバイザーの操作に virsh を利用することができます。

virsh は、”virsh {サブコマンド}” のような形のコマンドを実行することで仮想マシンを制御できます。

#### virsh help

”virsh help” コマンドを実行すると、サブコマンドの一覧が表示されます。

<sup>\*8</sup> 付録 B preseed を利用したインストールの自動化 <https://www.debian.org/releases/bullseye/amd64/apb.ja.html>

<sup>\*9</sup> 詳しくは、第 144 回東京エリア Debian 勉強会「preseed で Debian を自動インストール」を参照。

<sup>\*10</sup> 使った preseed の設定ファイルは <https://github.com/dictoss/utills/blob/master/debian-preseed/preseed-debian11.cfg>

```
$ virsh help
グループ別コマンド:

Domain Management (ヘルプのキーワード 'domain'):
attach-device      XML ファイルによるデバイスの接続
attach-disk        ディスクデバイスの接続
attach-interface   ネットワークインターフェースの接続
autostart          ドメインの自動起動
blkdeviotune       ブロックデバイスの I/O チューニングパラメーターの設定・取得
blkiothtune        ブロック I/O パラメーターの取得・設定
blockcommit        ブロックのコミット操作の開始
(以下、省略)
```

また、”virsh help { サブコマンド }” を実行すると、サブコマンドの使い方とオプションを調べることができます。

```
$ virsh help start
名前
start - 停止状態の (定義済み) ドメインの起動

形式
start <domain> [--console] [--paused] [--autodestroy] [--bypass-cache] [--force-boot] [--pass-fds <string>]
(以下、省略)
```

## virsh -connect

KVM のハイパーバイザーに接続するには ”virsh -connect” を実行します。接続先を示す ”qemu:///system” は URI 形式になっており、接続に利用するプロトコルを切り替えたり、リモート先のホストを指定することができます。以下の例は、KVM のハイパーバイザーが動作しているホストマシン上からローカル接続する場合のコマンドです。

```
$ virsh --connect qemu:///system
virsh によろこそ、仮想化対話式ターミナルです。

入力: 'help' コマンドのヘルプ
      'quit' 終了

virsh #
```

”virsh -connect” で KVM のハイパーバイザーへ接続すると以降は対話的な操作に移ります。接続中はサブコマンドのみで操作が可能になります。

## list サブコマンド

list サブコマンドは、仮想マシンの一覧を出力することができます。オプションの指定値で一覧に表示する仮想マシンが変わります。

単に ”list” と実行した場合は起動中の仮想マシンのみ表示し、”list -inactive” と実行した場合は停止中の仮想マシンのみを表示し、”list -all” と実行した場合はすべての仮想マシンを表示します。

```
virsh # list
Id 名前      状態
-----
1   kvm-deb11 実行中
```

```
virsh # list --inactive
Id 名前      状態
-----
-   kvm-deb11-2 シャットオフ
```

```
virsh # list --all
Id 名前      状態
-----
1   kvm-deb11 実行中
-   kvm-deb11-2 シャットオフ
```

## virsh edit

edit サブコマンドは、仮想マシンの設定ファイル (XML 形式) を開き、編集することができます。edit サブコマンドを実行すると環境変数 EDITOR で指定しているエディタで XML ファイルを開きます\*11。

```
virsh # edit kvm-deb11

<domain type='kvm'>
  <name>kvm-deb11</name>
  <uuid>6dec70d3-09e5-40c6-b269-eaf20a2651d9</uuid>
  <memory unit='KiB'>1048576</memory>
  <currentMemory unit='KiB'>1048576</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <os>
    <type arch='x86_64' machine='pc-i440fx-5.2'>hvm</type>
    <boot dev='hd' />
  </os>
  (以降、省略)
```

## start サブコマンド

start サブコマンドは、仮想マシンを起動することができます。start サブコマンドを実行すると仮想マシンはバックグラウンドで起動します。

```
virsh # start kvm-deb11
Domain 'kvm-deb11' started
```

”start -console” オプションを指定すると仮想マシンの起動と同時に仮想マシンのコンソールにシリアル接続します。

```
virsh # start --console kvm-deb11
Domain 'kvm-deb11' started
Connected to domain 'kvm-deb11-2'
Escape character is ^] (Ctrl + ])

(以降、コンソールに OS の起動ログが表示される)

kvm-deb11 login:
```

## virsh console

console サブコマンドは、起動中の仮想マシンのコンソールへシリアル接続することができます。

```
virsh # console kvm-deb11
Connected to domain 'kvm-deb11'
Escape character is ^] (Ctrl + ])

kvm-deb11 login:
```

## virsh destroy

destroy サブコマンドは、仮想マシンを強制的に停止することができます。なお、stop サブコマンドはありません。

```
virsh # destroy kvm-deb11
Domain 'kvm-deb11' destroyed
```

## virsh undefine

undefine サブコマンドは、仮想マシンの登録を削除します。undefine サブコマンドにオプションを指定せずに実行した場合は仮想マシンに接続していた仮想ディスクは削除されずに残ります\*12。

\*11 XML ファイルの定義は <https://libvirt.org/formatdomain.html> を参照。

\*12 仮想ディスクも一緒に削除したい場合は、-storage {string}、-remove-all-storage などのオプションが利用できます。

```
virsh # undefine kvm-deb11
Domain 'kvm-deb11' has been undefined
```

## virsh define

define サブコマンドは、仮想マシンの構成を定義した XML ファイルから仮想マシンを登録できます。

```
virsh # define /etc/libvirt/qemu/kvm-deb11-3.xml
```

## 3.5 おわりに

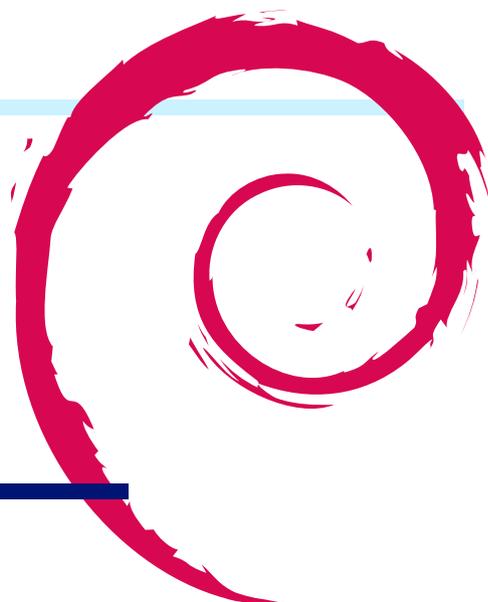
Debian 上で KVM を使えるように設定し、virt-install コマンドや virsh コマンドを説明しました。GUI から操作するより CLI から操作した方が便利な場合もありますので、有効に活用してみてください。

## 3.6 参考文献

- Home - linux-kvm.org [http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page)
- KVM - Debian Wiki <https://wiki.debian.org/KVM>
- 第 144 回東京エリア Debian 勉強会「preseed で Debian を自動インストールをしてみよう」 <https://tokyodebian-team.pages.debian.net/pdf2016/debianmeetingresume201610-presentation-sugimoto.pdf>

## 4 Debian パッケージング repack 編そのソフトウェア、そのまま配布して大丈夫だっけ

Kentaro Hayashi



本記事は Debian 勉強会 (2022 年 11 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。

資料については、Rabbit Slide Show<sup>\*13</sup>にて公開されています。

### 4.1 配布可能なパッケージにするために必要なこと

Debian のパッケージとして配布 (とくに main セクションで) できるようにするためには、まず自由なソフトウェア・不自由なソフトウェアのいずれかであるかをきちんと把握する必要があります。これは、不自由なソフトウェア由来のパッケージについては、パッケージのバージョンでその由来が区別できるようにする必要があります。<sup>\*14</sup>そのため、再パッケージングという課題を避けて通れません。本記事では、そのあたりを解説しつつ、再パッケージングのお作法について説明します。

### 4.2 パッケージングしたくなるような Debian(たまに) あるある

Debian を使っていて、使いたいソフトウェアがまだ apt でインストール不可という状況に出くわすことがあります。<sup>\*15</sup>

そのような場合、手元でだけ使えればよければローカルでビルドして使いますし、需要がありそうであれば「そうだ、パッケージングしよう!!」となります。自由なソフトウェアであれば、肅々とパッケージ化すればいいだけなので、今回の話題の対象外です。そうでなく、(Debian 的には) 不自由なソフトウェアである場合には扱いに注意が必要です。

### 4.3 自由なソフトウェアでない場合に関する注意

いわゆる自由でないソフトウェアといっても、いろんなパターンがあります。自由でないのが主要構成部分であれば、さすがにどうしようもないので諦めざるを得ません。upstream に交渉してライセンスを自由なものに変更してもらうという最終手段もありますが、あまり現実的ではないでしょう。<sup>\*16</sup>

<sup>\*13</sup> <https://slide.rabbit-shocker.org/authors/kenhys/tokyodebian-repack-howto-202211/>

<sup>\*14</sup> +dfsg ないし +ds がついているものが該当する。

<sup>\*15</sup> 筆者は sid を使っているので、安定版にくらべるといちはやくパッケージを試せるというメリットがありますが、それでも皆無ではありません。

<sup>\*16</sup> もちろん upstream にメリットがあることを納得してもらえるのであればよいのですが、そのための交渉にかかる時間や手間など、困難が伴います。もともとライセンスが不明瞭だったなら、交渉の余地はあるかもしれませんが。。。

一方、自由でないのが一部である場合には、まだ対処のしようがあります。例えば、rapidjson はバンドルしているソフトウェアに自由でない JSON ライセンス<sup>\*17</sup>なものを含んでいました。したがってそのままでは自由でないため配布できません。そのため、問題のある部分を除去して配布できるようにしています。

#### 4.4 再配布できるようにしたソフトウェアの見分け方

再配布できるように再パッケージングできるようにしたものは、通常のパッケージとは異なり、パッケージのバージョンに +dfsg や +ds といったサフィックスが付与されています。

このあたりのルールの詳細については、2021 年 12 月の Debian 勉強会で<sup>\*18</sup>解説しているので参考にしてみてください。

パッケージのバージョンに付与されるサフィックスをざっくり説明すると次のようになっています。

- +dfsg
  - DFSG<sup>\*19</sup>に適合しないものを除去した場合に付与します。例: rapidjson-dev 1.1.0+dfsg2-7.1
- +ds
  - DFSG との適合云々とは別の理由で除去した場合に付与します。例えば、バンドルしているソフトウェアがすでに Debian でパッケージ化されている場合など、バンドルしているソフトウェアが不要なため削除したときに使います。

#### 4.5 再パッケージングのしかた

サフィックスのつけかたのルールがわかったところで、再パッケージングのお作法について説明します。まず、再パッケージした次のようなソースアーカイブが必要になります。

- (パッケージ名)\_(バージョン)+dfsg.orig.tar.gz
- (パッケージ名)\_(バージョン)+ds.orig.tar.gz

DFSG に準拠したものなら +dfsg なものが必要ですし、DFSG 云々が関係ないなら +ds なアーカイブを用意する必要があります。

では、+dfsg や +ds なアーカイブはどうつくるのがよいのでしょうか。upstream がバージョンアップしたら、それに追従して再パッケージングを毎回しないといけません。実はそのあたりを便利にしてくれるしくみがあります。それが repack です。

#### 4.6 元のアーカイブから repack して除去するには

次の 2 つを実施すると、元のアーカイブから不要なファイルを除去したアーカイブを作成できます。

- debian/copyright に Files-Excluded: を書く
- debian/watch で repack ルールを記述する

debian/rules に get-orig-source ルールを書いて repack しているパッケージもありますが、それは古いやり方です。あるいは、debian/以下に再パッケージングするための専用のスクリプトを用意しておいて、アーカイブを作成するというやりかたをしているパッケージもありますが、よほどの理由がなければ素直に repack の仕組みにのっかったほうがよいでしょう。

<sup>\*17</sup> <https://wiki.debian.org/qa.debian.org/jsonevil>

<sup>\*18</sup> 参考: lintian に新規ルールを追加する方法 <https://slide.rabbit-shocker.org/authors/kenhys/tokyodebian-lintian-howto-202112/>

<sup>\*19</sup> [https://www.debian.org/social\\_contract](https://www.debian.org/social_contract)

## 4.7 Files-Excluded: について

Files-Excluded: の仕様やドキュメント化に関しては、バグ報告がなされているのでそちらを参照するとよいでしょう。

- <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=685506>
- <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=1000771>

このあたりについては、Debian Wiki にも `uscan` の拡張<sup>\*20</sup> ということでもとまっているのであわせて参照してみてください。

## 4.8 debian/copyright にどう書くか

debian/copyright に Files-Excluded: を次のようなルールで記述します。

```
Files-Excluded:
(削除対象のファイル・ディレクトリにマッチするパターンを列挙)
Comment:
(削除対象のファイル・ディレクトリに関する説明)
```

Comment: 部分は再パッケージングされるファイルそのものへは影響しないという意味では記述しなくてもよいと思うかもしれませんが、ただ、あとあとのメンテナンスの観点からは、なぜ削除したかについて明記しておくのがよいでしょう。

groonga-12.0.9+dfsg-1 の例を抜粋して以下に示します。

```
Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: Groonga
Upstream-Contact: Groonga Project <packages@groonga.org>
Source: http://packages.groonga.org/source/groonga/
Files-Excluded:
doc/locale/en/html/*.html
doc/locale/en/html/_static
doc/locale/en/html/searchindex.js
doc/locale/ja/html/*.html
doc/locale/ja/html/searchindex.js
doc/locale/ja/html/_static
vendor/rapidjson-1.1.0
Comment: The following files were removed because
doc/locale/en/html/*.html: regenerated during building package
...
Files: *
```

なお、Files-Excluded: を記述するにあたっては、必ずヘッダパラグラフに書かないといけません。

つまり、以下のように改行入れると機能しなくてはなりません。

```
Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: Groonga
Upstream-Contact: Groonga Project <packages@groonga.org>
Source: http://packages.groonga.org/source/groonga/

Files-Excluded:
doc/locale/en/html/*.html
...
vendor/rapidjson-1.1.0
Comment: The following files were removed because
doc/locale/en/html/*.html: regenerated during building package
...
Files: *
```

## 4.9 debian/watch をどう書くか

debian/copyright に Files-Excluded: を記述できたら、次は debian/watch に repack ルールを書きます。

debian/watch には、`repack` と `repacksuffix+=dfsg` を指定するというのがポイントです。さらに何度も `repack` してリリースし直す場合には、`repacksuffix+=dfsg2` というように増えていきます。

<sup>\*20</sup> <https://wiki.debian.org/UscanEnhancements>

debian/watch のサンプルを以下に示します。

```
version=4
opts=\
repack,\
dversionmangle=auto,\
repacksuffix=dfsg,\
pgpsigurlmangle=s/\/.asc/ \
https://packages.groonga.org/source/@PACKAGE@/@PACKAGE@-@ANY_VERSION@@ARCHIVE_EXT@
```

## 4.10 パッケージングの具体的な流れについて

ここまで準備できたら、実際にどのようなパッケージングの流れになるのかを説明します。なお、これまでに説明した repack のための準備がすべて整っているものとします。

まずは新しいバージョンをインポートするところからはじめます。

```
$ gbp import-orig -uscan
```

import-orig を実行することにより、次の成果物が得られます。

- +dfsg なアーカイブが作成される
- upstream ブランチにインポート
- upstream/(バージョン)+dfsg タグ付与
- pristine-tar ブランチに (パッケージ)\_(バージョン)+dfsg.orig.tar.xz.(delta,id) が作成される

次に、gbp dch で changelog を更新します。

```
$ gbp dch
```

他にもパッチを更新するなど、debian/配下をもろもろ整備します。

ここまでできたら、あとは gbp buildpackage を使って build-area 配下に生成されたソースパッケージに署名します。

```
$ debsign
```

署名できたら、あとは通常のメンテナンス作業同様に dput でアップロードすればよいだけです。

## 4.11 repack をやりなおす場合

もし、repack で期待通りにいかなかった場合、次のような流れで作業をやりなおすことになります。

- upstream ブランチリセット
- pristine-tar リセット
- git tag -d upstream/(バージョン)+dfsg
- gbp import-orig やり直し

gbp pq rebase で盛大にコンフリクトしたり、うっかり patch-queue 飛ばして泣くはめになることもあるので慎重に作業する必要があるでしょう。

## 4.12 さいごに

一部自由でないソフトウェアを Debian で配布できるようにするための repack というしくみについて解説しました。

- 部分的に不自由なソフトウェアを再配布可能にするには
  - d/copyright で Files-Excluded: を使え

– d/watch で repack,repacksuffix を記述

という知見をいづれ活用してもらえると嬉しいです。

## 5 BoF 相談・情報共有・もくもく会

参加者全員



### 5.1 実施日・場所

- 実施日：2022-06-18(土) 14:00-16:00
- 場所：オンラインによるビデオ会議 (Jitsi Meet)

### 5.2 参加者

- dictoss
- yy-y-ja.jp
- yosukesan
- knok
- ipv6waterstar
- metalefty
- ooonduke

#### 5.2.1 情報共有したこと

##### Debian 勉強会をリアル開催に戻すか

- COVID-19 前は、東京 Deiban 勉強会では荒川区の会議室を借りて開催していた
  - <https://machiya.acc-arakawa.jp/>
  - まだ COVID-19 対策をすること、参加者名簿の提出などが会議室を借りる条件に入っている
- オンラインになって東京、関西以外から参加している人もいるため、リアル開催 + オンラインの形で運営してほしいとの要望あり
  - 企業様の会場を借りて高速なインターネット回線があればリアル開催 + オンラインの開催はできそう
  - 会場が公民館の場合はインターネット回線がないことが多くスマートフォンなどのテザリングでビデオ会議をすることになる。運営者の設備負担が高くなり、ネットワークの安定度が下がりビデオ会議が通信エラーになる課題もある
- もう少しオンラインのみの開催で様子見する

##### Debian Mini Conf (Reunion) in ハンブルク

- アーカイブ (動画)
  - <https://meetings-archive.debian.net/pub/debian-meetings/2022/Debian-Reunion-Hamburg/>
- 講演タイトルを見て
  - aftp: TFTP 関連の話
    - \* サーバ利用では docker や k8s に駆逐されたように思える。組み込みはまだ使っている？

## Debconf

- イスラエルで開催する話は COVID-19 の影響でなくなった模様
- Debconf 2022 コソボ July/17- July/24
  - リアル開催するようだが、開催までこぎつけられるか？
- Debconf23 はインド

## GPD pocket3 を買って Debian をインストールして利用中

- ハードウェアが動作しない悩みあり
  - bullseye をインストールしている状態
  - Wi-Fi、Intel AX201 を搭載しているが、Wi-Fi が動作しないとのこと
  - Bluetooth も動作しない
  - sid で Wi-Fi が動いたことは確認したが、stable (bullseye) で動かしたい
- トラブルシュート
  - Windows で見たデバイスマネージャ情報
    - \* <https://pc.watch.impress.co.jp/docs/column/nishikawa/1400282.html>
  - Intel Wi-Fi 6 AX201 は linux-5.2 以降で動きそうな気がする
    - \* <https://wireless.wiki.kernel.org/en/users/drivers/iwlwifi>
  - bullseye は linux-5.10 のため動作しそうに思うがなぜだろうか。non-free の firmware をインストールしていないのかもしれない。
  - nonfree の firmware-iwlwifi、bullseye-backports の linux-image-5.16 をインストールすると Wifi が認識した！
  - Bluetooth も認識するようになった (bluetooth マウスが使えるようになった)

## NVIDIA GPU の OSS ドライバがリリース

- <https://developer.nvidia.com/blog/nvidia-releases-open-source-gpu-kernel-modules/>
  - NVIDIA が OSS の GPU ドライバをリリースした
  - Red Hat、SUSE、Ubuntu で使えるようになっているらしい
  - debian の unstable ではまだパッケージがないように思える
  - なにか情報を知っている方、いますか
    - \* linux のグラフィックボードは動かしやすい Intel 系 GPU を選んでいたため、あまり NVIDIA は経験がない

## GPU 関連

- GPU は一時期高かったが、最近価格が落ち着いてきた
- Intel Arc という外付けの GPU を出すらしい
  - <https://pc.watch.impress.co.jp/docs/news/1417442.html>

## docker hub のイメージ

- debian の slim イメージは普通のイメージと比べて何を削っている？
- alpine はとてもサイズが小さいので、Debian のイメージでも学んで小さくできないか？

## Mac

- 身近な人で M1 で Linux を動かした人はいたか？
- M2 Mac が近日発売される
  - 買った人は誰か動かすか試してほしい

## python-3.11 では、2 倍早くなるとの噂

- [https://news.mynavi.jp/techplus/kikaku/python\\_learn-10/](https://news.mynavi.jp/techplus/kikaku/python_learn-10/)
- 現在 upstream は python3.11b3 のため、bookworm には入りそうな予感

## 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 2022/06 ~ 2022/11 まで。

内容は無保証、つっこみなどがあれば勉強会にて。



あんどきゅめんでっど でびあん 2023 年夏号

2023 年 8 月 13 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>

