



体験版

うぶんちゅ!
まがじん
ざっぱ～ん♪

vol.8

うぶんちゅ！ まがじん ざっぱ〜ん♪ vol.8 【体
験版】

team zpn 著

2018-06-24 版 team zpn 発行

表紙イラスト：よかぜ
装幀：team zpn

目次

第 1 章	Ubuntu では始める楽しいゼミ運営	1
1.1	メッセージングの悩み	1
1.2	LINE をつけたゼミ運営の問題点	1
1.3	そこで Mattermost	2
1.4	Mattermost Server のインストール	2
1.5	ゼミでの運用	4
1.6	問題は解決したか？	5
第 2 章	ポメラ DM200 に Ubuntu をインストールする	7
2.1	ポメラのハードウェア仕様	7
2.2	バックアップと Debian のインストール	8
2.3	Ubuntu Base のカスタマイズ	8
2.4	Ubuntu 用 SD カードの準備	13
2.5	起動直後の設定	14
2.6	普段の利用方法	15
2.7	ポメラカーネルのカスタマイズ	15
第 3 章	Boomaga を使って PDF を小冊子印刷する方法	17
3.1	ピンチは突然訪れる	17
3.2	概要	18
3.3	インストール	18
3.4	印刷	19
3.5	印刷後	23
3.6	Ubuntu に対応したレーザープリンター	23
第 4 章	Nano Pi NEO で作成するテレビ視聴環境	24
4.1	経緯	24
4.2	必要なもの	24
4.3	armbian のインストール	25
4.4	PX-S1UD の初期設定	26
4.5	Mirakurun のインストール	27
4.6	視聴環境の整備	28
4.7	Chinachu との連携	29
第 5 章	いつでも始められる mpv	31
5.1	mpv って？	31
5.2	インストール	31
5.3	使ってみよう	32
5.4	カスタマイズ	33
5.5	その他	35
5.6	おわりに	38
第 6 章	らくごうさんちのノート PC 事情	39
6.1	最近の Let's note	39
6.2	ノートパソコンを買い替えた	40
6.3	右クリックができない	40

6.4	Type-C から映像端子への変換	41
6.5	充電のあれこれ	42
	USB Type-C と USB PowerDelivery と Thunderbolt 3	43
6.6	まとめ	43
第 7 章	Ubuntu で心理学実験	44
7.1	はじめに	44
7.2	PsychoPy とは	44
7.3	なぜ Ubuntu なのか	45
7.4	実験環境の構築	45
7.5	インストールの手順	45
7.6	もしかしたら	46
7.7	デモコード	46
7.8	おまけ ～spyder と jupyter notebook と jupyter lab～	47
第 8 章	ゲストページ	49
8.1	国際イベントの招致を手伝ってみましたよ	49
8.2	技術書典 4 で冊子版『ざっくりわかる Ubuntu 18.04 LTS』を頒布できなかった顛末	54
	あとがき	58
	著者紹介	59
	「うぶんちゅ! まがじん ざっぱ〜ん♪」バックナンバー	60

第 1 章 Ubuntu では始める楽しいゼミ運営

おしえたかし (mstdn.jp/@oshie)

大学のおしごとをしていて悩ましいのが、ゼミ生さんとのコミュニケーションです。Ubuntu と Mattermost で、オンラインでやりとりする方法を模索してみました。

1.1 メッセージングの悩み

もともと、オンラインでのゼミ生とのコミュニケーションには携帯電話のメールを使っていました。メールを送る時間帯にさえ気をつければ、たいていの場合メールで連絡を取ることができました。

ところが最近では、学生さんはメールをすっかり使わなくなってしまいました。その理由は、迷惑メールがあまりに多いことと、LINE が普及したことと筆者は見ています。メールを送ってもそもそも届かないことも多いです。おそらく、迷惑メールが多いのでアドレスを変更するものの、常用するのは LINE であり、メールはそもそも使っていないので新しいアドレスを連絡しないというようなことが多いのではないかと思います。メールの送り方を知らない学生¹も増えました。余談ですが、LINE はプロプライエタリであり、迷惑メールのせいで自由で開かれた手段が使われなくなるというのはきわめて不幸なことだと思います。

1.2 LINE をつけたゼミ運営の問題点

そこで筆者もしぶしぶ LINE を導入し、ゼミの LINE グループを作成して、各種連絡をそこに流すようにしました。

LINE は以下の点で優れたメッセージングサービスです。

1. デフォクトスタンダード。誰でも使っているの、インストールやアカウント作成等で手間取らない。
2. 豊富なスタンプ。筆者は孤独のグルメのスタンプ²が好きです (どうでもいい)。
3. 豊富 (すぎる) 機能。たとえば、LINE スケジュール³は、ゼミの飲み会を設定するとき等に重宝します。

LINE で連絡がすぐに取れるようになって、これはこれでよかったのですが、弊害もいくつか出てきました。以下に 5 つの問題点を挙げます。

1.2.1 1. 議論が成立しない

LINE のグループには返信機能がありません。そのため、各メンバーの各発言が、誰のどの発言に対するコメントなのかわかりません。会話の流れの中に複数のトピックがある場合、誰が何の話をしているのかさっぱりわからず、とてもカオスなことになってしまいます。また LINE にはスタンプがあり、スタンプは議論の文脈を壊してしまいます。スタンプそのものはとてもおもしろく、筆者もプライベートでよく使いますが、スタンプで埋めつくされると何の話だったかさっぱりわからなくなってしまいます。

というか、世の人はよくこんな不便なサービスでやりとりできますね……。大人でも混乱するのに、子ども同士でこんなのを使ったら LINE いじめが起こるのも必然な気がします。

¹ <https://twitter.com/oshie/status/552401816705826816>

² <https://store.line.me/stickershop/product/8506/ja>

³ <https://schedule.line.me/>

1.2.2 2. 強力すぎる通知

LINE のデフォルトの設定では、発言ごとに通知が届きます。通知の方法も押しつけがましいものがあり、スマートフォンの画面がオフになっていても、別のアプリを開いていても、メッセージがポップアップされます。これはかなり強力な通知ルールだと思います。必ずしも通知の必要のない、どうでもいい発言も、このような方法で通知されてしまいます。

ゼミのグループとはいえ、学生間の交流手段の 1 つとして、しょもない雑談で盛り上がってもよいと思うのですが、このような事情からか、ほとんど雑談が発生しません。筆者からの連絡事項ばかりが流れるグループになってしまっています。

1.2.3 3. 書式の設定ができない

筆者のゼミでは統計解析を行うことが多く、R⁴のコードレビューの機会も多いです。LINE は書式の設定ができないため、コードをペーストされてもとても見づらいものがあります。スマートフォンで撮影した写真を送ってくる学生さんもいましたが、確認のためいちいち打ち直さなければならず、きわめて非生産的です。

1.2.4 4. Ubuntu での使用

Ubuntu で LINE を使う場合、Chrome(ium) の拡張機能を使うか、Wine で Windows 版のクライアントを動かすかの 2 択になると思います。Wine は日本語入力で苦勞することも多いですし、なるべく避けたいところです。Chrome(ium) の拡張機能は一工夫すれば Dock に登録できる⁵ようですが、そもそも筆者のデフォルトブラウザは Firefox であり、わざわざ LINE のためだけに消費リソースを増やしたくありません。

PC 上ではブラウザで利用できれば、上記のような問題は起こりません。というか、独自のアプリを入れないと使えないサービスとか不便すぎるでしょ……。

1.2.5 5. プロプライエタリ

筆者のゼミの教育方針⁶の 1 つは「自由で開かれている風土を目指すこと」であり、プロプライエタリな LINE を使うことは一貫性に欠けます。可能な限り、FLOSS を使いたいところです。

1.3 そこで Mattermost

そこで代替サービスとして、Mattermost⁷に白羽の矢を立てました。Mattermost は Slack⁸の FLOSS クローンです。Slack を利用してもよかったのですが、Slack はプロプライエタリです。上記 1.~5. すべての問題を解決するためにも、FLOSS である Mattermost を使用することにしました。

1.4 Mattermost Server のインストール

筆者は ServersMan@VPS Entry プランで Ubuntu 14.04 LTS 環境を利用しています。Mattermost サーバーを以下の要領でインストールしました。といっても、マニュアル⁹どおりですが、16.04 以上の環境の方は別のガイド¹⁰を参照してください。また「password」には適切な文字

⁴ <https://www.r-project.org/>

⁵ <https://qiita.com/k-seta/items/08c1600d6993104f258c>

⁶ <http://oshie.edu.yamaguchi-u.ac.jp/index.php/%E6%95%99%E8%82%B2%E6%96%B9%E9%87%9D/>

⁷ <https://about.mattermost.com/>

⁸ <https://slack.com/intl/ja-jp>

⁹ <https://docs.mattermost.com/install/install-ubuntu-1404.html>

¹⁰ <https://docs.mattermost.com/install/install-ubuntu-1604.html>

列を、「hostname」には適切なホスト名を設定してください。

```
$ sudo apt-get install mysql-server-5.6
$ mysql -u root -p
mysql> create user 'mmuser'@'%' identified by 'password';
mysql> create database mattermost;
mysql> grant all privileges on mattermost.* to 'username'@'%' ;
mysql> exit
$ wget https://releases.mattermost.com/4.6.1/mattermost-4.6.1-linux-amd64.tar.gz
$ tar xvf mattermost-4.6.1-linux-amd64.tar.gz
$ sudo mv mattermost /opt
$ sudo mkdir /opt/mattermost/data
$ sudo useradd --system --user-group mattermost
$ sudo chown -R mattermost:mattermost /opt/mattermost
$ sudo chmod -R g+w /opt/mattermost
```

「sudo vim /opt/mattermost/config/config.json」などとして、config.json のうち 2 箇所を設定します。まず”DriverName”は”mysql”に設定してください。次に”DataSource”を次のように変更します。このうち password は先ほど設定したものと同じです。

```
”mmuser:password@tcp(localhost:3306)/mattermost?charset=utf8mb4,utf8&readTimeout=30s&writeTimeout=30s”
```

Upstart のデーモンとして登録しておきましょう。「sudo vim /etc/init/mattermost.conf」などとして以下の内容を入力します。

```
start on runlevel [2345]
stop on runlevel [016]
respawn
limit nofile 50000 50000
chdir /opt/mattermost
setuid mattermost
exec bin/platform
```

「sudo start mattermost」してサーバーを起動します。ブラウザで http://hostname:8065/ を開いてみましょう。

指示にしたがってチームとユーザーを作成します。ログインしたら、左ペインのハンバーガメニュー（三）をクリックし、[Account Settings] を開きましょう。[Display] → [Language] とクリックし、[日本語] を選んで [Save] をクリックします。これでメニューが日本語で表示されます。次に、同じくハンバーガメニューから [システムコンソール] を開きましょう。

- [全般] セクション内の [設定]
 - [サイト URL] を [http://hostname:8065] に。
 - 左下の [保存] をクリックしておきます。
- [全般] セクション内の [言語]
 - [デフォルトのサーバー言語] を [日本語] に。
 - [デフォルトのクライアント言語] を [日本語] に。
 - 左下の [保存] をクリックしておきます。
- [通知] セクション内の [電子メール]
 - [通知電子メールを有効にする] を [有効] に。
 - [通知電子メールでの表示名] を [No-Reply] に。
 - [通知電子メールでの電子メールアドレス] を適切なメールアドレスに。筆者は面倒なので筆者の大学のメールアドレスを入力しています。
 - [SMTP サーバー] を入力。筆者は面倒なので筆者の大学の SMTP サーバーを利用しています。あわせて [SMTP サーバーポート]、[SMTP 認証を有効にする]、[SMTP サーバーの

第 2 章 ポメラ DM200 に Ubuntu をインストールする

柴田充也

「理想のモバイル端末」は有史以来人類が追い求めている夢のひとつです。粘土板やパピルス・竹簡・木簡が紙になり、電子デバイスへと移行したあとも、大きな画面と小さく軽い筐体、高速な処理能力と長い動作時間という相反する目的を常に追い求めてきました。人によって重視するポイントが異なることから、技術が進歩するごとにさまざまなデバイスが現れては消えています。今回はそんなデバイスの「現時点における解」のひとつであるポメラ DM200 に Ubuntu 18.04 LTS をインストールしてみましょう。

2.1 ポメラのハードウェア仕様

ポメラシリーズはキングジムが販売する「テキスト入力専用マシン」です。無理のないキーピッチと配列のキーボード、高速な起動、シンプルなメモアプリと最強の日本語入力システム ATOK を擁することで、2008 年発売の初代 DM10 から 2017 年の DM200 や 2018 年の DM30 に至るまで、主に執筆を生業とするユーザーに対して大ヒットするシリーズに成長しました。

「テキスト入力専用マシン」と名乗っていることからわかるように、デバイスの目的を「執筆」に絞っているため、一般的なモバイルデバイスとは趣きを異にしています。しかながら「執筆」に絞っているからこそ、「物書き」とっては非常に魅力的なデバイスになっているのです。

DM200¹は 2017 年に発売された液晶型ポメラシリーズの最新作です。DM200 では大幅に機能が拡充され（あわせてサイズも大きくなり）、Wi-Fi を利用したネットワーク関連の機能も搭載されました。もちろんネットワーク対応はあくまで執筆原稿の同期・転送・バックアップが目的であり、ウェブブラウザのような時間を浪費する悪魔は搭載されていません。他にも先代のポメラと同様の特徴も継承しています。

- ・ 高速な起動とシャットダウン
- ・ 起動したらとにかく問答無用でエディターが起動する
- ・ かな漢字変換システムとして ATOK がプリインストール済み
- ・ 横 17mm ピッチのフルキーボード
- ・ SD カードへの保存

利用者から見ると「高機能化・大型化したポメラ」なのですが、ハードウェア・ソフトウェアを見ると実はただの ARM ベースの Linux 端末だったりします。

- ・ Rockchip RK3128
- ・ ARMv7 1.2GHz クアッドコア（約 900MHz までに制限）
- ・ 512MiB RAM
- ・ eMMC NAND 4GiB（ブートローダー・OS・ファームウェア領域）
- ・ microUSB ポート（受電可・給電不可）
- ・ SD カードスロット
- ・ Wi-Fi IEEE802.11b/g/n（2.4GHz）
- ・ Bluetooth 4.0 + EDR
- ・ 7 インチ 1024×600 液晶
- ・ ブートローダー：U-Boot 2014.10
- ・ OS：Linux 3.10

ポイントは OS として **Linux カーネル**を使っていること、そして書き換え可能な NAND 領域に OS

¹ <http://www.kingjim.co.jp/pomera/dm200/>

がインストールされていることです。

カーネルが Linux である以上、ユーザーランドさえ差し替えれば一般的な Linux ディストリビューションを動かせる可能性が高くなります。さらに NAND が書き換え可能状態であるならば、カーネルを差し替えることすら可能になります。

2.2 バックアップと Debian のインストール

Linux が動いているという話から、一般的な Linux コンソール端末として動かせないか調査し、その成果をまとめてくれたのが@ichinomoto²さんです。

しかもバックアップスクリプトと Debian のインストーラーまで公開されているため、16GB 以上の SD カードを用意すれば誰でも簡単に各自の責任においてポメラをポメラ OS と Debian のデュアルブートにできます。

Debian 用のカーネルと initramfs はリカバリー領域にインストールするため、普通に電源を入れればポメラとして動作し、Alt+ 右 Shift+ 電源ボタンで起動すれば Debian が起動します。つまりひと粒で二度美味しいマシンになるのです。

さて本章ではこのインストーラーをベースに、ユーザーランドのみを Ubuntu 化します。つまりカーネルと initramfs は上記 Debian 版をそのまま流用します。Ubuntu 化しなければならない理由は特にありません。単に普段使っているのが Ubuntu だから Ubuntu にしたいというだけです。GNOME Shell が十分な速度で動くとは言い難い状況なので、デスクトップとして使うのであればむしろ Debian のままで十分でしょう。ここではあくまで CLI としての Ubuntu を用意する方法の説明に特化します。よってデスクトップに必要な設定などは飛ばしているのでご注意ください。

ちなみにカーネルは、Debian 版で使っているカーネルというだけで、Debian 公式のカーネルではありません。Rockchip がメーカーに提供し、キングジムがカスタマイズしたカーネルにいくつかの修正とコンフィグ有効化を行ったカーネルです。

バックアップと Debian 化の方法については@ichinomoto さんの「Linux on Pamera DM200 人柱版 その 2」³を参照してください。その技術的な詳細は「Linux on Pamera DM200」⁴が参考になります。

以降はオリジナルの eMMC NAND のバックアップを行い、Debian 化が完了し、一度でも Debian で正しく起動できたことを前提とします。なお、Debian 版の SD カードのいくつかのデータを流用するため、Ubuntu 化にあたってはもう一枚 16GB 以上の SD カードを用意してください。ただし Ubuntu の CLI 環境だけであれば 8GB でも間に合います。

Ubuntu 化に当たって実際に必要なのは Debian 化したルートファイルシステムの/opt ディレクトリ以下だけです。

2.3 Ubuntu Base のカスタマイズ

ここからは Ubuntu 用 SD カードにインストールするルートファイルシステムを作成します。とりあえず PC 上の任意のディレクトリ以下に作成し、最後に SD カードに保存します。

Ubuntu Base は組み込み機器向けに提供されている最小のルートファイルシステムです。今回は Ubuntu Base をベースに、必要なソフトウェアをインストールしていきましょう。まずは Ubuntu Base のダウンロードと展開です。

```
$ mkdir ~/pomera/ && cd $_
$ wget http://cdimages.ubuntu.com/ubuntu-base/releases/18.04/release/ubuntu-base-18.04-base-armhf.tar.gz
$ mkdir rootfs
$ sudo tar xvf ubuntu-base-18.04-base-armhf.tar.gz -C rootfs/
```

² <https://twitter.com/ichinomoto>

³ <https://www.ekesete.net/log/?p=8940>

⁴ <https://ekesete.booth.pm/items/616812>

2.3.1 rootfs への chroot

rootfs ディレクトリ以下を仮想的なルートファイルシステムとして扱い、apt コマンドでパッケージをインストールします。

一番簡単な方法は chroot コマンドを使うことです。ただしホスト PC のアーキテクチャーが amd64 の場合、そのままではポメラ用の armhf アーキテクチャーのパッケージをインストールできません。そこで QEMU を用いて chroot 時にエミュレーションを行います。特に難しいことをするわけではなく、qemu-user-static パッケージにある qemu-arm-static コマンドを chroot 先の usr/bin 以下にコピーするだけです。

```
$ sudo apt install qemu-user-static
$ sudo cp /usr/bin/qemu-arm-static rootfs/usr/bin/
```

さらに chroot 先でネットワークが動くよう設定ファイルもコピーした上で、chroot コマンドを実行します。

```
$ sudo cp /etc/resolv.conf rootfs/etc/resolv.conf
$ sudo mount -t proc /proc rootfs/proc
$ sudo ln -s ../proc/mounts rootfs/etc/mtab
$ sudo mount --rbind /dev rootfs/dev
$ sudo mount --make-rslave rootfs/dev
$ sudo mount -t sysfs -o nosuid,nodev,noexec sysfs rootfs/sys
$ sudo ln -s /dev/shm rootfs/run/shm
$ sudo chroot rootfs/
```

2.3.2 パッケージのインストールと更新

最初にパッケージリポジトリを設定します。Ubuntu Base の初期状態では universe/multiverse が無効化されているので有効化しておきましょう。

```
# sed -i "s/^# ¥(deb .*bionic.*verse.*¥)/¥1/g" /etc/apt/sources.list
```

また Ubuntu Base ではできるだけイメージサイズを小さくすることを目的として、man ページや changelog ファイルなどをインストールしないように dpkg コマンドの設定が行われています。通常の CLI 環境だと man ページが存在しないのは不便なので、その設定も解除しておきましょう。

```
# /usr/local/sbin/unminimize
```

設定が解除されると同時に ubuntu-minimal パッケージがインストールされます。これにより less や passwd など最低限のコマンドなどがインストールされます。制限の強い環境で生活することに快感を覚えるような人でない限りは必須とってください。ちなみに ubuntu-standard パッケージもインストールすることが一般的ではありますが、サイズが大きくなりがちなので実機上でインストールすることをおすすめします。

次にパッケージリストを更新し、最新のアップデートを適用した上で、必要なパッケージ一式をインストールします。

```
# apt update
# apt full-upgrade
# apt install -y wpasupplicant iw wireless-tools fbterm fonts-noto-cjk fonts-ricity-diminished ¥
uim-fep uim-mozc language-selector-common language-pack-ja
```

第3章 Boomaga を使って PDF を小冊子印刷する方法

いくや

今回は簡単な操作で PDF を小冊子印刷する方法を紹介します。

3.1 ピンチは突然訪れる

皆さんも同人イベントに出展した際に印刷所に発注した冊子が見当たらず、それを見本誌に使うと目論んでいたのでも別途用意しておらず、結局見本誌なしでダウンロード版の販売を行った経験があると思います。

……いや、ないわ。これはないわ。改めて書いてみて思ったわ。

この話はひとまずさておいて、そんな場合でも小冊子印刷を行うことができれば見本誌をサクッと用意することができます。ただ、Ubuntu で小冊子印刷をしようとするとなかなか難しいのです。

Linux 版 Adobe Reader が生きていた頃は機能としてあったのでわりと簡単にできました。LibreOffice Writer にも小冊子印刷機能はありますが、異なった用紙サイズに変更したりとか、そもそも PDF を印刷するような機能はなく、一から書いたもののみ対応します。psbook とかいうコマンドでなんとかできるらしいというのはわかるのですが、この手のことをコマンドラインでやるというのは直感的ではないのでできれば避けたいです。

Evince (ドキュメントビューアー) のヘルプでも一応説明されていますが (図 3.1)、

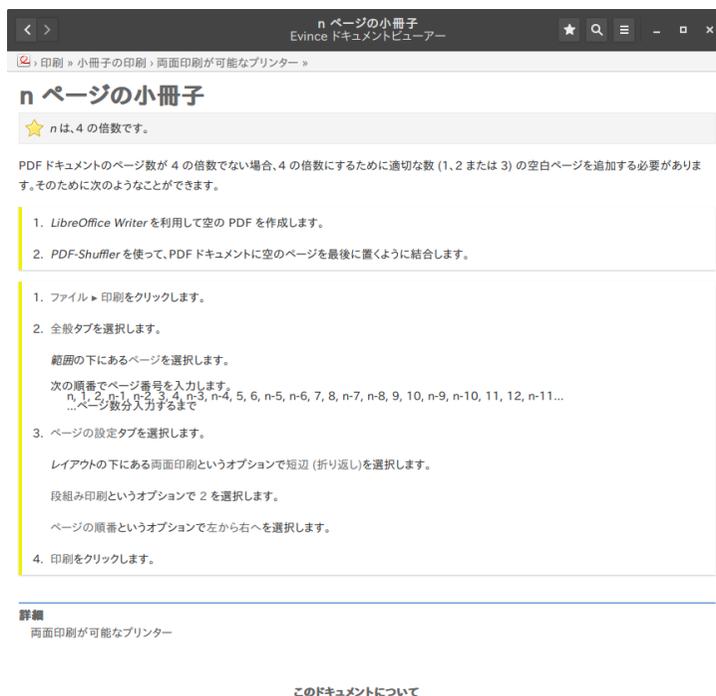


図 3.1: Evince のマニュアルにある小冊子印刷の方法

手で計算しろと申すか……。

というわけで、しょうがないので Windows の Adobe Reader から印刷するという始末でした。

3.2 概要

もっといい方法はないかと長年（やや誇張）検索していたのですが、ついに見つけました。それが Boomaga¹です。

今回は使用しないものの、Ubuntu のリポジトリにもある著名（たぶん）なツールなのに今までどうして引っかけたこなかったのが謎です。まあ筆者の検索キーワードが悪かったのでしょうか。

Boomaga は「メタプリンター」として動作するところが特徴です。アプリケーション（通常はドキュメントビューアーである Evince か Atril でしょう）²でプリンターとして Boomaga を指定し、各種設定を行います。その設定の中には実際に印刷するプリンターも含まれるので、アプリケーションもプリンターも自由に選べるというわけです。CUPS のしくみをうまく使っていて、その発想に惚れ惚れとします。

御託はこのぐらいにして、実際に印刷してみましょう。

対象の Ubuntu のバージョンは 16.04 LTS、17.10、18.04 LTS です。今回は 18.04 LTS で動作を確認しました。

サンプルの PDF は『ざっくりわかる Ubuntu 18.04 LTS』³を使用します。

3.3 インストール

Boomaga は前述のとおり Ubuntu のリポジトリにもありますが、バージョンが古いので PPA⁴からインストールします。次のコマンドを実行してください。

```
$ sudo add-apt-repository ppa:boomaga
$ sudo apt install boomaga
```

今までは add-apt-repository コマンド実行後に「apt update」コマンドの実行が必要でしたが、18.04 LTS からは自動実行されるようになったので省略できます。そんなわけで 16.04 LTS の場合は手動で実行してください。

インストール完了後、Boomaga がプリンターとして追加されます（図 3.2）。

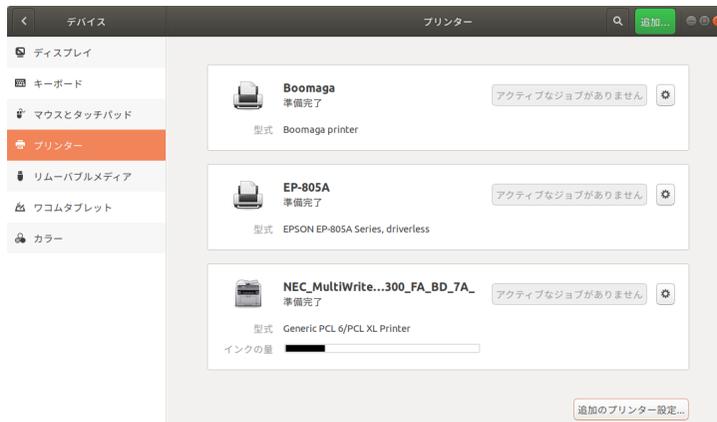


図 3.2: うちのプリンター事情

¹ <https://github.com/Boomaga/boomaga>

² Web ブラウザーからたくさんの Web ページを印刷するにも使えそうですが、それって辛いお仕事の検収用……うっ涙が……

³ <http://zappppaan.freepub.jp/article/183055647.html>

⁴ <https://launchpad.net/~boomaga/+archive/ubuntu/ppa>

3.4 印刷

では実際に印刷してみましょう。まずはドキュメントビューアーで PDF を開き、PDF のプロパティを確認します。というのも、最初の段階で用紙サイズ（ページサイズ）の指定を間違えると余白が多すぎる小冊子が出来上がってしまいます（図 3.3）。



図 3.3: PDF のプロパティ

ページサイズは B5(JIS) であることがわかりました。B5(ISO) と B5(JIS) で微妙にサイズが違うとかトラップですよね……。

それはさておき、いよいよ印刷してみます。選択するプリンターは「Boomaga」です。（図 3.4）「ページの設定」タブを開き、「用紙サイズ」を「B5(JIS)」にします。（図 3.5）

第 4 章 Nano Pi NEO で作成するテレビ視聴環境

ryunuda(@ryunuda)

超小型評価ボードである Nano Pi NEO¹を使用してテレビ視聴環境を構築した記録です。



図: 視聴環境

4.1 経緯

現行使用しているファイルサーバ兼録画サーバのハードウェア (HP MicroServer) 寿命が気になって、NAS へのリプレースを検討していました。

NAS へのリプレースに関して、録画サーバ機能に関しては別な機器へ移動する必要があります。ここで PC を使うことは消費電力 (というより電気代) の観点から得策ではないので、ARM ボードを利用した環境を構築したいと考えていました。

以前、PCIe を持つ ARM ボードである HummingBoard と PT3 でチューナサーバを構築したのですが、HummingBoard の発熱が大きいことや、PT3 自体が終息していることもあり移行に踏み切れませんでした。

何か別な方法がないかを調べていたところ、PX-S1UD²というチューナが、USB 接続で、かつ、2018/06 時点でも販売中であり、また、Raspberry Pi を始めとした ARM ボードと組み合わせた作例も何件も見かけたので PT3 の代替として良さそうだと感じました。そこで、PX-S1UD と ARM ボードを利用したチューナ環境を作成する方向で検討を始めました。

使用する ARM ボードとして Raspberry Pi ではフットプリントが大きいと感じたので、何かもっと小さいものがないかと調べて見たところ、秋葉原の秋月電子でたまたま Nano Pi NEO を発見しました。Nano Pi NEO の大きさは 40mmx40mm で、これならば PX-S1UD の長さを含めても Raspberry Pi の本体と同程度の大きさで環境が構築できそうでした。それでは、実際に実現できるのか試してみようということで、Nano Pi NEO と PX-S1UD の接続を実験することにしました。

4.2 必要なもの

Nano Pi NEO と PX-S1UD を接続する実験をするために以下のものを用意しました。

- ・ Nano Pi NEO 512MB
 - 256 MB メモリ版もありますが、リソースが足りなくなる可能性が高いので 512 MB メモリ版を使用します。

¹ http://www.friendlyarm.com/index.php?route=product/product&product_id=132

² <http://www.plex-net.co.jp/product/px-s1udv2/>

- Nano Pi NEO 2 を選ばないように注意が必要です。NEO 2 は 64bit arm で、ARM 64bit kernel 付属のドライバにはバグ³があり、PX-S1UD の認識に失敗します。
- ・ ヒートシンク
 - 専用のものでなくとも、SoC に貼り付けられる 1.5~2.0cm 角のものでも構わないと思います。ヒートシンクを設置しないことは SoC の発熱を考えるとおすすめできません。
- ・ USB AC アダプター/電源用 USB ケーブル
- ・ PX-S1UD x2
- ・ IC カードリーダー (NTT SCR3310-NTTCom)
 - 筆者が予備として所持していたものを使用しました。
- ・ 同軸ケーブル
- ・ 分配器
- ・ microSD カード (16GB; class10)
 - Nano Pi NEO の OS インストール用です。
- ・ OS インストール用 PC (Ubuntu)
 - 今回は Ubuntu 17.10 Desktop を使用しました。Etcher⁴などのイメージを SD カードへ焼けるツールができれば OS は問いません。
- ・ USB - シリアル変換ケーブル⁵
 - 後述の通り、ない場合の回避策もありますので必須ではありません。

4.3 armbian のインストール

armbian⁶は ARM ボード向けの Debian/Ubuntu ベースのディストリビューションです。Nano Pi NEO にも対応しています。

実際に Nano Pi NEO へ armbian をインストールしていきます。まず armbian から Xenial 用の image をインストール用 PC へダウンロードします。ダウンロードが完了したら、7z を展開するための p7zip-full をインストールして、7z ファイルを展開します。

```
$ sudo apt install p7zip-full
$ 7z e Armbian_5.38_Nanopineo_Ubuntu_xenial_next_4.14.14.7z
$ ls
Armbian_5.38_Nanopineo_Ubuntu_xenial_next_4.14.14.7z
Armbian_5.38_Nanopineo_Ubuntu_xenial_next_4.14.14.img
Armbian_5.38_Nanopineo_Ubuntu_xenial_next_4.14.14.img.asc
```

展開したら、インストール PC へ microSD カードを挿入して、Etcher で image (raw ファイル) を microSD カードへコピーします。

USB-シリアルケーブルがない場合はコピー後に SD カードをマウントさせて、etc/Network/interfaces を書き換えて IP アドレスを静的に設定してしまう作業をしておくこと初回起動時から ssh 接続可能です。

microSD カードへ image を書き終わったら、Nano Pi NEO へ SD カードを挿入し、AC アダプターを接続して電源を投入します。LED 点灯し始めたら、USB-シリアルケーブルを PC へ接続し、PC から screen などでもシリアルコンソールをとります。

```
$ sudo /bin/bash -c 'LANG=C screen /dev/ttyUSB0 115200 -L'
```

初回は root でログインします。パスワードは 1234 です。ログインに成功すると、root のパスワード

³ <https://lkml.org/lkml/2018/3/3/122>

⁴ <https://etcher.io/>

⁵ 例えば <https://www.switch-science.com/catalog/1196/> の様なものです。

⁶ <http://docs.armbian.com/>

ド変更を求められますので任意のものを入力します。その後一般ユーザーの作成を求められるので任意のユーザー名とパスワードを入力します。上記の作業は必須です。なお、この時に作成する一般ユーザーは uid 1000 で、同時に作成される同名の gid 1000 なグループに所属しています。また、sudo グループにも追加されているため追加作業なしで sudo コマンドが実行可能です⁷。今後は root では作業せず、root を「passwd -l」でロックして一般ユーザーのみで作業したほうがセキュリティ的にもよいでしょう。

コンソールが取れるようになったら、PX-S1UD が USB デバイスとして見えているか lsusb コマンドで確認します。lspci して "VidzMedia Pte Ltd" が見えれば正しく PX-S1UD が USB デバイスとして見えています。そうでない場合は変換アダプタの接続が正しくできているか等物理的な接続を確認します。

```
$ lsusb | grep Vidz
Bus 003 Device 005: ID 3275:0080 VidzMedia Pte Ltd
Bus 003 Device 004: ID 3275:0080 VidzMedia Pte Ltd
```

確認ができれば最低限の OS セットアップは完了です。筆者は以下の作業も併せて実施しました。

- パッケージのアップデート
 - ネットワークにさえ繋がっていれば「sudo apt upgrade」でアップデート可能です
- IP アドレスの固定化
- ntp クライアントのインストールと設定
- タイムゾーンの設定 (UTC → JST)

4.4 PX-S1UD の初期設定

armbian のインストールと初期設定が完了したら、PX-S1UD を使用するために必要な初期設定を実施していきます。基本的には x86_64 マシンでの手順と全く同じです。

armbian には DVB 版の PX-S1UD 用ドライバ (smsusb、smsmdtv) がデフォルトで組み込まれていますので、ドライバのインストール作業は不要です。

```
$ lsmod | grep smsusb
smsusb          16384  0
smsmdtv         40960  2 smsdvb, smsusb
```

ファームウェアに関しては組み込まれていないため、Windows 版のものから抜き出して /lib/firmware へ設置します。

```
$ wget http://plex-net.co.jp/plex/px-s1ud/PX-S1UD_driver_Ver.1.0.1.zip
$ unzip PX-S1UD_driver_Ver.1.0.1.zip
$ sudo cp PX-S1UD_driver_Ver.1.0.1/x64/amd64/isdbt_rio.inp /lib/firmware/
```

一度 reboot して、/dev/dvb/adapter[0-9]/が存在していることを確認します。

```
$ ls -l /dev/dvb/adapter0/
demux0
dvr0
frontend0
$ ls -l /dev/dvb/adapter1/
demux0
dvr0
frontend0
```

⁷ Ubuntu のインストール時に追加する一般ユーザーと同じです。

第5章 いつでも始められる mpv

kazken3(@kazken3)

暖かい春も過ぎ、こころ穏やかな日々もつかの間、暑さを感じられる季節が近づいてきました。そんな日々ではありますが、最近良くなってきたと感じている mpv を試してみました。¹

5.1 mpv って？

mpv²は、MPlayer から fork された mplayer2³をもとに作られた、マルチプラットフォームのメディアプレイヤーです。デコードライブラリとして FFMpeg⁴を標準で利用し、現在もアクティブに開発が行われています。

トップページも特徴が挙げられていますが、筆者としては次の内容が特徴的だと思っています。

- ・ OSC⁵
- ・ 高品質のビデオ出力
- ・ FFMpeg を利用した GPU ビデオデコード
- ・ ショートカットがわかりやすい

5.2 インストール

mpv は universe⁶にあり、インストールは簡単です。

```
$ sudo apt install mpv
```

引数なしで起動すると、オプションが表示されます。Ubuntu 18.04 LTS の場合はバージョン 0.27.2 がインストールされます。

```
$ mpv
mpv 0.27.2 (C) 2000-2017 mpv/MPlayer/mplayer2 projects
built on UNKNOWN
ffmpeg library versions:
  libavutil      55.78.100
  libavcodec     57.107.100
  libavformat    57.83.100
  libswscale     4.8.100
  libavfilter    6.107.100
  libswresample  2.9.100
ffmpeg version: 3.4.2-2

Usage:  mpv [options] [url|path/]filename

Basic options:
--start=<time>    seek to given (percent, seconds, or hh:mm:ss) position
--no-audio       do not play sound
--no-video       do not play video
--fs             fullscreen playback
--sub-file=<file> specify subtitle file to use
```

¹ 季節が変わったので挨拶も変えた…

² <https://mpv.io/>

³ mplayer2 はすでに開発は停止しています。

⁴ libav も利用可能ですが ffmpeg が妥当でしょう。

⁵ オンスクリーンコントローラ。画面上にマウスオーバーすると表示されるちいさな GUI です。

⁶ Ubuntu 14.04 LTS から universe で提供されています。

```
--playlist=<file> specify playlist file
--list-options    list all mpv options
--h=<string>      print options which contain the given string in their name
```

「mpv --list-options」で更に細かいオプションを確認することができます。カスタマイズを行う際に参考になるでしょう。

5.3 使ってみよう

5.3.1 再生

基本は、ファイル名を指定すれば再生することができます。再生できるフォーマットは FFmpeg で再生できるものはすべて再生できると考えて良いでしょう。

```
$ mpv movie.mp4
```

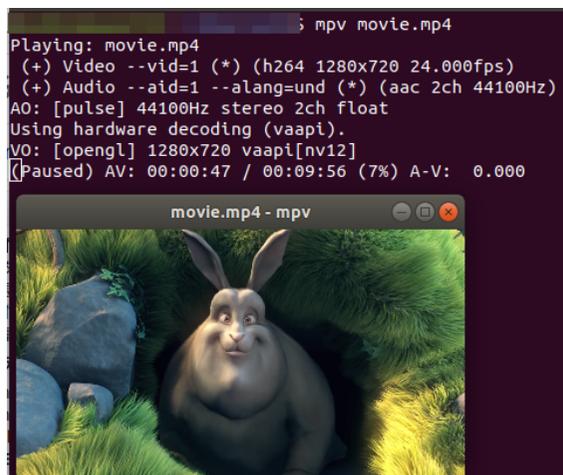


図 5.1: mpv 再生画面

超簡単ですね。^{*7}

5.3.2 ショートカットを使おう

OSC が提供する UI によって、ある程度の GUI 上での利用も可能ですが、mpv ではショートカットを利用することでさらに柔軟な再生を行うことができます。デフォルトでは、ヘルプ^{*8}にある動作となりますが、次に代表的なものだけ挙げておきます。

^{*7} This movie has been licensed under the Creative Commons Attribution 3.0 license(CC BY 3.0).

^{*8} <https://mpv.io/manual/master/#interactive-control>

キー	機能
右左矢印キー	5 秒単位で早送り/早戻し。 Shift キーを併用すると 1 秒単位で行う。
上下キー	1 分単位で早送り/早戻し。 Shift キーを併用すると 5 秒単位で行う。
[] キー	再生速度を 10% 減らす/増やす
{ } キー	再生速度を半減/倍増
バックスペースキー	再生スピードをリセット
p またはスペースキー	ポーズ (再度押すと再開)
. キー	ステップ再生。押すたびに 1 フレーム再生しポーズ
, キー	ステップ戻し。押すたびに 1 フレーム戻しポーズ
q キー	終了
Q キー	終了位置を記憶して終了。同じファイルを再生するときに終了位置から再生
/, * キー	ボリュームを減らす/増やす
9, 0 キー	ボリュームを減らす/増やす
m キー	音声をミュート
f キー	フルスクリーンモード切り替え (トグル)
ESC キー	フルスクリーンモード終了
T キー	常に最前面に切り替え (トグル)
l キー	A-B ループポイントの作成/クリア
L	ループ再生 (トグル)
Ctrl++, Ctrl+- キー	映像と音声のずれを 0.1 秒単位で調整する
s キー	スクリーンショットを取る
Ctrl+s キー	現在のウィンドウサイズでスクリーンショットを取る
d キー	インタレースの ON/OFF
A キー	アスペクト比のサイクル切り替え
1, 2 キー	コントラスト調整
3, 4 キー	輝度調整
5, 6 キー	ガンマ調整
7, 8 キー	彩度調整
Alt+0 キー	映像をオリジナルサイズの半分にリサイズ
Alt+1 キー	映像をオリジナルサイズにリサイズ
Alt+2 キー	映像をオリジナルサイズの 2 倍にリサイズ

ある程度、英語とくくりついていたりするのでわかりやすいと思いますが、キーバインドの好みもあるかと思います。次節ではキーバインドの変更も含めたカスタマイズについて触れます。

5.4 カスタマイズ

5.4.1 設定のカスタマイズ

mpv の設定はキーバインド (次項で触れます) を除き、`/etc/mpv.conf` がシステムワイドの設定となります。設定を変更したい場合は`~/.config/mpv/mpv.conf`を作成しファイルを変更することで、設定をオーバーライドすることができるため、システムワイドの設定を汚すこともなく設定ができます。パッケージからインストールした場合は、`/usr/share/doc/mpv` にてテンプレートとして利用できる `mpv.conf` がありますので、まずはそれを展開してコピーします。

```
$ mkdir -p ~/.config/mpv
$ cd ~/.config/mpv
$ zcat /usr/share/doc/mpv/mpv.conf.gz > mpv.conf
$ vi mpv.conf
```

筆者の場合、次の設定⁹をデフォルトから変更しています。詳細は `OPTION`¹⁰の欄を確認しなが

⁹ HW エンコーダーの設定はデフォルトでは auto ですが、筆者の Sandy Bridge の環境では起動時に「libvdpau_i965.so が無いんですよ」ってエラーが出ているため明示的に `vaapi` を指定しています。ただしこの設定の有無による再生時の差は筆者の環境では出てないので、結果としては精神衛生上入れている感じですね。

¹⁰ <https://mpv.io/manual/stable/#options>

第 6 章 らくごうさんちのノート PC 事情

Rakugou

今まで頑張ってきた Let's note から新しいノートパソコンへ移行することになりましたので、その備忘録です。

6.1 最近の Let's note

ざっぱ〜ん♪ vol.3¹にて Let's note の 2in1 タブレットとして（無理やり）使用方法を紹介しました。執筆当時（Ubuntu のバージョンは 14.04 LTS）はディスプレイを折りたたんでラップトップモードからタブレットモードに移行する際、タブレットモードの状態ではキー入力することができませんでした。そのため、あらかじめラップトップモードで Onboard を起動してからタブレットモードに切り替えるという手法を取っていましたが、17.10 で試してみたところこの作業が不要となりました。キーボードを出現させる方法も至って単純で、画面下から上へスワイプするだけというものです。フリック入力には対応しておりませんが、10 インチ超のディスプレイでは両手親指でタップするのに十分な広さがあります。タブレットモードでは基本的に電子書籍の閲覧など、入力を伴う作業を基本的には行わないことが想定されます。そのため、スマートフォンではフリック入力を常用する筆者でも、QWERTY 配列だけで十分ではないかと思えます。

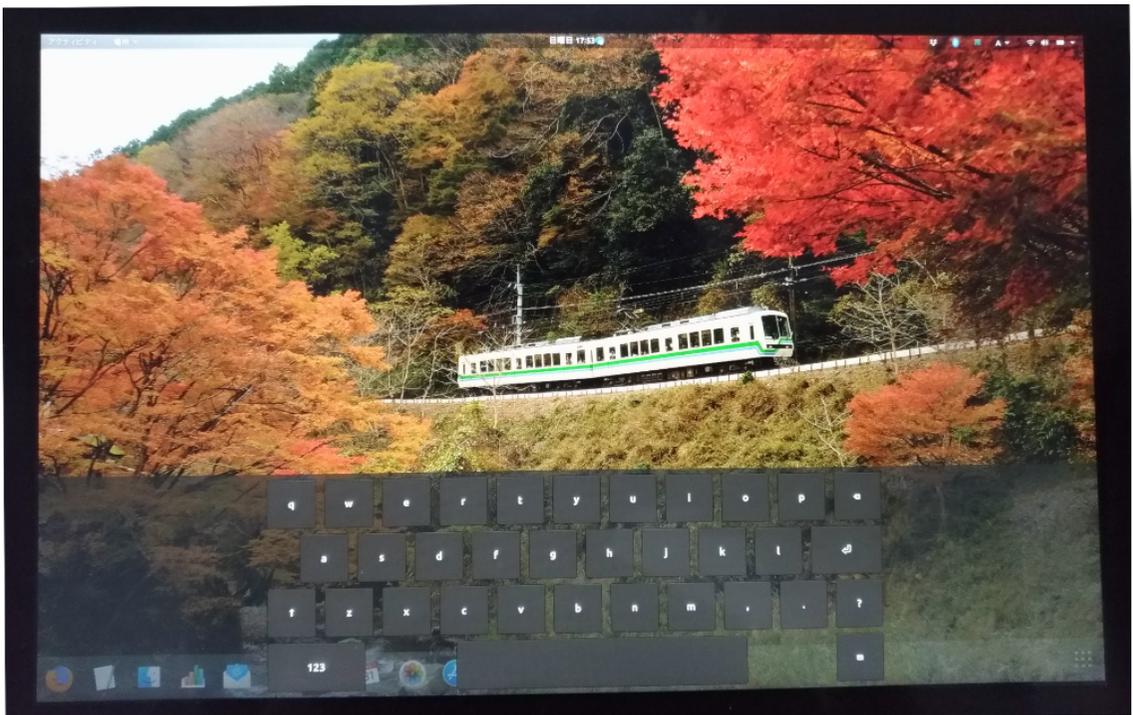


図: キーボードを出したときの様子。スクリーンショットが撮れなかったので写真を撮影することになった。

また、タブレットモードからラップトップモードに戻す際にサスペンド状態になっていましたが、

¹ <http://zapppaaan.freepub.jp/article/156663726.html>

Ubuntu 17.10 ではそのような現象は見られませんでした。このため、タブレットモードとしてはざっば〜ん♪ vol.3 の当時よりはかなり使いやすくなっています。2in1 のノートパソコンの種類が増えているので、他のノートパソコンに Ubuntu をインストールさせる後押しになるのではないのでしょうか。

6.2 ノートパソコンを買い替えた

長らく使用していた Let's note ですが、バッテリーがある頃から急激に減少して電池切れになってしまふことがありました。加えて電源レバーをスライドさせてもなかなか起動しないことが度々あったため、HP の [Elitebook Folio G1](#)² に買い換えました³。決め手はキータッチの良さと、1kg を切る重量、何より USB Type-C での給電になるため、筆者が持っている Zenfone 3 と同じ充電ケーブルを使用することができます。そのため、意外とかさばるノートパソコン専用 AC アダプタを外出先へ持っていく必要がなくなります。

そして、[Ubuntu Weekly Recipe 第 432 回](#)⁴にてロードテストが公開されていた機種なので、先人がいると購入のための心理的障壁は格段に下がります。なかなか高価な買い物なので、「いざ買ってみただけでも全くもって使えなかった」という場合の心理的喪失は計り知れません。とはいえ、この発想がチキンレースにつながりかねず、新しめのノートパソコンを購入するための判断材料が生まれにくいというジレンマも抱えています。

6.3 右クリックができない

Ubuntu 18.04 LTS をインストールして軽く操作してみたところ、タッチパッドの右下方の押下で右クリックができないことが判明しました。タッチパッドの設定確認も行ってみましたが、主ボタンしか認識しないようです。なお、二本指でタップすると、右クリックと同じ動作になります。

また、タッチパッドのボタン押下時間の長さで主ボタンと副ボタンの機能切り替えのタイミングを調節する場合、[設定] から [ユニバーサルアクセス] → [クリック支援] をクリックします。すると、メニューが表示されるので、[副ボタンのクリックの代替] をオンにします。どれだけ長く押すことにより副ボタンのクリックの代替をさせるかは、認識するまでの間隔を調節するレバーで調節します。個人的な感覚ですが、一番短くしても「普通に右クリックしたつもりだったのに」ということにはならないと感じました。



図: 副ボタンのクリック代替のメニュー

² http://jp.ext.hp.com/m/notebooks/business/elitebook_folio_g1/#contents_model

³ 入れ替えられた Let's note は、致命的な不具合はないため、Windows に入れ替えて引き続き頼もしく稼働していただく予定です。

⁴ <https://gihyo.jp/admin/serial/01/ubuntu-recipe/0432>

6.4 Type-C から映像端子への変換

Ubuntu Weekly Recipe では、USB Type-C 端子では USB メモリの認識のみで、映像系の端子は認識できなかったとありましたが、18.04 ではどうなっているのでしょうか。今回 Type-C から様々な機器を接続するためのハブとして、HP 純正のハブではなく、RayCue の USB ハブ⁵を選びました。これにした理由は、映像系の端子を持つ Type-C の多機能ハブとしては 5,000 円を下回るお手頃な価格であったからです。Elitebook で試してみたところ、HDMI 端子、VGA 端子いずれを使用してもディスプレイに表示させることが可能でした。この 2 つの端子で接続可能であれば、外出先で映像を投影する際にも、ほとんどの場合困ることはないでしょう。

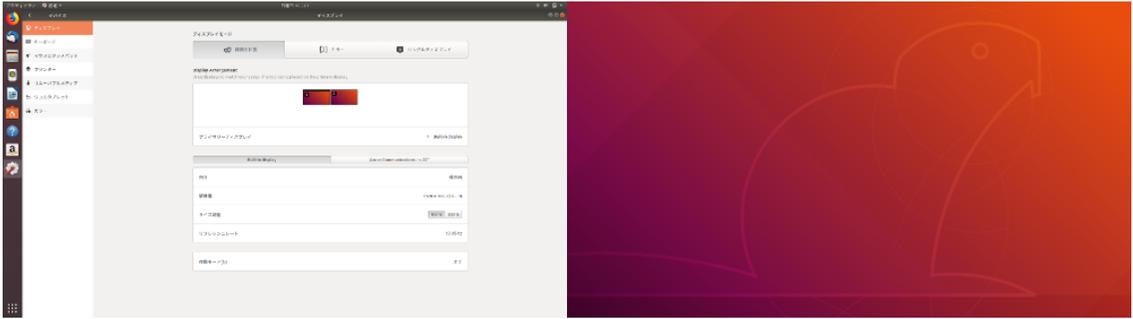


図: 2 画面表示が成功した

ケーブルが薄型で、折りたたんで本体下部に収納し、使わない時はコンパクトにできることもメリットですが、欠点としては、特に Elitebook で用いる場合に、USB Type-C の端子が右側にしかないため、取り回しが難しいところです。このように映像系の端子が後ろに、充電用 USB Type-C 端子が前に来ているので、手元の配線が散らかってしまうこともあるでしょう。



図: RayCue のハブを接続した様子。使用者側から見るとロゴが反転する。

ちなみに、たまたま原稿執筆中に入手した頂きものの ELECOM の USB ハブ⁶でも試してみましたが、RayCue のハブと同様に HDMI 端子が認識します。こちらはノートパソコン側のケーブルの長さがある程度あるので、パソコン周りをすっきりさせつつ各端末を接続することができます。

⁵ <https://www.amazon.co.jp/dp/B074FWNSW8>

⁶ <http://www2.elecom.co.jp/products/U3HC-DC03BBK.html>

第7章 Ubuntu で心理学実験

はにゅう

ざっぱ～んに寄稿するのは初めてです。はじめまして。はにゅうと言います。関西某所で院生をしています。研究するにあたり、Ubuntu を使って実験環境を整えているので、今回は Ubuntu を使ってつくる実験環境のお話をしようと思います。初心者の方への助けになれば幸いです。

7.1 はじめに

わたしの専攻は（主に）認知心理学です。認知心理学の研究では、主に PC を用いた心理学実験を行います。心理学実験では、例えば、PC モニタにある文字が^{ていじ}呈示されたら、できるだけ速く・正確に何かのボタンを押す、といったようなことを行います。わたしは、そうした行動の結果得られた、ボタンを押すまでにかかった時間（反応時間）・間違えた回答の回数・ある一方の反応に偏る確率などの数値データを見ることで、人間にどのようなしくみが備わっているのかを明らかにするのが、「心理学実験」だと考えています。

現代において、心理学実験の制御には PC が必須です。なぜなら、PC を用いれば、文字などの刺激をいつ、どのような順番で、どれぐらいの輝度で呈示するのか、などの細かい条件指定に答えられ、0.0 何秒といった細かい時間をデータとして得ることができるからです。そういった実験の制御を行うには、プログラムを組み立てなければいけません。

そこで役立つのが「PsychoPy」です。PsychoPy の導入方法に入る前に、PsychoPy とはそもそも何なのか、なぜ Ubuntu を用いるのかをお話しようと思います。

7.2 PsychoPy とは

PsychoPy は、公式ホームページ¹では以下のように紹介されています（以下引用）。

PsychoPy is an open-source application allowing you run a wide range of neuroscience, psychology and psychophysics experiments.

It's a free, powerful alternative to Presentation(tm) or e-Prime(tm), written in Python(a free alternative to Matlab(tm)).

補足も入れて簡単に要約すると、

PsychoPy とは、Python という言語で動く、オープンソースのアプリケーションです。

PsychoPy では、心理学実験、心理物理実験、神経科学に関する実験など、幅広い実験プログラムを簡単に組み立てることができます。

元々実験系心理学、神経科学の研究者の人たちは、主に Matlab という有償ソフトウェアを使っていました。その Matlab に代わるものとして、無償の PsychoPy が誕生しました。

というところでしょうか。

この説明の通り、PsychoPy は初心者の人でも簡単に実験を組み立てられる、優れものなのです。

ところで、PsychoPy は Python で動くのですが、Python は数値計算だけでなく、大切な文章を書くのにも向いていますね。

なお、今回導入する PsychoPy は、ソフトウェアの方ではなく、実験・分析に必要なツールを集めたパッケージの方です。

¹ <http://www.psychopy.org/>

7.3 なぜ Ubuntu なのか

子供の頃、自分よりずっと年上の、親やお姉さんお兄さん、アニメやゲームの登場人物が使う、靴・服・扱う武器に、憧れたことはありますか。わたしは憧れました。なんだったら今でも……。とーはんどがんず……。

真面目な理由としては、2つあります。

1つは、Ubuntu は重くなりにくく、安定しているためです。人間のふるまいの指標を反応時間とするような実験や、刺激の実際の呈示時間を制御することが必要な実験では、できるだけ精密なデータを採ることが重要になります。ところが、データを採るとき、OS の挙動によって邪魔をされてしまったり、ノイズが入ってしまったりすることがあります。Ubuntu はその点、そういった邪魔が入ることが少なく安定しているため、心理学実験を行うのに適している環境であると考えています。

もう1つは、Matlab に依存しないためです。もちろん Matlab オンリーでいくのを否定しているわけではありません。Ubuntu で手軽に実験環境を整えることができれば、いつでもどこでも、自分の思うがままに実験を行うことができます。これは、Matlab に依存してはできないことだと考えています。

7.4 実験環境の構築

それでは早速実験環境の構築を行きましょう。

今回用いたのは、Dell precision Tower 3420 で、Ubuntu のバージョンは 16.04 LTS でした。PsychoPy インストールの際に導入するものは以下の通りです。

- pyenv: 複数の Python のバージョンを管理することができるツール。
- anaconda: データ処理などに使われる Python のモジュールをまとめたパッケージ。簡易版もあります (miniconda)。

7.5 インストールの手順

7.5.1 pyenv をインストールする

Ubuntu はシステムの至るところで Python が使われているようで、素直にインストールすると、こながらがってえらいことになってしまいます。わたしはそのせいで GNOME-terminal が死にました。めちやくちや焦りました。

というわけで (?), 今回は pyenv の上に anaconda を導入します。

```
$ git clone https://github.com/yyuu/pyenv.git ~/.pyenv
$ echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bashrc
$ echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bashrc
$ echo 'eval "$(pyenv init -)"' >> ~/.bashrc
$ source ~/.bashrc
```

なお、anaconda と PsychoPy に必要な Python のバージョンは 2.7.14 です。何故 2 系なのかは以下で述べています。Ubuntu 16.04 LTS の Python のバージョンは、2.7 と 3.5 の 2 つがあるようです。

7.5.2 pyenv 上で anaconda をインストールする

ここで注意しなければいけないのは、導入する anaconda のバージョンです。導入する PsychoPy の関係で、3 系の anaconda を導入しても 2 系の anaconda に降ろされるので、2 系を導入します。2 系の最新バージョンは一行目の結果で確認します。3 系を使いたい場合は、仮想環境で走らせることができます。

```
$ pyenv install -l | grep ana
$ pyenv install anaconda2-xxx
$ pyenv rehash
$ pyenv global anaconda2-xxx
$ echo 'export PATH="$PYENV_ROOT/versions/anaconda2-xxx/bin/:$PATH"' >> ~/.bashrc
$ source ~/.bashrc
$ conda update conda
```

また、仮想環境に入る際に用いる「activate」が pyenv と anaconda で衝突する場合がありますので、念の為、衝突しないように path に書いておきます。

7.5.3 PsychoPy を導入する

PsychoPy、何バージョンかありますが、cogsci psychopy を導入します。eric psychopy を入れてみたりしましたが、わたしの環境でうまく行ったのは cogsci psychopy のみでした。

```
$ conda install -c cogsci psychopy
```

以上で終了です！ お疲れ様でした。

7.6 もしかしたら

PsychoPy の導入の際、入れていないはずの PsychoPy と、今から入れようとしている PsychoPy が衝突して入れられないよ！ という状況に何度も遭遇しました。

はっきりとした原因の解明までは力がなくて行きつけなかったのですが、かつて（かつて）PsychoPy を「sudo apt-get install」で入れようとした（実際には導入をやめた）ことに起因していたようです。

この手順でも導入できない場合は、「conda list」や「dpkg -l」、「vim ~/.bashrc」等のコマンドで、PsychoPy がどのような状況になっているかを確認し、パスがどこにいつているかを見てみてください。

7.7 デモコード

Hello world が順番で出てくるだけのデモコードです。次節以降で紹介する開発環境を使って、走らせてみてください。PsychoPy を使って、こんな風にして実験プログラムを書いていきます。

```
from psychopy import visual, core
win = visual.Window([400, 400])
me = visual.TextStim(win, text='hello')
me.setAutoDraw(True)
win.flip()
core.wait(1)
me.setText('world')
win.flip()
core.wait(1)
win.close()
core.quit()
exit()
```

それでは、楽しい心理学実験ライフを！

第 8 章 ゲストページ

8.1 国際イベントの招致を手伝ってみましたよ

おがさわらなるひこ

Ubuntu ユーザーな私が、なぜだか 2017 年 10 月 21 日、22 日に開催された「openSUSE.Asia Summit」というイベントの東京招致をお手伝いすることになったりしたので、そのメモ書きなどをここに残しておこうと思います。別のディストリビューションのイベントですけど固有の話はそんなにないですし、FLOSS な国際イベントをホストするって体験は皆様にも面白いかなど。あくまでも個人的な体験をダラダラ書いているだけなので、今すぐ役立つノウハウ！ みたいなのは期待しないでお読みくださいませ。

8.1.1 openSUSE.Asia Summit とは

openSUSE¹について私は説明できるほど詳しくないですが、なんとなくヨーロッパ色が強いディストリビューションな気がしています。だからでしょうか。年に 1 度の openSUSE のカンファレンス openSUSE Conference (oSC) はヨーロッパ (チェコのプラハだったかな?) で開催されています。

しかし、中国 (少なくとも北京) や台湾には SUSE のオフィスがある関係でユーザーもおり、日本にもアクティブに活動しているメンバーがいます。そういう人たちが相互に会って話せる場がヨーロッパにしかないってのはなかなかつらいよね²、だから Asia カンファレンスをやろうよ! というモチベーションで開催されるようになったのが openSUSE.Asia というイベント……なんだそうです。

イベントそのものについての情報は公式ページ³もありますし、openSUSE な人によるレポート⁴や彼らによる同人誌⁵に招致レポートは載ってますし、まあこれぐらいにしておきますね。

8.1.2 私と openSUSE.Asia

ちょっと自分語りですけど。

FLOSS なデスクトップ環境である GNOME には GNOME.Asia⁶ というイベントがありまして、これもきっと似たような動機で開催されてるイベントだと思うんですけど。なんとなくネット見てたらこのイベント見つけて、そのときはたまたま 2014 年で北京開催だったんですけど、北京だったら近いじゃんって思って遊びに行ったんですね⁷。

そしたら、そこで SUSE 北京オフィスの人たちがブースを出していて、「日本から来たの? 日本にも openSUSE でアクティブに活動してる人いるでしょう、知ってる?」「うん、もちろん知ってるけど」「今年、Asia カンファレンスをやろうって思ってるの、北京で。だから一緒に手伝ってくれない? くれるよね?」「え、うん、まあ、それはできることなら手伝いますよ」みたいになって。

手伝うといってもときどき口を出す (そして当日参加する) レベルで、本当に頑張ってたのは日本 openSUSE ユーザ会の武山さんなんですけど、まあ、なんにせよ、始めることは大事だし、初回の openSUSE.Asia Summit 2014 に立ち会えたのは誇らしくも嬉しかったです。

¹ <https://ja.opensuse.org/>

² 私が主に活動している LibreOffice についても年次カンファレンスは基本的にヨーロッパで行われており、こちらへの事情には親近感があります。

³ openSUSE.Asia Summit 2017 Tokyo <https://events.opensuse.org/conference/summitasia17>

⁴ openSUSE.Asia Summit 2017 イベントレポート by 橋本修太 <http://blog.geeko.jp/syuta-hashimoto/1671> なお Geeko Blog には 2016 年以前のレポートも掲載されています。

⁵ Geeko Magazine Special Edition 2017 冬 <http://blog.geeko.jp/ftake/1692>

⁶ 今年のページ <https://2018.gnome.asia/> 今年は台湾で開催。台湾最大の FLOSS イベント COSCUP と、さらに openSUSE.Asia と併催です。

⁷ このとき RMS が基調講演に来てて、FSF のステッカー貰いに行ったんだけど緊張のあまり話しかけられなかった……という話は、まあおいといて。

引き続き 2015 年の台湾では、武山さんの頑張りもあって日本からの参加者も増えて、その翌年はインドネシアはジョグジャカルタで、これも日本から何人が参加して、私は行かなかったんだけどものすごい熱気だったらしく^{*8}、そこのクロージング LT で「来年は東京だ！」ってやったら大盛り上がりだった、って話を聞きまして。

「で、東京に招致しようって話があるんですけど、どう思います？」と聞かれたのが 2016 年の秋、インドネシアのイベントが終わってからしばらく経ってだと思います。

「うーん、GNOME.Asia のときも、次は日本でやろうぜってみんなに言われたし日本でやれば来たって人多いと思うんだけど、正直、傍から見てる今の日本の openSUSE コミュニティの体力でやるのは相当きついんじゃないかな？ただ、背伸びしてでもイベントやらないとコミュニティに体力がつかないって考えはあると思うし、やるというなら、できる範囲で手伝いますよ」と返事をしたような気がします。

8.1.3 助走期間

記録をたぐると 2016 年 10 月には、日本 openSUSE ユーザ会の slack に専用のチャンネルができて、11 月 2 日にキックオフ的な会合をやっていますね。

やるなら手伝いますよって人がまあまあ集まって、これまでの実績で各国から日本に来そうな人数と、日本の openSUSE およびその周辺コミュニティの人間を数えると、集客 100 人ぐらいはできるのかなあって感じになって、それならやりますかって温度感になっていったと思います。中国の重慶が GNOME.Asia の招致を決めて、それと併催って情報が入ってきたので、ただ提案出せば OK というわけでもないし、このイベントとコラボできないかとかあその場所使えないかとか思いつきを話したりしてた時期ですね。

2 月中旬ぐらいまでは割とチャットも活発だったけど、そこからしばらく停滞期があって、3 月中旬にグローバルのほうで CHF (Call for Host; 開催地募集) の日程が「4 月いっぱい」と決まって、また動き出したという感じ。

8.1.4 CFH Proposal 作成

で、CFH に出す Proposal (提案書) づくりが本格化。

openSUSE.Asia Summit の場合、Proposal に書く内容はおおむね次のとおりでした。

- ・なぜそこで (今回は東京) 開催するのか。その意義は
- ・場所
- ・日程
- ・大まかなプログラム (講演が何トラックあって、何人が発表できて、ワークショップの開催の可否、などなど)
- ・予算
 - 支出: なににいくらぐらい必要か
 - 収入: イベント自体や懇親会を有償化するならそれ、あと日本チームで集められそうなスポンサー
- ・現地への交通手段
- ・おすすめホテル (ざっくりとしたコスト)
- ・食事

決めごととして一番大きいのは、まずは場所と開催日程なんですね。これが決まらないと色々書きようがない。

- ・少なく見積もって 100 人が収容できて

^{*8} ジョグジャカルタは JICA の支援によって初等教育の E ラーニングシステムに openSUSE を入れているそうで、それもあってインドネシアは若い人たちに Linux、FLOSS に関心がある層が多いんだそうです。

- ・ オープニング・キーノート・クロージングでは全員入れる部屋があって
- ・ 数トラック並列で講演をできて
- ・ さらに並列でハンズオンが実施できて
- ・ お金があんまりかからなくて⁹
- ・ 4月の時点で10月の予約ができて
- ・ 万が一、CFHで他の候補地に負けたら無償で（or 安価で）キャンセルできること

これらの条件を満たす場所を探さなければいけない。

予算の都合で有償のイベントスペースはNG。予約の関係で公共系施設は難しい。となると学校系なんですが、学校系はけっこう制約がおおくて、正面からお願いすると断られることも多くてですね……。

どうも決め手にかけて困っていたら、IBMのBluemixな人脈から、電気通信大学を紹介してもらいました。内部のベンチャー企業¹⁰を通すので手続き的には問題ないと。で、確定したのが4月16日。今から思い返すとギリギリだな……。ということで、まずは場所抑えるってことを意識したほうがいいと思います。

あと大変というか面倒だったのは予算ですね。ただ、これについては私あまり関わってないので割愛。

Proposalは全体を通して、大雑把に担当を決めて、Google Docs上で直接編集して作成していきましました。みんながみんな英語が得意ではないので（私も）、書きたい思いがある人に日本語で書いてもらって、それを別の人が英語にしたり。オンラインだとなかなかうまく進まないこともあるので、作業日を用意して一気に進めることも大事ですね。

英語については、カジュアルな言い回しを好む人、かっちりした言い回しを好む人の違いなんかもありましたが、ここは最終的には、リーダーの武山さんの感覚に揃えることになりました。Google翻訳を日→英、英→日と二回かけて意味が通るかチェックしたり、文法チェック Grammary¹¹を利用したりも大事。

そんなわけで書き上げたCFH ProposalをPDF化して実行委員にメールで送付。つかれたー。

で、重慶と決選投票の結果、大量リードで東京採択。我々のライバルでなければ、重慶のプロポーザルも魅力的だったんですけどねえ。まーともかくめでたしめでたし。一息つく間もなく、実際の準備に追われることとなります。

……このペースで書いていると終わらないので、以下、私に関わった中で特筆すべきことに限ります。

8.1.5 インフラなど

主に日本ローカルチームで使ったインフラは、日常の会話：slack、作成した文書やファイルの置き場：Google Drive、タスク管理：Trelloです。外部からのメールの受け口はgoogle groupsを利用しました。基本的には独自のインフラを持たない方向でやってみました。

定例ミーティングはビデオチャットでやったのですが、20人を超えるメンバーをさばける、2時間程度の連続したミーティング可能、携帯からも利用可能、画面共有ができるということで、最終的にはGoogleさんの有償サービス、Hangout Meetに落ち着きました（代金は予算から充当）。当時はDiscordは画面共有がまいちのことだったので今はどうなんだろう。

グローバルも定例がslackでタスク管理はTrelloですね。後述のロゴコンテストのサイトは、北京コミュニティが作成したツールをインドネシアチームがホストして立ててくれました。

イベントのメインページや後述CFPは、openSUSEコミュニティが開発しているOSEM¹²というオープンソースを用いたサーバーがグローバルで立っておりそれを用いました。参加登録は、日本国内での認知度などを考えてconnpassを用いたのですが、OSEMにも参加登録機能があるので「なんで使わないの」ってBoardにチクリと言われたりしました。納得はしてもらいましたけど。

⁹ 言語系のカンファレンスのように数万円チケットを売っても即日完売するような、あるいはスポンサーをかき集められる見込みがあればいいんですけど、まあそこは守るしかなかったです。

¹⁰ <http://www.usa-mimi.jp/>

¹¹ <https://www.grammarly.com>

¹² <https://github.com/openSUSE/osem>

「うぶんちゅ! まがじんざっぱ〜ん♪」バックナンバー

vol.1 (2013年12月20日発行)

表紙 瀬尾浩史
プチ帰ってきた『行っとけ! Ubuntu 道場!』
特別編 hito
Ubuntu で作るおうち録画環境 kazken3
Unity から自由を奪還せよ 柴田充也
これで完璧!! Ubuntu で印刷 again! おがさわらなるひこ
Enju Leaf でつくるオレオレ蔵書管理サーバ 長南 浩
Ubuntu マシンで艦これを動かして遠隔プレイできる環境を作ってみた 鶴ノ子餅すあま
SD 連載 Ubuntu Monthly Report における動物の変遷 SoftwareDesign 編集部 金田富士男
This is me, with Ubuntu. おしえたかし
うぶんちゅ まがじんざっぱ〜ん Vol.01 発売おめでとう 水野源
Ubuntu と私 あわしろいくや
終わりに あわしろいくや
価格: 500 円
販売サイト・体験版:
<http://zapppaaan.freepub.jp/article/82821893.html>

vol.3 (2015年7月25日発行)

表紙 写真: 水野源
OpenNebula で PCI passthrough おおたあきひこ
Let' s note CF-RZ4 に Ubuntu をインストールしてみる話 Rakugou
21 世紀の Device Tree 柴田充也
ちょーなんさんちのノート PC 事情 長南 浩
かよちんとボクと、時々、録画 kazken3
磁気センサーを使った冷蔵庫監視システムの構築 水野源
新し目の Mozc をビルドする あわしろいくや
著者紹介
編集後記 あわしろいくや
価格: 700 円
販売サイト・体験版:
<http://zapppaaan.freepub.jp/article/156663726.html>

vol.2 (2014年11月15日発行)

表紙 Ubuntu 4.10
プチ帰ってきた『行っとけ! Ubuntu 道場!』
特別編 第二回 hito
あのプロダクトは今!? 柴田充也
Ubuntu を「未来をうかがう」道具にする 長南 浩
普通の社会人が【録画環境】を(もう少し)やってみた kazken3
Ubuntu 10 歳、CUPS 15 歳 おがさわらなるひこ
Ubuntu 14.04/14.10 でも ATOK X 3 を動かす あわしろいくや
Ubuntu Studio フレーバー 7 周年 坂本 貴史
Ubuntu で Blink(1) mk2 を動かしてみる kazken3
Ubuntu8.04 はこんなだった あわしろいくや
人と Ubuntu と私 芝田 静間
第二弾おめでとうございます Ueno
私が愛してきた OS 達 長南 浩
普通の大学教員が【Ubuntu】やってみた。おしえたかし
終わりに あわしろいくや
価格: 700 円
販売サイト・体験版:
<http://zapppaaan.freepub.jp/article/105631657.html>

vol.4 (2016年1月30日発行)

表紙 イラスト: よかぜ
艦これで学ぶ通信傍受入門 柴田充也
お得でおいしい SSL 証明書のとりかた 長南 浩
Xymon Maniax Hajime MIZUNO
Ubuntu と知の巨象、Evernote との共存をめぐる Rakugou
Ubuntu GNOME の再インストールと棚卸し あわしろいくや
Blu-ray を Ubuntu で観よう kazken3
海外カンファレンスの楽しみ おがさわらなるひこ
著者紹介
記録に残るものが記憶に残るもの〜あとがきに代えて〜 あわしろいくや
価格 (Gumroad): 500 円
価格 (DLsite): 756 円
販売サイト・体験版:
<http://zapppaaan.freepub.jp/article/173095293.html>

vol.5 (2016年8月21日発行)

表紙 イラスト：よかぜ
続 OpenNebula で PCIpassthrough おお
たあきひこ
snap パッケージを作ってみよう kazken3
HummingBoard/PT3 でつくる小型録画ペ
アポーン ryunuda
Cinnamon のおかしな翻訳にツッコミを入
れる あわしろいくや
Ubuntu 16.04 と Windows 10 でデュアル
ブート Rakugou
LibreOffice Online で遊んでみよう おがさ
わらなるひこ
あとがき あわしろいくや
著者紹介
価格 (Gumroad・BOOTH) : 700 円
価格 (DLsite) : 972 円
販売サイト・体験版 :
[http://zappaaan.freepub.jp/article/
176563093.html](http://zappaaan.freepub.jp/article/176563093.html)

vol.6 (2017年4月9日発行)

表紙 イラスト：よかぜ
OpenNebula の Private
MarketPlaceApp おおたあきひこ
いかにして翻訳(ほんやく)をするか 柴田
充也
Ubuntu 音楽再生アプリ探訪 長南 浩
2017年のUbuntu 録画環境 kazken3
Cyclograph で GPS を使わず自転車ライフ
ログ Rakugou
Docker のアプリコンテナとして Re:VIEW
を動かそう 柴田充也
Ubuntu で血迷ってアダルトサイトを作っ
てみた話 長南 浩
特別コラム あわしろいくや
著者紹介
価格 (Gumroad・BOOTH) : 700 円
価格 (DLsite) : 972 円
販売サイト・体験版 :
[http://zappaaan.freepub.jp/article/
179274142.html](http://zappaaan.freepub.jp/article/179274142.html)

総集編 1 (2017年4月9日発行)

表紙 イラスト：よかぜ
発刊に寄せて あわしろいくや
Ubuntu を「未来をうかがう」道具にする
長南 浩
21 世紀の Device Tree 柴田充也
Xymon Maniax Hajime MIZUNO
snap パッケージを作ってみよう kazken3
HummingBoard/PT3 でつくる小型録画ペ
アポーン ryunuda
LibreOffice Online で遊んでみよう おがさ
わらなるひこ
関係者より一言
価格 (Gumroad・BOOTH) : 700 円
価格 (DLsite) : 972 円
販売サイト・体験版 :
[http://zappaaan.freepub.jp/article/
179274130.html](http://zappaaan.freepub.jp/article/179274130.html)

vol.7 (2018年2月11日発行)

表紙 イラスト：よかぜ
OpenNebula で Nextcloud サーバー構築
おおたあきひこ
サーバーレスでオレオレ画像アップロー
ダーを作る 水野源
BuildStream で GNOME アプリをビルド
柴田充也
LibreOffice 6.0 Writer で縦書き小冊子を
作成する あわしろいくや
Ubuntu を macOS High Sierra 風に変え
てみる。Rakugou
ざっぱ〜んを支える技術:ダウンロード編 柴
田充也
Flatpak で Ubuntu 16.04 LTS 時代の
gedit をよみがえらせよう あわしろいくや
あとがき あわしろいくや
著者紹介
価格 (Gumroad・BOOTH) : 700 円
価格 (DLsite) : 756 円
販売サイト・体験版 :
[http://zappaaan.freepub.jp/article/
182391181.html](http://zappaaan.freepub.jp/article/182391181.html)

うぶんちゅ！ まがじん ざっぱ〜ん♪ vol.8 【体験版】

2018年6月24日 v1.0 発行

著者 team zpn
発行所 team zpn

(C) 2018 team zpn

- Ubuntuではじめる楽しいゼミ運営
- ポメラDM200にUbuntuをインストールする
- Boomagaを使ってPDFを小冊子印刷する方法
- Nano Pi NEOで作成するテレビ視聴環境
- いつでも始められるmpv
- らくごうさんちのノートPC事情
- Ubuntuで心理学実験
- 国際イベントの招致を手伝ってみましたよ
- 技術書典4で冊子版
『ざっくりわかるUbuntu 18.04 LTS』
を頒布できなかった顛末