

The book of Arch

あーちのほん



RabbitHouse
ラビットハウス・ジャパン

Nomuken氏 著

1 はじめに

じえんつーの本があるなら、あーちの本があってもいいじゃないということで書くことにしました。確実に他の人が既書いている気はするのですが、近年の ArchLinux への感謝の気持ちを込め、いかに使いやすいのかを伝えるために書きたいと思います。

今回の内容は、簡単にひとまずインストール編です。次回があるのかは完全にわからないですが、デスクトップ環境に lightdm + cinnamon + mikutter 環境を構築することを目標に、最速で（省略できるところは省略して）インストールを目指します。

おそらくこの本を終えると、「極めて簡単 + 高速」に自分が望む環境を構築出来ることが理解していただけるものと思います。そして、これを読んで Linux が必要になった時、ArchLinux も選択肢に入れて貰えればなぁと思います。

最後に、日々大学関係でお世話になっていて、この本の内容をギリギリまで書かずに多大なる迷惑をかけた@wakamesoba98 氏に多大なる感謝を。あと、辛い毎日を支えてくれている、ご注文はうさぎですか??に感謝を。

なお、表紙は ArchLinux のロゴと、フォントに Koruri を利用させていただきました。ちなみに、Koruri は ArchLinux だと極めて簡単にインストールでき利用できる、複数のフォントを組み合わせた美しい日本語のフォントです。^{*1}

- "Archlogo" by Arch Linux Developpers - <http://archlinux.org/art/>
- "Koruri" by Hotaka Hitagi (lindwurm) - <http://koruri.lindwurm.biz/>

また、記事内では ArchLinux の TM 表記を省略しています。

なお、この文章は https://spica.bz/arch-b00k/palloc_pro/document.pdf でダウンロードすることが出来ます。PC からの閲覧に利用していただければと思います。また、2016/2/1 あたりに無償公開されます、こちらも合わせてご了承ください。

^{*1} AUR ttf-koruri

2 何故 ArchLinux ?

これに関しては、多分自分が書くよりも公式のドキュメントを読むのが一番でしょう。

- ArchWiki(The Arch Way)
https://wiki.archlinuxjp.org/index.php/The_Arch_Way
- ArchWiki(Arch と他のディストリビューションの比較)
https://wiki.archlinuxjp.org/index.php/Arch_%E3%81%A8%E4%BB%96%E3%81%AE%E3%83%87%E3%82%A3%E3%82%B9%E3%83%88%E3%83%AA%E3%83%93%E3%83%A5%E3%83%BC%E3%82%B7%E3%83%A7%E3%83%B3%E3%81%AE%E6%AF%94%E8%BC%83

軽くまとめてお伝えするのであれば、ArchLinux はユーザに常に選択を求めるディストリビューションです。特に、ArchLinux ではユーザが指定しなければ基本的に自分で勝手にセットアップして起動するようなことはしません。そして、設定が行いやすいように ArchLinux では可能な限りバニラ (ArchLinux 側でソースコードを改変していない) な状態でパッケージが提供されます。^{*2} よって、ArchLinux を利用すれば (わかっているのであれば) 高速、簡単に環境を構築します。

え？設定ファイルを書くのが大変ですか？安心してください、ArchLinux は Linux 界の家庭の医学とも言える恐ろしい量のパッケージセットアップの方法が記述されています。とても、使いやすく読みやすいのでぜひインストールにはご一読ください。^{*3}

ArchWiki を見ても分からないことがありましたか？ご安心ください、日本ではどうも人気が無いですが ArchLinux の forum^{*4} を利用すればプロの人が教えてくれます。もし、あなたが英語が達者であれば活発な本流の forum を使っても良いと思いますが、日本の forum でも応答してもらえますと思います。決して恥ずかしいと思わず、あなたが調べてどうしても分からなければ利用しても良いと思います。

Let's ArchLinux!

^{*2} ただし、Linux Kernel は 2015/12/26 現在 patch が一つあたっています。

^{*3} とはいえ翻訳が間に合っていないところも多く、英語版も合わせて読むことをおすすめします。例えば LXC の記事は英語のほうが参考になります。

^{*4} Arch Linux フォーラム - <https://bbs.archlinuxjp.org/>

3 高速にインストールしよう

というわけで、ArchLinux を手短かにインストールしましょう。おそらく、一番楽な ArchLinux の利用方法は Vagrant を利用し ArchLinux の box を入手して利用することですが、手でインストールする方法をご紹介します。この章を読むためには、ブラウザ等で ArchLinux のビギナーズガイドを参照しながら読むことをおすすめします。

「ビギナーズガイド」

<https://wiki.archlinux.jp.org/index.php/%E3%83%93%E3%82%AE%E3%83%8A%E3%83%BC%E3%82%BA%E3%82%AC%E3%82%A4%E3%83%89>

さて、準備はできましたか？インストールしていきましょう。

3.1 インストールの流れ

インストールでは以下の条件でお話を進めます。

- GPT + BIOS 環境
- 一番シンプルな環境
- lightdm + Cinnamon + mikutter 環境がゴール
- といっても、google-chrome も使いたいよね

筆者の都合上、インストールは VM 環境で行いながらこれを書いておりますが、多くの場合自分のコンピュータに入っている Windows を抹殺して ArchLinux に入れ替えたいと思っている人も居ると思います、その場合は多分 UEFI 環境になります。冗談抜きで言えば、ArchLinux と Windows のデュアルブートをしたい人が主かもしれません。……ここまで書いておきながら、申し訳ないのですがその人たち全員を網羅することはできません。(この本の締め切りを 3 回も伸ばすくらいに時間がない*⁵)

とはいえ、そのインストール方法のヒントくらいは脚注とかそんな感じで残したいと思います。

そんなわけで、インストールは以下の流れで行います。

1. iso イメージで起動する
2. 作業……その前に
3. パーティションを分ける
4. pacman の設定
5. 必要なパッケージを選ぶ

*⁵ @wakamesoba98 氏、すみません

6. 必要最低限の設定をする

- (a) 新環境に入る
- (b) 時刻の設定
- (c) 言語の設定
- (d) ホスト名とネットワークの設定
- (e) ユーザの設定
- (f) ブートローダの設定
- (g) 各種サービスの有効化

7. やったぜ

それでは、進めていきましょう。

3.2 ArchLinux の iso イメージで起動する

iso イメージを USB メモリに焼く方法に関しては、詳しく述べませんが以下のようなツールを使うと良いかもです。

- (Windows 向け) dd for Windows
<http://www.si-linux.co.jp/techinfo/index.php?DD%20for%20Windows>
- (Mac,Linux 向け) dd コマンド

そして、iso イメージから起動します。

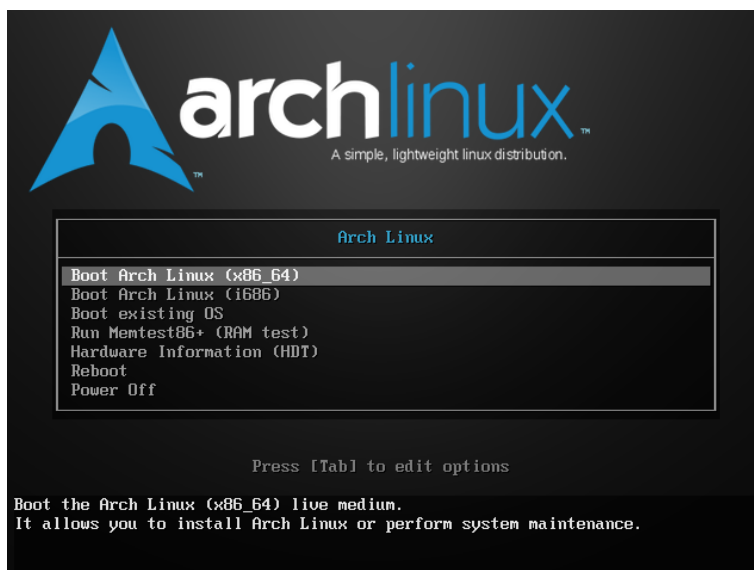


図 1 起動画面

この時の画面を忘れないでください！もし、画像のように表示されたらあなたは「BIOS 環境」として ArchLinux の iso が認識しています。あなたがコンピュータのプロで UEFI しか

許さないというなら、もう一度 UEFI の設定を見なおしてください。

もし、ArchLinux のロゴが表示されておらず黒い画面の中に ArchLinux のブートを選ぶ画面が出ているならあなたは UEFI 環境でブートしています。

3.3 作業.....その前に

おっと、作業を始める前にもしあなたが jp 配列のキーボードを利用しているなら以下の画面を入力してください。

```
# loadkeys jp106
```

3.4 パーティションを分ける

パーティションを分けます。玄人の方はパーティションの構造だけで多くの時間を消費するという噂も有りますが、シンプルに考えましょう。今回の要件を満たすにはパーティションは2つだけで問題ありません。以下がその設定です。

- BIOS boot Partition(100M)
- その他

これだけで、問題ありません。もし、あなたがより良いパーティション構成が浮かぶのであればそちらを利用して問題ありません。その際には、パーティションを分け、マウントするときにマウントポイントを含わせてマウントしてください。^{*6}

^{*6} ビギナーズガイド 7.8. パーティションのマウント - <https://wiki.archlinuxjp.org/index.php/%E3%83%93%E3%82%AE%E3%83%8A%E3%83%BC%E3%82%BA%E3%82%AC%E3%82%A4%E3%83%89#.E3.83.91.E3.83.BC.E3.83.86.E3.82.A3.E3.82.B7.E3.83.A7.E3.83.B3.E3.81.AE.E3.83.9E.E3.82.A6.E3.83.B3.E3.83.88>

あなたが Windows とデュアルブートを検討している場合はここで説明される手順を行うと、Windows 全てを消し飛ばしてしまいます。詳しくは以下の文章を参照してください。

ArchWiki 「Windows と Arch のデュアルブート」

https://wiki.archlinuxjp.org/index.php/Windows_%E3%81%A8_Arch_%E3%81%AE%E3%83%87%E3%83%A5%E3%82%A2%E3%83%AB%E3%83%96%E3%83%BC%E3%83%88

ArchLinux と Windows を共存させるには、まず先に Windows をインストールしてください。その後ディスクの管理から、Windows 領域を縮小させてから ArchLinux の ISO イメージをブートしインストールすると、上手く行きます。ただ、その際にどこに Windows 領域があるのかを忘れないようにしてください。UEFI 環境下で Windows と ArchLinux のデュアルブートをしようとしているなら、おそらくパーティションの 2 番目は EFI Boot Partition になっているはずです。このパーティションを、後で /boot にマウントしてください。

さて、パーティションを分ける前に自分のディスクがどこにマウントされているか確認しましょう。大体は/dev/sda にマウントされているはずですが、時折違うところにマウントされてしまうので注意してください。概ね、自分が使っている SSD もしくは HDD の容量を参考に見つけると良いでしょう。

```
# lsblk
NAME      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda         8:0    0 465.8G  0 disk
  sda1      8:1    0   450M  0 part
  sda2      8:2    0    99M  0 part
  sda3      8:3    0    16M  0 part
  sda4      8:4    0   59.5G  0 part
....skip
```

/dev/sda のところに注目してください。SIZE の部分がおよそ 500GB 付近になっているのが解ると思います。このようにして、見分けることが出来ます。なお、本記事ではコピーによるデータ消失の悲劇を防ぐために、インストール先のディスクを/dev/sdX と表記します。

さて、実際にパーティションを分けていきましょう。ここでは、パーティションを分けるために gdisk というコマンドを利用します。以下の手順に従ってください。


```

\\ Partition table を初期化する
Command (? for help): o \\GPT で作り直し
Proceed? (Y/N): y

\\ BIOS boot Partition を作成する
Command (? for help): n \\パーティションを作成
Partition number (1-128, default 1): \\ default で良いので Enter
First sector (34-62914526, default = 2048) or {+-}size{KMGTP}:
\\ 100M 分の領域を確保する
Last sector (2048-62914526, default = 62914526) or {+-}size{KMGTP}:+100M
\\ BIOS boot Partition を選択
Hex code or GUID (L to show codes, Enter = 8300): ef02

\\ その他のデータを格納するパーティションを作成
\\ 基本的に既定値を利用する
Command (? for help): n
Partition number (2-128, default 2):
First sector (34-62914526, default = 206848) or {+-}size{KMGTP}:
Last sector (206848-62914526, default = 62914526) or {+-}size{KMGTP}:
Hex code or GUID (L to show codes, Enter = 8300):

\\ 確認
Command (? for help): p

```

Number	Start (sector)	End (sector)	Size	Code	Name
1	2048	206847	100.0 MiB	EF02	BIOS boot partition
2	206848	62914526	29.9 GiB	8300	Linux filesystem

```

\\ Size は関係なしに, Name が同じ感じになっていれば OK

\\書き込み
Command (? for help): w
Do you want to proceed? (Y/N): y

```

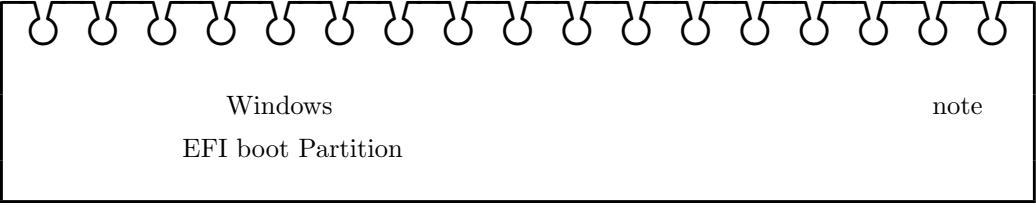
パーティションの操作は、難しいことが多くここで躓いてしまう人も多いかもしれません。でも、冷静に考えていけば理解は難しくないはずで、特に「表示されるメッセージをしっかりと読む」ということを心がけるようにしましょう。多くのトラブルはエラー文を読めば、原因もきっと分かるはずで、原因が分かれば Google で検索する際のキーワードも見つかりやすくなるはずで、

さて、次に確保した領域をフォーマットしましょう。ファイルシステムに関しても `btrfs` が良いなどのこだわりがある人も居たりするようらしいが、シンプルに `ext4` でフォーマットしよう。先ほどの、その他の領域として構築した場所に `Linux` をインストールするので先ほどの `gdisk` で確認した番号で言うところの 2 番目をフォーマットします。^{*7}

```
# mkfs.ext4 /dev/sdX2
```

で、マウントします

```
# mount /dev/sdX /mnt
```



もし、あなたが `Windows` とのデュアルブートを考えているのであれば先ほどの `note` で紹介した `EFI boot Partition` をマウントすることを忘れないください。

^{*7} 忘れてしまいましたか？それならもう一度 `lsblk` コマンドで確認してみてください、もしかしたら、`lsblk -f` としたほうが分かりやすいかもしれません

3.5 pacman の設定

mirror の選択

この本を読んでいる人の多くは日本在住の方が多いと推定するので、ここでは日本のミラーを選択します。もし他の Linux を使っていた人からすると fastmirror のようなものが無いのかと思う人もいるかもしれませんが、一応、ArchLinux には似たようなものとして、rankmirrors というコマンドがあります。これを利用すれば最速のミラーを選び mirrorlist を作成できます。

「ミラー#速度で並び替える」

<https://wiki.archlinuxjp.org/index.php/%E3%83%9F%E3%83%A9%E3%83%BC#.E9.80.9F.E5.BA.A6.E3.81.A7.E4.B8.A6.E3.81.B3.E6.9B.BF.E3.81.88.E3.82.8B>

なお、今後エディタは vim を用いて設定ファイルを編集を行っていますが、もしあなたが emacs であったり nano であったり他のエディタを使い慣れているならそちらに置き換えて利用してください。なお、もしあなたが CUI で操作するエディタに慣れていないなら、nano というエディタを利用することをおすすめします。そうするのであれば vim というコマンドを nano に置き換えて進めてください。^{*8}

```
# vim /etc/pacman.d/mirrorlist
\\こんな感じに編集（一番上に tsukuba のミラーサーバをコピー&ペーストする）
##
## Arch Linux repository mirrorlist
## Sorted by mirror score from mirror status page
## Generated on 2015-12-01
##

Server = http://ftp.tsukuba.wide.ad.jp/Linux/archlinux/$repo/os/$arch

...skip
```

multilib と archlinuxfr の有効化

ArchLinux では、32bit ライブラリが利用できるようにするには手動で multilib というリポジトリを有効にする必要が有ります。結構 32bit のライブラリを利用することが多いので、有効化しておきましょう。

あと、ArchLinux の最大の魅力である AUR (ArchUserRepository) ^{*9} とよばれる非公式な

^{*8} 難しいですか？ここに操作方法が書いてあります https://wiki.archlinuxjp.org/index.php/Nano#nano_.E3.81.AE.E4.BD.BF.E7.94.A8.E6.96.B9.E6.B3.95

^{*9} https://wiki.archlinuxjp.org/index.php/Arch_User_Repository

がらも任意のユーザが書いたインストールスクリプトを簡単に利用出来るようになる `yaourt` というコマンドも使えるようにしましょう。

```
# vim /etc/pacman.conf
\\以下の用に編集

\\multilib の項目をコメントアウト（#を外す）
[multilib]
Include = /etc/pacman.d/mirrorlist

\\上の 2 つの行の下に以下を追加
[archlinuxfr]
SigLevel=Never
Server=http://repo.archlinux.fr/$arch
```

これで `pacman` の設定は完了です。

3.6 必要なパッケージを選択する

初めてだと結構難しいかもしれませんが、冒頭で書いたゴールを具体的に振り返るとこんな感じでしょう。

- `lightdm` + `Cinnamon` + `mikutter`
- 最大限日本語で
- ゴールは `mikutter` で
- `Chrome` も使いたい
- 今後も使えるように.....

これらのことを踏まえると、以下のようなパッケージを入れると良いです。

```
base base-devel lightdm lightdm-gtk-greeter cinnamon nemo gnome-terminal grub
networkmanager yaourt vim xorg-server xorg-server-utils mesa mesa-libgl fcitx-im
fcitx-configtool fcitx-mozc
```

また、GUI 動作のために以下からグラフィックドライバを追加で選択してください。^{*10}

^{*10} 詳しくはここを参照してください。 <https://wiki.archlinuxjp.org/index.php/Xorg#.E3.83.89.E3.83.A9.E3.82.A4.E3.83.90.E3.83.BC.E3.81.AE.E3.82.A4.E3.83.B3.E3.82.B9.E3.83.88.E3.83.BC.E3.83.AB>

- Intel 系 GPU を利用
xf86-video-intel
- 最新の nvidia 系 GPU を利用
nvidia
- AMD 系 GPU を利用
xf86-video-ati

なお NVIDIA のグラフィックカードを利用している方で、とても古い 4 桁番台の nvidia のグラフィックカードを使っている場合、ドライバが変わります。詳しくは次の記事を確認してください

ArchWiki「NVIDIA」

<https://wiki.archlinuxjp.org/index.php/NVIDIA>

さて一気にインストールしてしまいましょう。

ここで、UEFI 環境下で Windows とデュアルブートを検討している人はブートローダを変えたほうが良いです。正確には、変えなくても良いのですが、個人的な感覚として UEFI 環境下の grub はあまり適切に動作しません。もちろん、今後どうなるかわかりませんが、個人的には bootctl(旧:gummiboot) を推奨しています。ちなみにこれは、systemd と呼ばれる必須のパッケージに組み込まれているので新たにパッケージを選ぶ必要はありません。

また、windows の検出には EFI の領域を触るには dosfstools というツールが必要になります。したがって grub の代わりに以下のパッケージを導入してください。

dosfstools

で、もう一個おまけです。もし、あなたが UEFI 環境下ではなく BIOS 環境下の Windows と ArchLinux をデュアルブートをしたいなら grub を選択しつつ、以下のパッケージも追加でインストールしてください。これは、grub が Windows を検知できるようにするパッケージです。

os-prober

以下のコマンドで環境を構築します。

```
# pacstrap /mnt base base-devel lightdm cinnamon nemo \\  
gnome-terminal grub networkmanager yaourt vim \\  
xorg-server xorg-server-utils mesa mesa-libgl \\  
fcitx-im fcitx-configtool fcitx-mozc <GPU ドライバ>
```

コマンド中の \\ は改行を表しています。入力せず続けて入力してください。コマンドを実行すると、否応無しにデフォルトパッケージがインストールされます。^{*11}

3.7 必要最低限の設定をする

新環境に入る

まずは、新環境に入るのですが pacman の設定を引き継ぐとともに fstab の設定をしましょう。以下のコマンドを実行してください。

```
# cat /etc/pacman.conf > /mnt/etc/pacman.conf  
# genfstab -p /mnt >> /mnt/etc/fstab
```

さて、準備は大丈夫ですね。新環境に入りましょう。

```
# arch-chroot /mnt /bin/bash
```

実行すると新環境に移行します。^{*12}

時刻の設定

時刻の設定を行います。日本のタイムゾーンである Asia/Tokyo を選択し、システム時計を UTC にセットします。

```
# ln -s /usr/share/zoneinfo/Asia/Tokyo /etc/localtime  
# hwclock --systohc --utc
```

^{*11} 実は、確認フェーズを省略するようにしてあります。もし pacstrap で確認を受けたいのであれば -i オプションをつけてインストールしてください。

^{*12} もしかしたら味気ない使いづらそうな画面だと感じたかもしれません。archiso では、標準で zsh というシェルを利用しているのに対し、新環境には bash という一般的なシェルで入りました。もし、どうしても前のやつが良いのであれば、新環境で `pacman -Syy && pacman -S --noconfirm zsh grml-zsh-config && zsh` を実行してください。

Windows とデュアルブートを検討している人は注意してください。というのも、Windows は localtime(コンピュータの時計が日本の時刻) になっているからです。-utc は内臓時計が utc で動いていると設定するもので、Windows から見ると時刻がずれてしまいます。修正方法は、Windows のレジストリを編集して Windows を UTC で動作するように設定することです。詳しくはこちらを参照してください

「時刻#Windows で UTC を使う」

https://wiki.archlinuxjp.org/index.php/%E6%99%82%E5%88%BB#Windows_.E3.81.A7-UTC_.E3.82.92.E4.BD.BF.E3.81.86

ただ、ArchLinux が合わせるということも出来ます。もし、あなたが ArchLinux を localtime で動作させたいなら、-utc ではなく -localtime に変更し、実行してください。

言語の設定

言語の設定を行います。日本語と仮定して行いますがもしあなたが別の言語を使っているのであれば、以下のコマンドは実行せずに、エディタを利用して/etc/locale.gen から自分の使っている言語をコメントアウトして保存してください。^{*13}

```
# echo 'en_US.UTF-8 UTF-8' >> /etc/locale.gen
# echo 'ja_JP.UTF-8 UTF-8' >> /etc/locale.gen
# locale-gen
```

また、システムが動作する言語とキーマップを指定しましょう。^{*14}

```
# echo LANG=ja_JP.UTF-8 > /etc/locale.conf
# echo "KEYMAP=jp106" > /etc/vconsole.conf
```

ホスト名とネットワークの設定

ホスト名の設定、ネットワークの設定を行います。

^{*13} そもそも、それが正しいやり方なのですが端折ります。その方法は次の URL を確認してください「ビギナーズガイド#ロケール」<https://wiki.archlinuxjp.org/index.php/%E3%83%93%E3%82%AE%E3%83%8A%E3%83%BC%E3%82%BA%E3%82%AC%E3%82%A4%E3%83%89#.E3.83.AD.E3.82.B1.E3.83.BC.E3.83.AB>

^{*14} もしあなたが筆者のように US 配列以外のキーボードは使えず、US 配列を利用しているユーザは echo "KEYMAP=jp106" ... のコマンドは実行しないでください。

```
# echo "<好きなホスト名>" > /etc/hostname  
# systemctl enable NetworkManager
```

そういえば、systemctl というコマンドを利用しました。これは、簡単に言ってしまうと裏側で立ち上げておきたいサービスを自動起動するようにするためのコマンドであると、認識しておいてください。プロの方々には怒られてしまいそうですが、ひとまず最初はこのような認識で良いと思います。詳しくは、以下の記事を確認してください。

「Systemd」

<https://wiki.archlinuxjp.org/index.php/Systemd>

ところでここでは、NetworkManager によってネットワークを管理するように設定しました。これは、セットアップしてるコンピュータが無線 LAN を利用していて、インタラクティブに設定を変更したほうが生産性が高いと判断したためです。しかし、有線 LAN のみの接続である場合は dhcpcd(<https://wiki.archlinuxjp.org/index.php/dhcpcd>) を利用しても良いですし、もし、サーバのように IP アドレスが固定でネットワークが変わらないのであれば、netctl(<https://wiki.archlinuxjp.org/index.php/netctl>) を選択するのも良いです。

ユーザの設定

新環境で利用するユーザを作成しましょう。ここでは root のパスワードを決定し、新規ユーザの登録とパスワードの設定、そして新規ユーザが sudo (一時的な権限昇格) を許可するように設定しましょう。


```
# passwd
Enter new UNIX password: \\入力しても表示されないが実際は入力されてる
Retype new UNIX password:
passwd: password updated successfully

# useradd -m -g users -G wheel <ここにユーザ名>

# passwd <先ほどのユーザ名>
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully

\\ エディタに nano を選んでるならば.....
# EDITOR=nano visudo
\\ エディタに vim を選んでいるならば.....
# visudo

\\ 以下の %wheel ALL=(ALL) ALL の行のコメントを外す .

...skip

## Uncomment to allow members of group wheel to execute any command
%wheel ALL=(ALL) ALL

...skip
```

これで、ユーザ作成は完了しました。

ところで、脚注で説明した zsh に切り替えている人はいらっしゃいますか？もしかしたら、もう bash でログインはしたくないという人もいるかもしれません。以下のコマンドで、zsh に切り替えられます。

```
# chsh -s /bin/zsh
```

一般ユーザでもログイン後同様のコマンドを入力すると zsh に切り替えられます。.....今頃申し訳ないかもしれませんが、useradd のコマンド時に -s /bin/zsh を引数に加えることでユーザ作成時から zsh にすることも出来ます。

ブートローダの設定

ブートローダの設定です。以下のようなコマンドを実行してください。

```
# grub-install --recheck /dev/sdX
Installing for i386-pc platform.
Installation finished. No error reported.

# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/initramfs-linux.img
Found fallback initramfs image: /boot/initramfs-linux-fallback.img
done
```

もし、エラー表示になってしまった場合は、パーティションテーブルを見なおして見てください。

もし、UEFI 環境下で Windows とデュアルブートをしているのであれば、以下の URL を参考にしてください。

「ビギナーズガイド#UEFI マザーボードの場合」

https://wiki.archlinuxjp.org/index.php/%E3%83%93%E3%82%AE%E3%83%8A%E3%83%BC%E3%82%BA%E3%82%AC%E3%82%A4%E3%83%89#UEFI_.E3.83.9E.E3.82.B6.E3.83.BC.E3.83.9C.E3.83.BC.E3.83.89.E3.81.AE.E5.A0.B4.E5.90.88

各種サービスの有効化

お疲れ様です、あともう少し。起動時にログインマネージャが動作するように有効化しましょう。

```
# systemctl enable lightdm
```

これで、ひとまず GUI 環境は整いました。exit して reboot しましょう！

```
# exit
# reboot
```

なお、再起動後に usb メモリから boot するようにしている場合は解除してください。

3.8 やったぜ

.....と思ったら、上手く行っていないことに気がつくはずです。そう、文字が汚いし、そもそも mikutter とかをインストールしてないのです。由々しき問題です、すぐにインストールしましょう。Ctrl+Alt+T を押すと仮想ターミナルが起動します。そちらからインストールを行いましょう。

```
$ yaourt -S --noconfirm\\
mikutter google-chrome ttf-koruri ttf-vlgothic
Password: <一般ユーザのパスワードを入力>
```

これで準備は完了です。それでは mikutter からつぶやきましょう。仮想ターミナルに以下のコマンドを入力してください。特に&マークを忘れないようにしましょう。これが無いとターミナルを閉じた瞬間に mikutter が終了してしまいます。

```
$ mikutter &
```

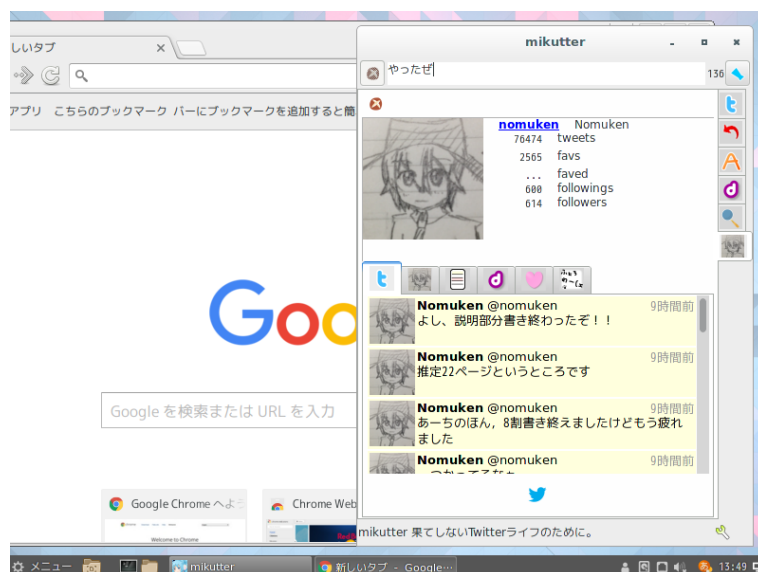


図2 インストール完了

昔の ArchLinux では、fcitx の自動起動はありませんでした。今の自動起動は autostart にインストール後登録されすぐ使えるようになっています。この流れが便利かどうか、もしくは望まれることであるのかは各個人の好き嫌いがあると思います。それはさておき、起動しない場合はターミナルで以下のコマンドを入力してください。

```
$ fcitx &
```

何故こんな話をするかというと、例えば xmonad や qtile のような GNOME、KDE 系でないデスクトップマネージャを利用した場合は自動起動しないためです。

4 インストール後

さて、あなたは完全に動作する ArchLinux の環境を手に入れました。まさか、インストールを終えて満足なんてことはありませんよね？ ArchLinux は超便利です。100 台近くある授業用のコンピュータにも、サーバにも、RaspberryPi にも使える柔軟でシンプルな Linux です。

もちろん Windows の代わりとして使うことだって出来ます。ぜひ、もっともっと ArchLinux を使って自分にとって最も使いやすい環境を構築してください。ここから先は以下の記事が役に立つでしょう。

「一般的な推奨事項」

<https://wiki.archlinuxjp.org/index.php/一般的な推奨事項>

今回は、万人受けする一般的な環境の構築を行いました。Cinnamon は美しくて綺麗なデスクトップマネージャです。しかし、筆者は Qtile と呼ばれるデスクトップマネージャを利用しています。こんな具合に人それぞれの環境があります。ぜひいろんな人の環境を調べて自分の環境に取り込んでいければ良いのかなと思います。

4.1 パッケージマネージャ yaourt の一般的な使い方

最後にパッケージマネージャの簡単な使い方を紹介します。

- パッケージの検索

```
yaourt < パッケージ名 >
```

- パッケージのインストール

```
yaourt -S < パッケージ名 >
```

- パッケージの削除

```
yaourt -R < パッケージ名 >
```

- パッケージの更新

```
yaourt -Syua
```

- データベースの更新

```
yaourt -Syy
```

特に、一日以上開けて ArchLinux をのパッケージをインストールしたりする場合は、データベースの更新を忘れないようにしましょう。

もし、あなたが長期間に渡って ArchLinux をアップデートせずキーチェックでエラーが出てしまうようなら以下のコマンドを試してみてください。

```
$ sudo pacman -S archlinux-keyring
```

<あとがき>

というわけで、書き上がりました。あーちのほん。
締め切りギリギリのため隣に現在末広町のマクドナルドの一角にて
@wakamesoba98氏監修のもと執筆が行われていました。

そもそもこのような状態になってしまった理由を述べると、
私の可愛いサーバさんが、Nomuken氏の操作ミスにより、
爆発炎上したことから始まります。
もう、lxcは嫌です。僕はDockerに生きます。

最後まで迷惑をかけた@wakamesoba98氏に多大なる感謝とともに、
スペシャルエディションの表紙に顔写真を提供してくださった、
@porisuteru氏に感謝します。

できればもう本は書きたくないです。



The book of Arch

Nomuken (@nomuken) ・ 著

落丁本・乱丁本はお取り替えしません，諦めてください

-

発行日: 2015/12/30