

閲覧に注意なデータを外に出さないための

教室内 Web サーバの構築の 手引き

Linux (Ubuntu) 版
Samba 導入と web サーバ構築
令和 5 年 2 月

株式会社 穴吹カレッジサービス

目 次

はじめに	1
Linux とは	2
Linux のメリット	3
Linux のデメリット	4
第1章 Linux の導入	
Linux のディストリビューション	6
Ubuntu のインストール	7
Ubuntu 起動用 USB メモリ作成	12
OS の書き換え	18
Windows10 に Hyper-V で仮想マシンを作成する場合	24
第2章 Ubuntu を使う	
Ubuntu の基本的な使い方	27
コマンドライン	33
Samba のインストール	38
Samba の設定	41
ユーザーの追加	53
Ubuntu のアップグレード	56
第3章 Web サーバの構築	
サーバとは	63
Apache のインストール	64
Apache のコマンド	69
設定ファイル	70
Web サーバ動作確認	71
ポート・IP アドレス	73
ファイアウォール設定	74
ufw について	75
ポート開放	76
HTTP サーバの動作確認	78
HTML ディレクトリ	82
サーバの仕組み	93

はじめに

専門教育を行う上で、閲覧に注意が必要な専門的な文書、画像、音声、動画などセキュリティレベルを強くしなければならないデータを取り扱う場面が多くあります。

このようにセキュリティレベルが非常に強いネットワークを構築しなければいけない場合、最適な方法は物理に分離する方式です。つまり、内部ネットワークと外部接続ネットワークを物理的にわけます。これにより外部からの侵入や攻撃が出来なくなります。

そこで、まず教室内のみのネットワークを構築して外部に出したくないファイルをサーバに格納するためファイルサーバを構築し、次に無線でデータを送信する手順例を作成しました。

また、Window や Mac OS など OS が数年ごとにバージョンアップし、ソフトウェアもバージョンアップされ、古い OS 上でソフトが動かなくなり PC が使えなくなる場面を多く見えています。そのような PCこそ Linux で再生し再利用できたらと思い、Windows7 など古くて使えなくなった PC を使います。そのため非常に動作が軽く、様々な無料ソフトが用意されている Linux を PC にインストールします。

このテキストでは、使われなくなった

T社

2011年発売

windows7

メモリ 8GB

Intel i7-M640 2.8GHz

250GB HDD

ノートパソコンを使用します。

Linux（リナックス）とは

Linux は Windows や Mac OS と同じ OS の一種です。OS とは「オペレーティングシステム（Operating System）」の略語で、アプリケーションソフトとハードウェア（機械）をつなぐ役割を担っています。エクセルなどの「ソフトウェア」で作成した表を、ハードウェアに保存するとき、正しく保存したり、データを読み込んだりできるのは、OS が正しく働いているからです。また、OS はユーザーの操作（マウスやキーボード、音声など）をコンピュータへ正しく伝える役割ももっています。

Linux が Windows や Mac OS と異なる点は、「無料で使えるオープンソース」であることです。オープンソースとは、世界中に公開されている無料のソースコードです。これにより、無料の Linux をベースとした開発や改良を誰でも行え、世界中のユーザーに公開・配布できます。そのため国内外のユーザーが自由にカスタマイズを行い再配布しています。

現在、企業を中心に使われている Linux は、CentOS、Red Hat、Ubuntu などです。最も多く利用されている CentOS のサポートが終了することから、今後は新しい AlmaLinux や Rocky Linux が使われると予想されています。ところが、これらの OS はまだインターネット上の情報も多くなく、不具合な点もおおいので、このテキストでは、インターネットに多くの情報が出ている Ubuntu を使ってサーバの構築をしていきます。

将来的には AlmaLinux や Rocky Linux が主流になっていきますが、基本的な考え方は変わらないので、Ubuntu を使っていればすぐ移行できると思います。

Linux のメリット

<無料>

Windows や Mac OS などの OS は有料で提供されていますが、Linux は無料で公開・配布されているため、誰でも手軽に使えます。

<カスタマイズができる>

無料で公開・配布されているため、自由にカスタマイズできます。例えば、スマホの OS の Android も Linux がベースになっている OS です。

<自由にサーバを構築できる>

Linux は自由に改良できるオープンソースの OS であるため、必要な機能のみでサーバを構築できます。セキュリティや安定性を求めると多くのサーバが必要になりコストが発生します。その点 Linux は無料であるため、コストを大幅に抑えてサーバを立てられるのが魅力です。

<高い安全性>

Linux は無料で使えるうえ、ユーザー権限が厳しく管理されています。常に厳重なセキュリティ体制のもとで最新版をリリースしているため、安全性の高いサーバとして活用できます。

<低スペック PC で使える>

Linux は、必要最低限の機能だけを備えた OS です。したがって、動きが非常にスピーディーという利点があります。

Linux のデメリット

＜他の OS 用のソフトが使えない＞

Microsoft Office などのソフトは使えません。しかし、その代わりとなる統合ソフトは無料で使うことができます。

＜周辺機器が正常に動かないことがある＞

USB メモリなど、周辺機器の種類によっては正常に作動しないことがあります。その都度、Linux 専用のフォーマットへ変更する作業が発生するほか、Linux に対応している機器を揃える必要があります。

＜数多くのソフトがあり用途に応じたものを選ぶ必要がある＞

無料で多くのソフトが使えますが、その数も膨大で目的にあったソフトを探すのに少し苦労するかもしれません。

＜サポートしてくれる人が周りにいない＞

ウィンドウズは、ワード・エクセルなどの操作の慣れている人が多くわからないことがあればすぐに聞けますが、Linux は周りに相談できる人が少なくネットに頼るしかない場面が多くあります。

＜バグの修正が常に行われている＞

修正が頻繁に行われているのはよいことですが、修正により新たなバグが発見されるという面もあります。常に最新版を使用することを心がけていても、旧版でないと動作しないということもあります。このあたりが Windows や MacOS と同じようにいかないところです。ですので、デメリットも分かった上で使う必要があります。

第1章 Linuxの導入

Linux のディストリビューション

では、まず Linux のインストールをします。

Linux はオープンソースなので、様々な OS が配布されています。Linux から派生したものを「ディストリビューション」といいます。Linux のディストリビューションは大きく3つに分かれます。

- ①RedHat 系
 - ②Debian 系
 - ③Slackware 系
- です。

それぞれについて解説をします。

RedHat 系

IBM の子会社、RedHat 社が開発しています。ソフトは無償ですが、アップデート・アップグレード・保守サポートが有償です。主に、CentOS、Fedora、Oracle Linux などがあり、Amazon、IBM、富士通、NTT、ソフトバンクなどとパートナーシップを結んでいます。

Debian 系

ボランティアが中心になって開発を進めています。ユーザフレンドリーな考え方があって親切に作られています。人気の高い Ubuntu や Linux mint などがあります。

Slackware 系

安定性・セキュリティが高く、処理速度が速いという特徴がありますが、初心者には少し扱いが難しい部分があります。

この章では、Ubuntu をインストールしファイルサーバを構築します。

大きな流れは、Ubuntu インストール→ファイルサーバ用ソフト(Samba)のインストールと設定を行います。

Ubuntu のインストール

①Ubuntu のダウンロード

<https://www.ubuntulinux.jp/home>
に行きます。



ここで、Ubuntu のダウンロードをクリックします。



上のリンクボタンをクリックすると以下の画面になります。

現在お使いのオペレーティングシステムが Windows や Mac OS の場合でも、Ubuntu デスクトップ版をダウンロードして置き換え、または共存が可能です。

Ubuntu Desktop 22.04.1 LTS は、2027 年 4 月まで無料のセキュリティアップデートとメンテナンスアップデートが保証されています。

Ubuntu Desktop

現在お使いのオペレーティングシステムが Windows 7 以降の OS の場合でも、Ubuntu デスクトップ版をダウンロードして置き換え、または共存が可能です。

[Ubuntu Desktop の詳細](#)

アップグレードの場合は、[シンプルガイド](#)をご覧ください。

推奨システム:

- 2 GHz デュアルコアプロセッサ以上
- 4 GB システムメモリ
- 25 GB のハードドライブ空き容量
- DVD ドライブまたは USB ポート (インストールメディアとして使用)
- インターネットアクセス (推奨)

Ubuntu Desktop 22.04.1 LTS

デスクトップ PC およびノート PC 向けの Ubuntu LTS 版の最新バージョンをダウンロードいただけます。LTS (long-term support (長期サポート)) の略称です。2027 年 4 月までの 5 年間、無料のセキュリティアップデートおよびメンテナンスアップデートが保証されています。

[ダウンロード](#)

[Ubuntu 22.04 LTS release notes](#)

Ubuntu Desktop 22.10

デスクトップ PC およびノート PC 向けの Ubuntu オペレーティングシステムの最新バージョンです。Ubuntu 22.10 では、2023 年 7 月 1 月までの 4 か月間、セキュリティアップデートおよびメンテナンスアップデートが提供されます。

[ダウンロード](#)

[Ubuntu 22.10 release notes](#)



下のリンクボタンをクリックすると以下の画面になります。日本語 Remix イメージの方が、日本語変換が多くされているので、日本語で使用する場合はこちらが便利です。



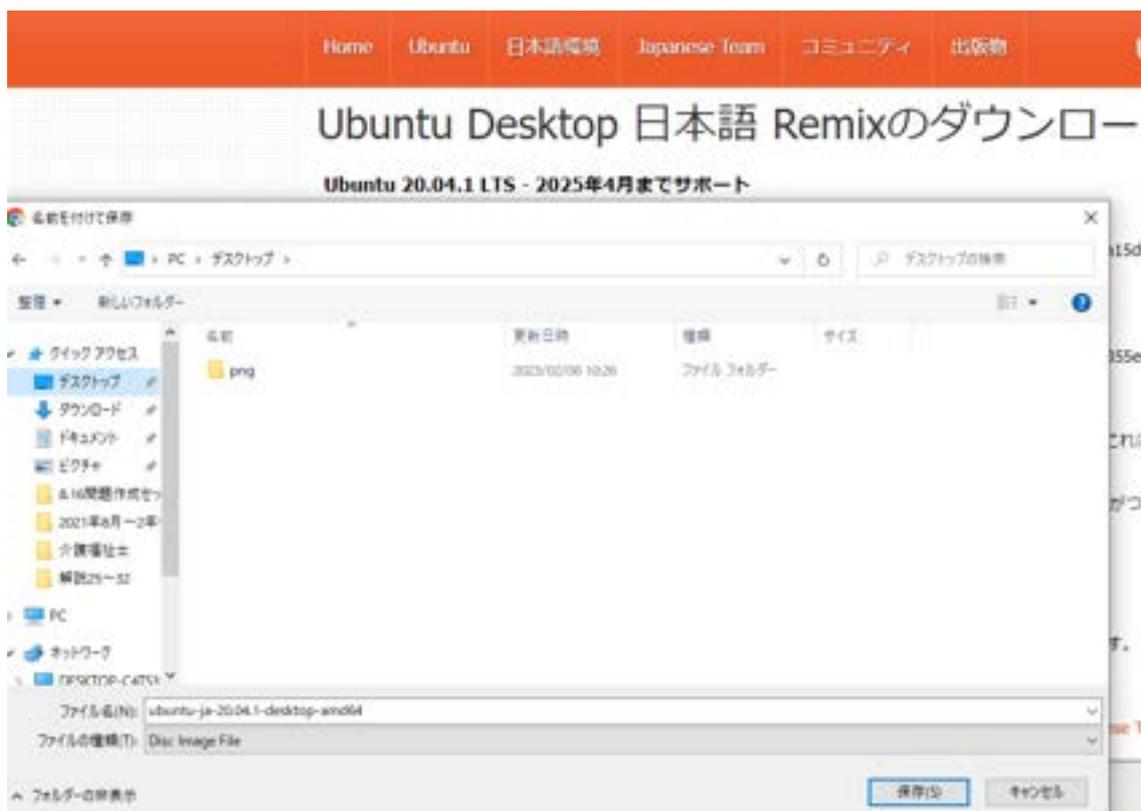
ここでは、2025年4月までのサポートのISOイメージファイルを選びます。

右クリックして「名前を付けてリンク先を保存」を選びます。



この時、インターネットをchromeで閲覧している場合、「危険なのでダウンロードできない」という意味の表示がされます。この場合は、「継続」を選んでダウンロードをすすめてください。

適切な場所にisoファイルを保存します。（ここではデスクトップに保存します）



Ubuntu 起動用 USB メモリ作成

空の USB メモリを用意してください。

起動用 USB を作成できるソフトはいくつかあります。「Rufus」

「USBWriter」「AomeiPE Builder」「ISO to USB」など Windows が正常に起動しない場合のレスキューディスク専用のソフトが有名ですが、ここでは Linux を使うので、「balenaEtcher」を使用します。Linux に対応しているソフトは他には「Ventoy」などがあります。

まず、以下の URL に飛びます。

<https://etcher.io/>



上のページが表示されるので、ダウンロードをクリックします。



このような画面になります。Linuxを入れようとしている古いパソコンは多くの場合、Windows PC だと思うので、64bit の場合は一番上、32bit の場合は3番目をダウンロードします。

OSがWindows10・11の場合、Windows マークと「X」キーを同時に押して、「システム」をクリックし、「システムの種類」に書いてあります。

OSがWindows7・8・8.1の場合、コンピュータアイコンを右クリックして、プロパティを選びます。表示されたコントロールパネルに書いてあります。

ウインドウズ32ビットと64ビットの違い

<動くソフトウェアの違い>

32ビット版のWindowsで実行できるのは32ビット版のソフトのみですが、64ビット版のWindowsは32ビット版、64ビット版のソフトが実行できます。

<メモリの認識>

32ビット版WindowsはRAMが4GBまでしか認識できません。4GBのRAMは、OSやビデオカードに使用されるので、ソフトウェアで利用できるメモリの量は少なくなります。

64ビット版のWindowsはOSによるメモリ容量の制限がありません。64ビット版のWindowsを使用するコンピュータでは、物理的なメモリの量で、どれだけのメモリを利用可能かが決まります。64ビット版Windowsのコンピュータには、通常4GB以上のメモリが搭載されています。

ダウンロードが終わったら、タスクバーに表示されている「balenaEtcher-Setu...exe」をクリックします。そうすると、ライセンス契約書のウィンドウが開くので、「同意する」を押します。



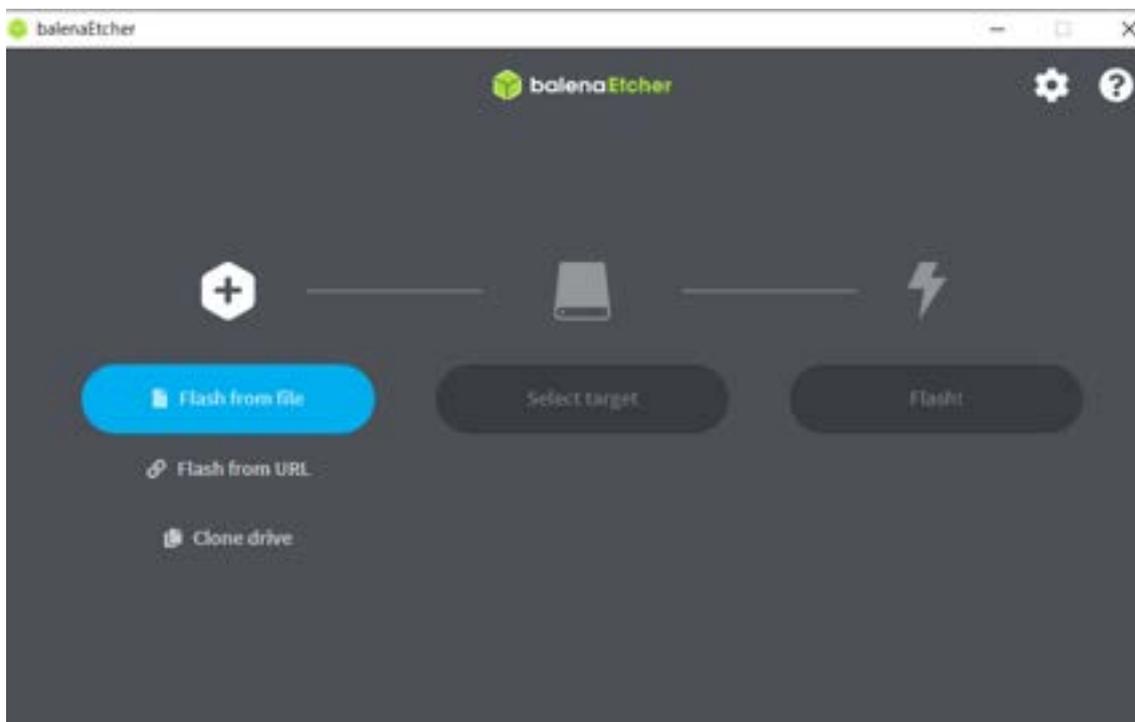
Frequently Asked Questions

Why is my drive not bootable?

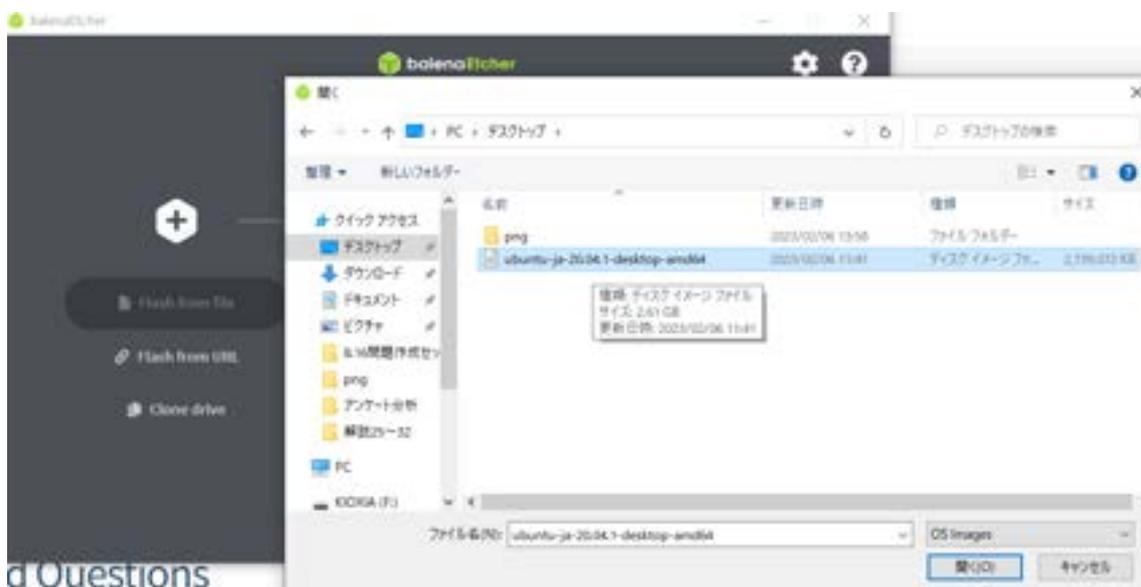
How can I configure persistent storage?



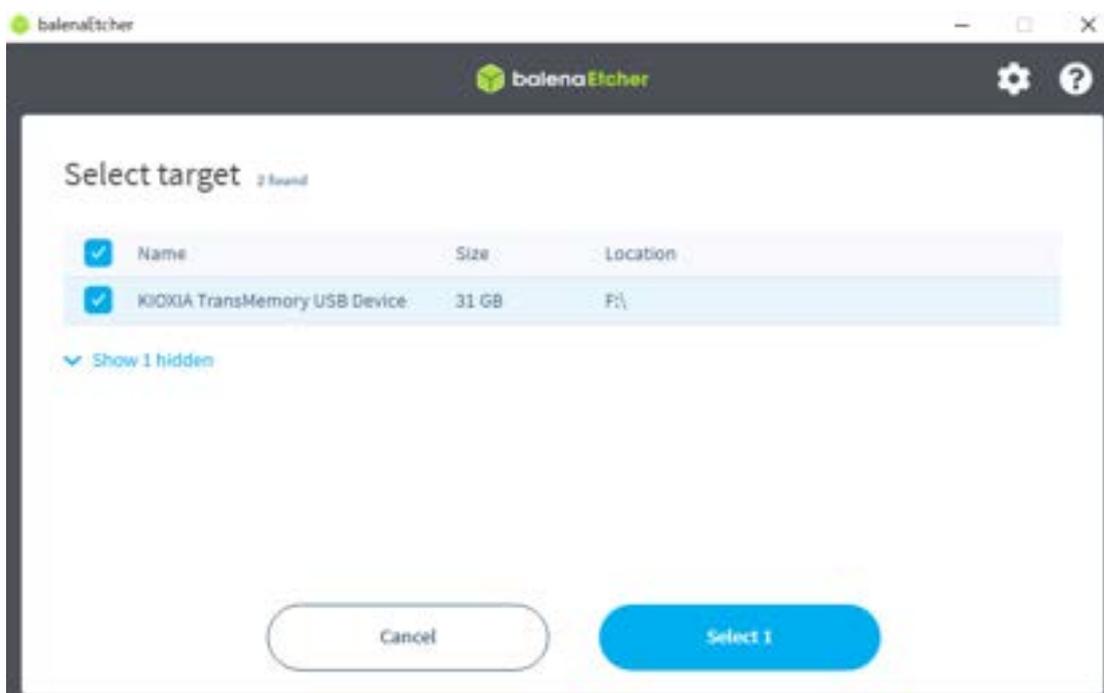
そうするとこのようなウィンドウが開きます。



ここで、用意した空の USB メモリを PC に接続して、一番上の「Flash from file」をクリックします。



先ほどダウンロードした ubuntu のディスクイメージファイルを選び、「開く」を押します。

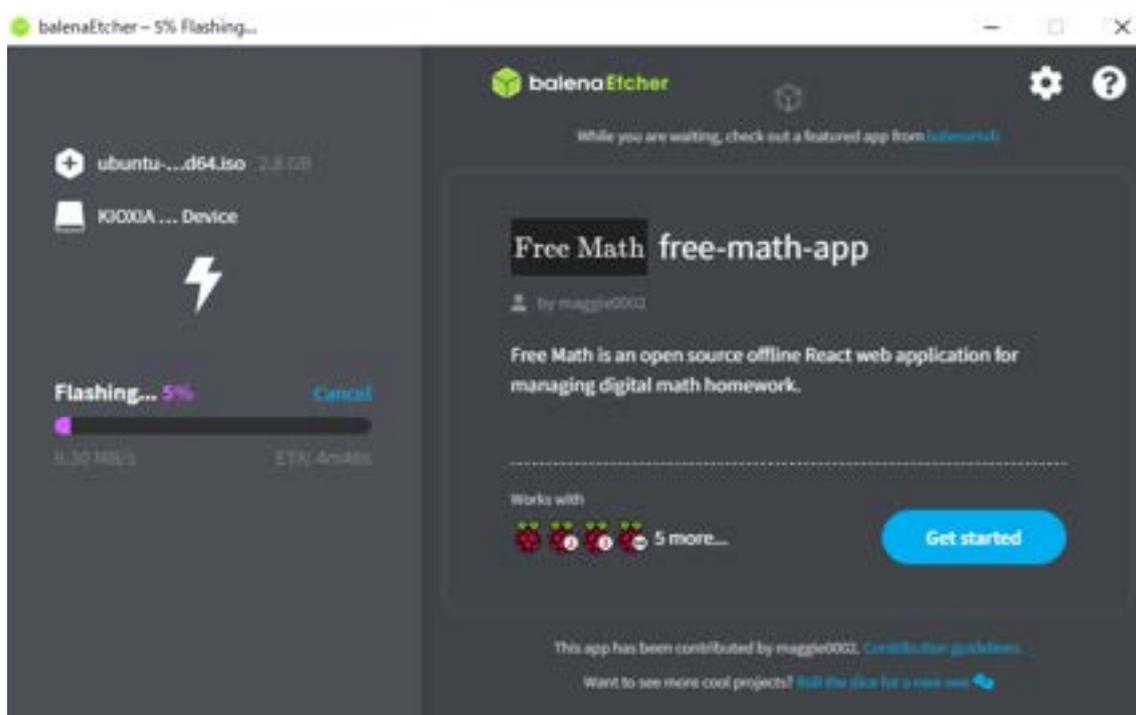


どこに保存するかと聞いてくるので、USB メモリにチェックを入れて、「Select 1」を押します。

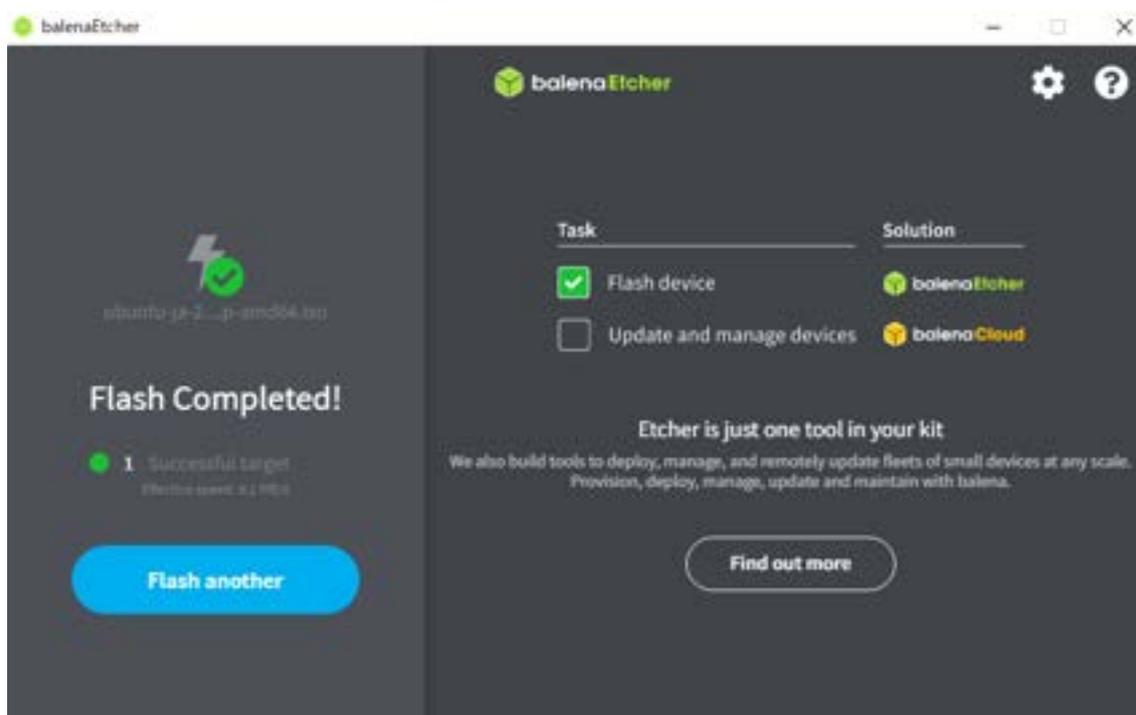
「Flash」を押します。



書き込みが始まります。



USB メモリにイメージファイルが書き込まれ、Linux (Ubuntu) の起動 USB が出来ました。



さて、これから windows で動いていたノートパソコンを、Linux で動くノートパソコンにします。

ノートパソコンをシャットダウンしてください。

OS の書き換え

USB メモリをノートパソコンに差し込んで、ノートパソコンの電源を入れます。

起動するとき、ブートメニュー (BOOT MENU) を出します。

ブートメニューとは、起動するときどこからシステムを読み込むかを指定するメニューで、変更することができます。広い意味で BIOS や UEFI ともいいます。

ブートメニューを立ち上げるためには、電源を入れた直後にメーカーごとに決まったキーを押します。以下に代表的なキーを示しますが、うまくいかない時はインストールをするパソコンの HP で確認してください。

富士通 : F12 か F2

NEC : F2

東芝 : F12

レノボ : F12

HP : F10

DELL : F12

Panasonic : F2

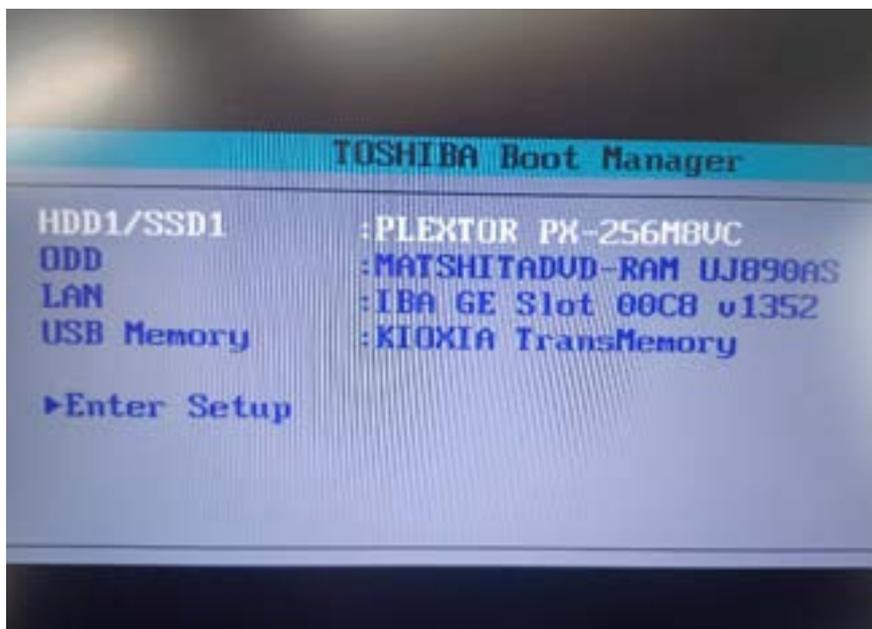
Sony : F2

Acer : F2 か F12

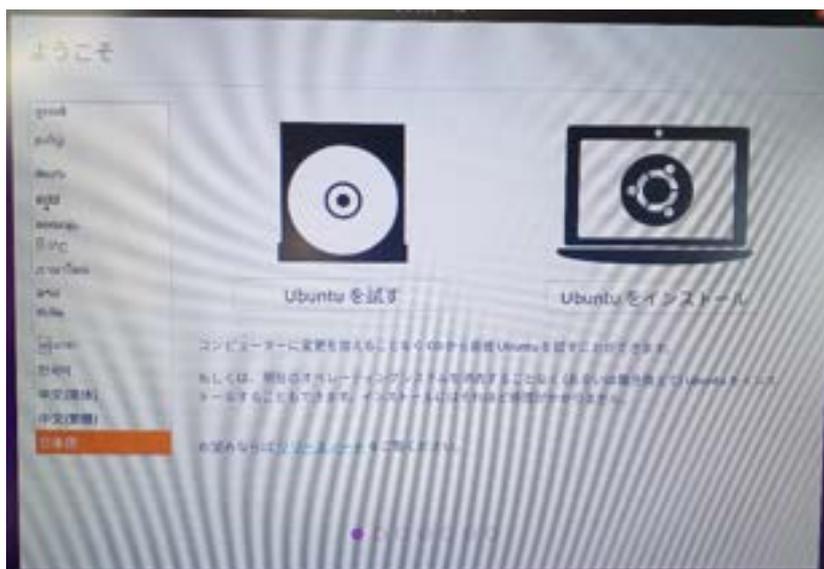
Asus:F8

では、ブートメニューを起動するキーを押しながら電源を入れてください。うまくいかない時は、電源を入れたあとすぐに「トントン」とキーを押し続けてください。

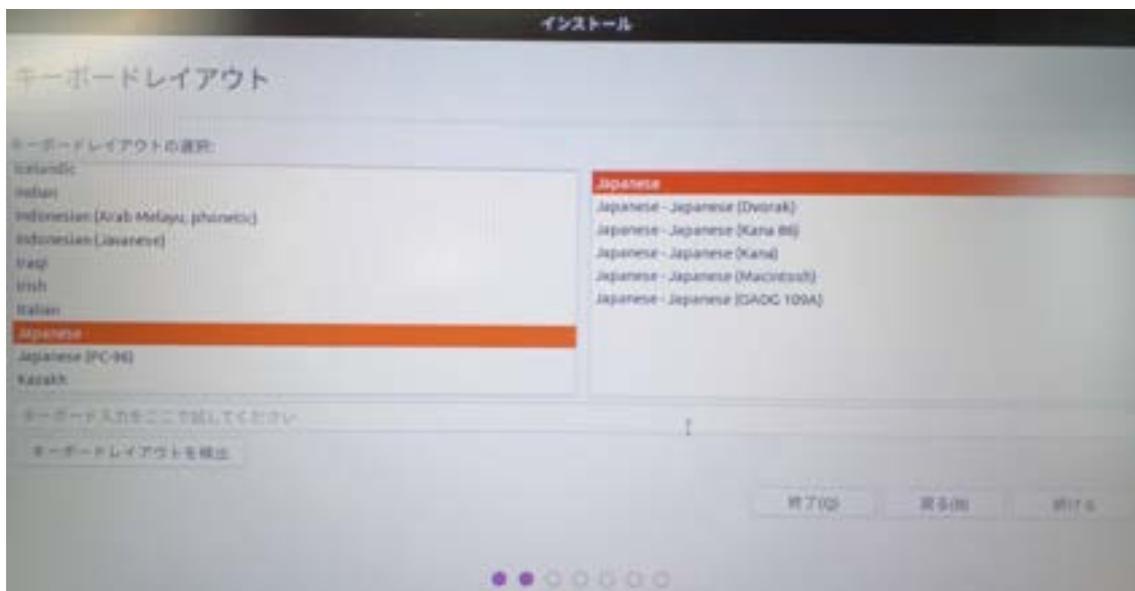
画面のように、どこから起動するかを聞いてきます。この画面では、一番上がハードディスク、2番目がDVD、3番目がLAN、4番目がUSBメモリになっています。ここで、USBメモリを選びます。そうすると、USBメモリに書きこんだシステムを読み込みます。



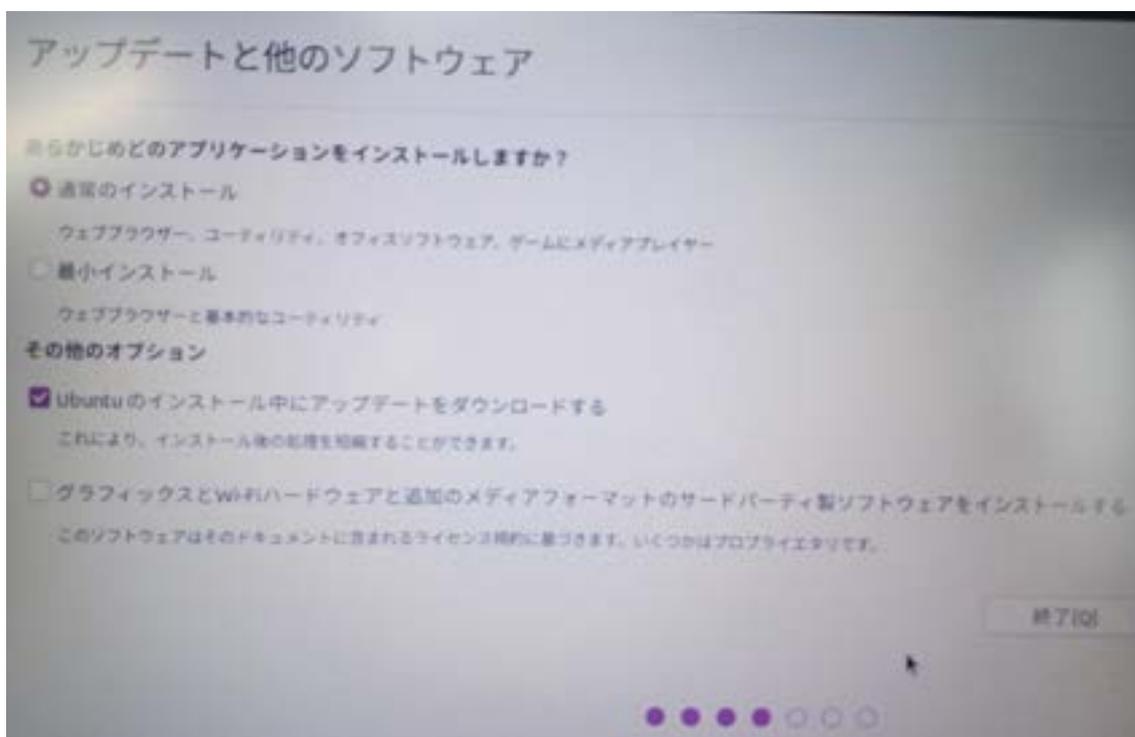
読み込みが終わると、この画面になります。右のUbuntuをインストールを選びます。



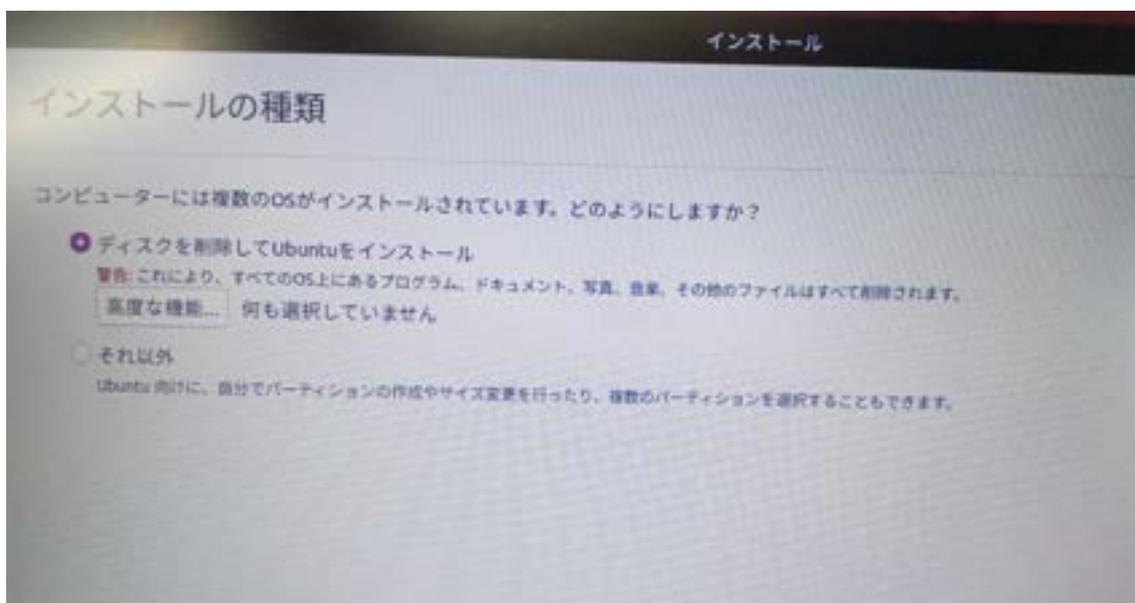
キーボードレイアウトは日本語になっていますので、続けるをクリックします。



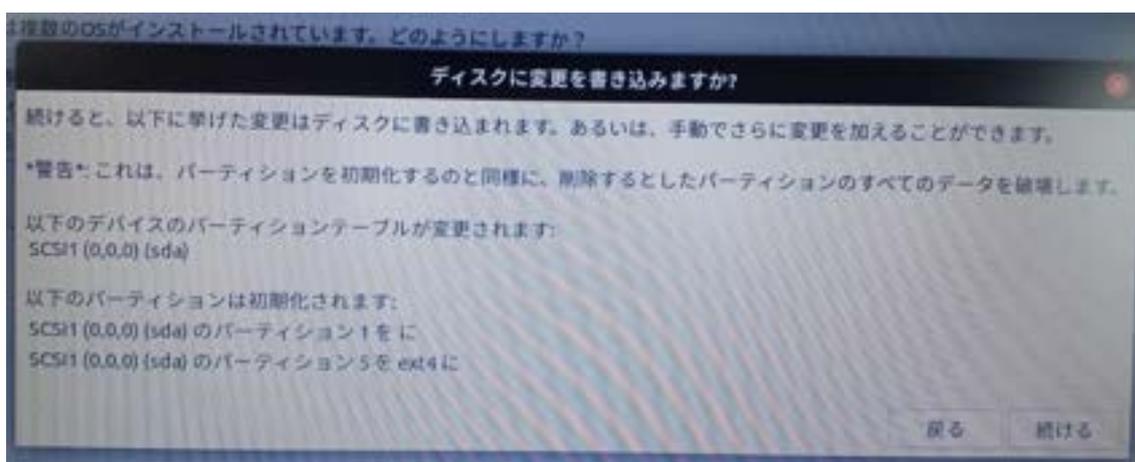
通常のインストールを選んで終了を押します。



インストールの種類は、ディスクを削除してインストールを選びます。
※ウィンドウズとリナックスを共存される場合、これは選択しないでください。修復ができなくなります。共存させる場合は違う手順が必要なので、押さないでください。共存させる場合は「Hyper-V」を使用します。

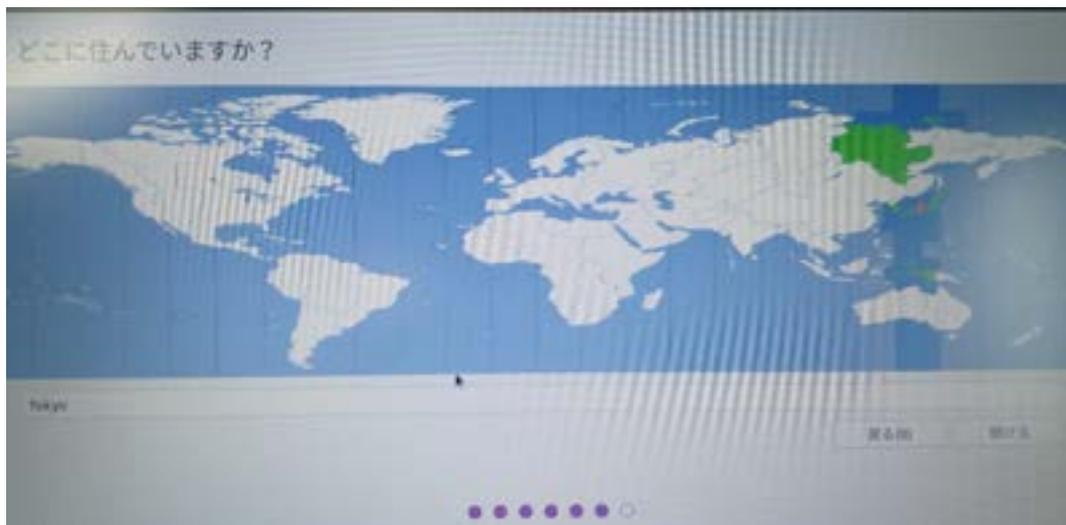


確認画面がでてきます。



「続ける」を押します。

東京になっているので、「続ける」を押します。



ユーザー情報を入力します。

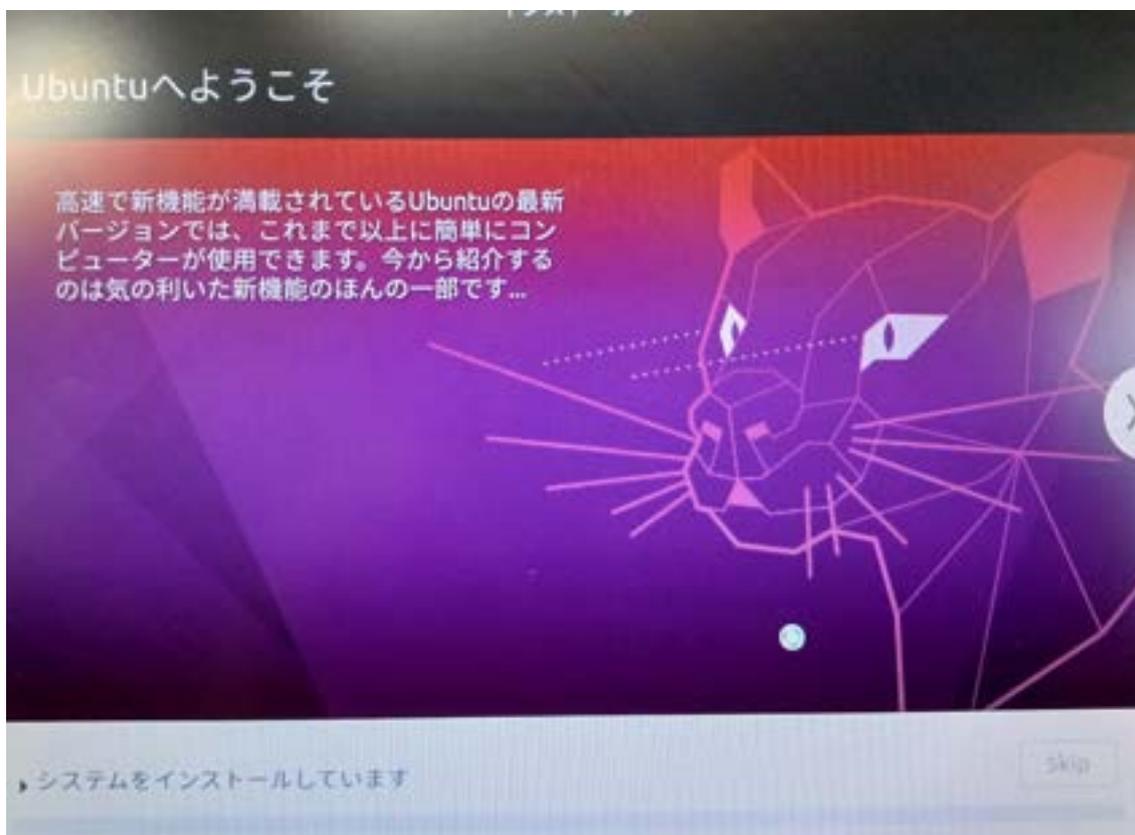
コンピュータの名前を hiro としました。

この名前がユーザー名になるので、自由に名前をつけてください。

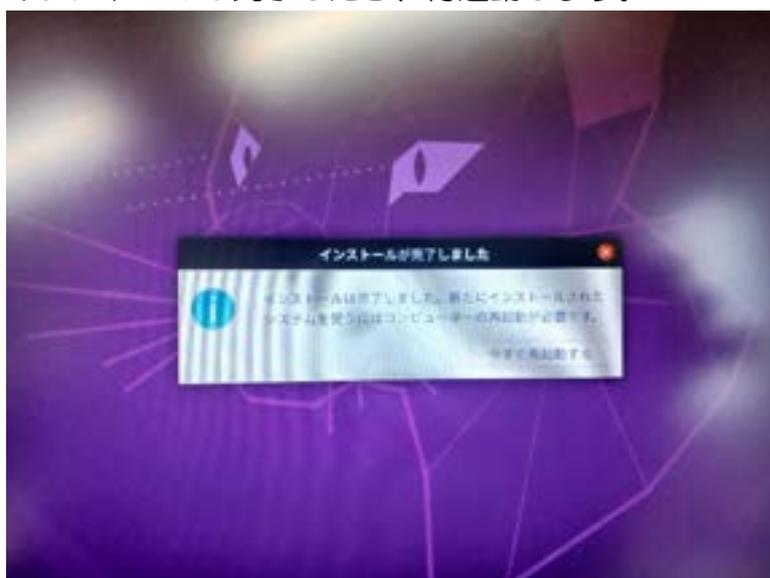
※以降ユーザー名 hiro に表示されているところが、皆さんの名前になります。

パスワードはしばしば聞かれますので、忘れないものにしてください。

システムがインストールされます。



インストールが完了したら、再起動します。



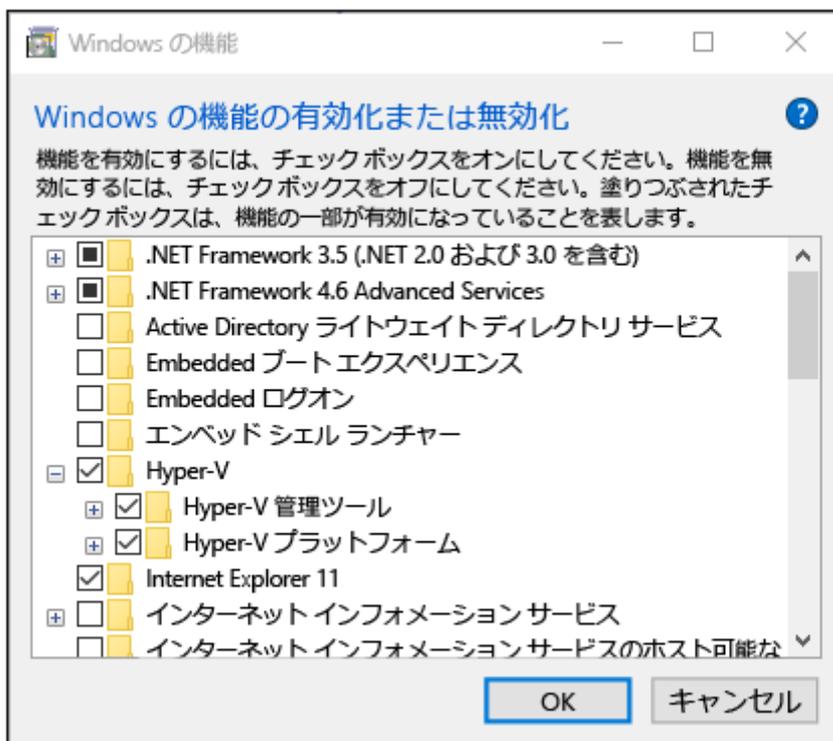
USB を抜いて、再起動(remove)するためにはエンターキーを押してくださいの表記がでたら、エンターキーを押します。

Windows10 に Hyper-V で仮想マシンを作成する場合

Hyper-V を使用すると、Windows 上の仮想マシンとして複数のオペレーティングシステムを実行することができます。例えば、以前のバージョンの Windows（8 や 7、XP、VISTA など）や、Linux を実行することができます。

ただし、64 ビット版の Windows10Pro、Enterprise、Education で使用できますが、Home Edition では使用できません。

Hyper-V を有効にします。



Windows ボタン右クリック、アプリと機能、プログラムと機能、windows の機能の有効化または無効化を選択、その中の Hyper-v のチェックをつけ、OK を押します。

再起動後有効になります。

スタートメニュー → hyper-v マネージャー → 操作 →簡易作成を選択
します。

仮想マシンをカスタマイズします。

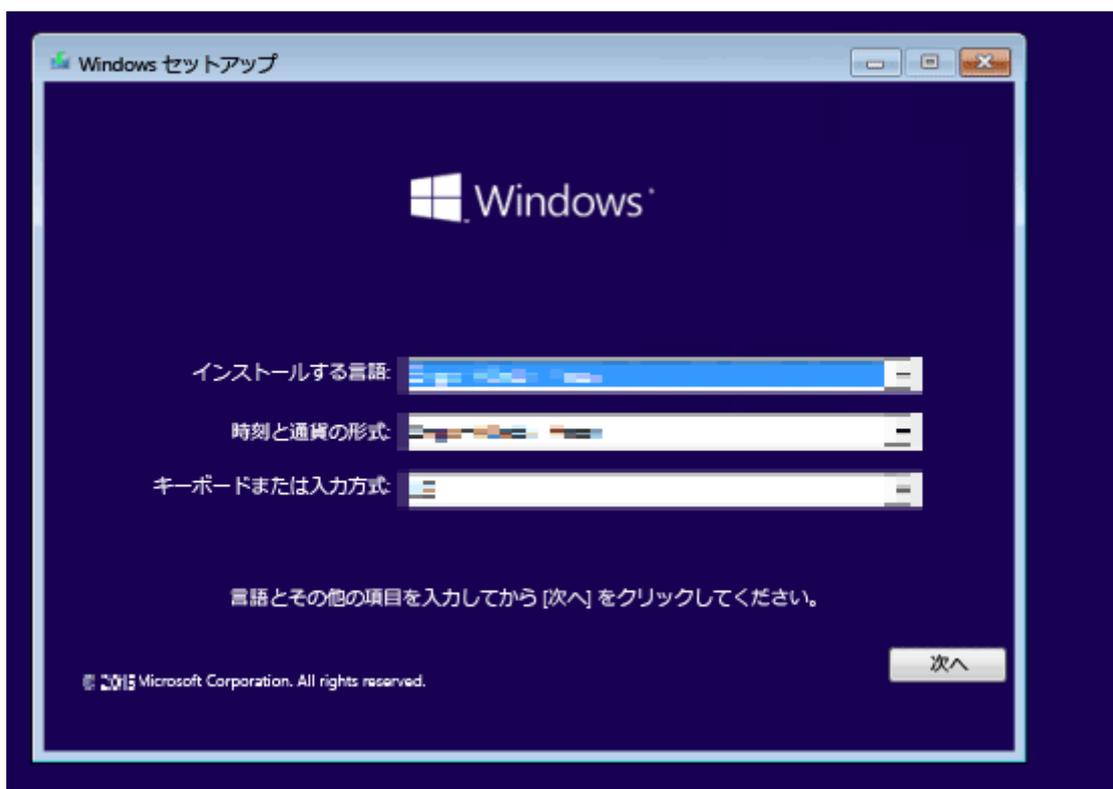
仮想マシンに名前を付けます。

仮想マシンのインストールメディアを選択します。このテキストの場合
は、USBメモリの中の.iso ファイルになります。

仮想マシンに別バージョンの Windows をインストールする場合は、
Windows セキュアブートを有効にできますが、Linux の場合は、オフに
します。

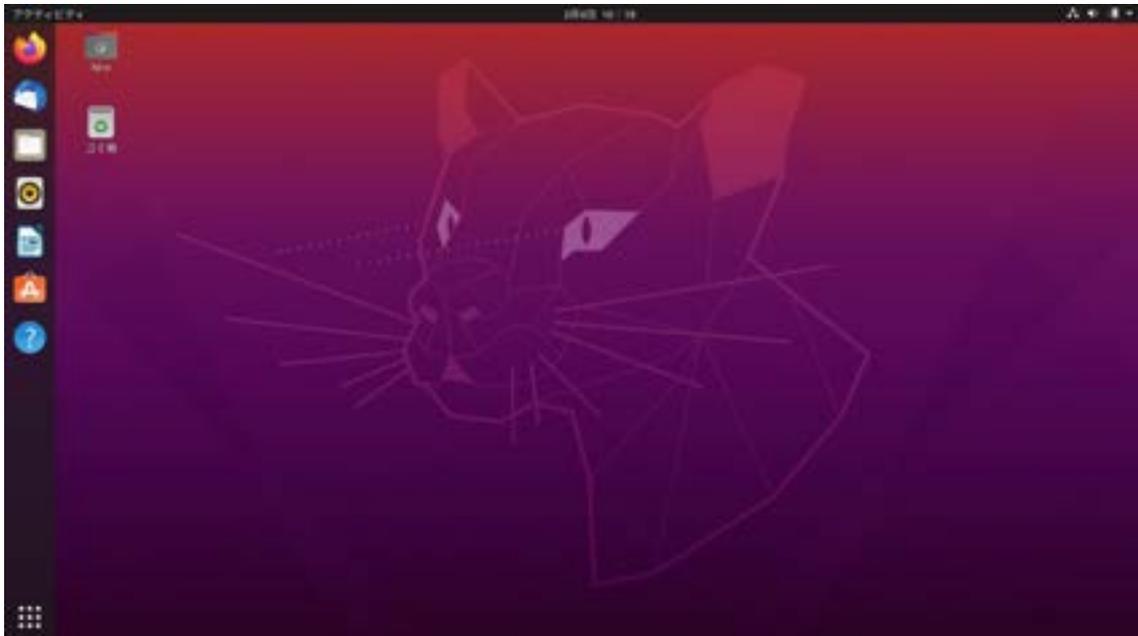
ネットワークを設定します。

[作成] をクリックして、仮想マシンの作成を開始します。



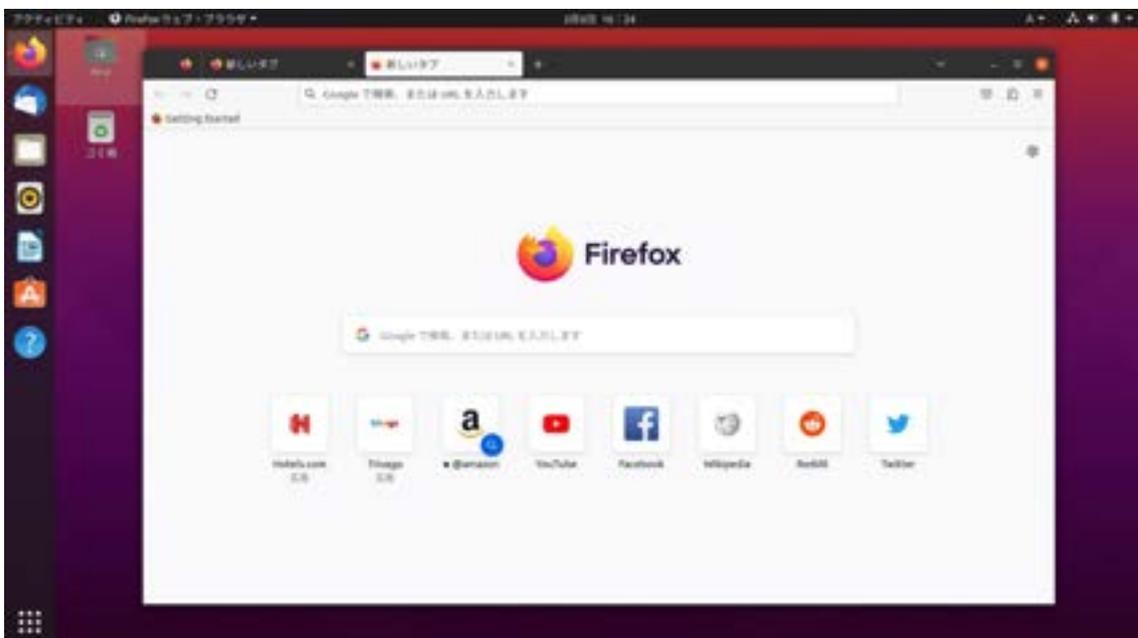
第2章 Ubuntu を使う

Ubuntu の基本的な使い方

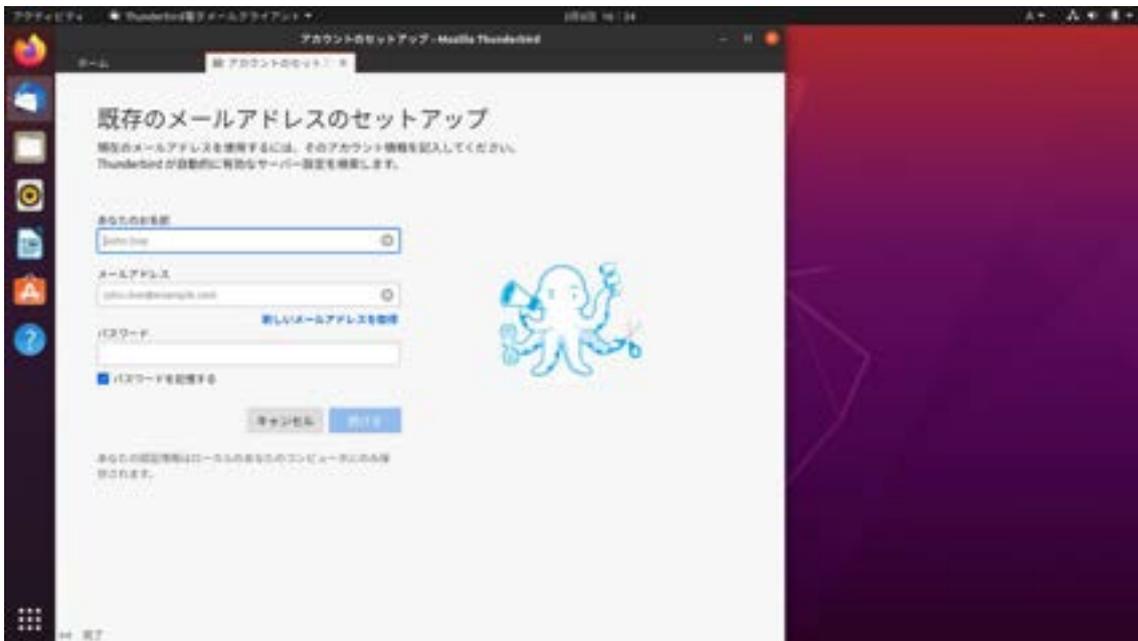


※バージョンによって表示やアイコンが異なります。

一番上のアイコンは、ブラウザ「ファイアーフォックス」です。



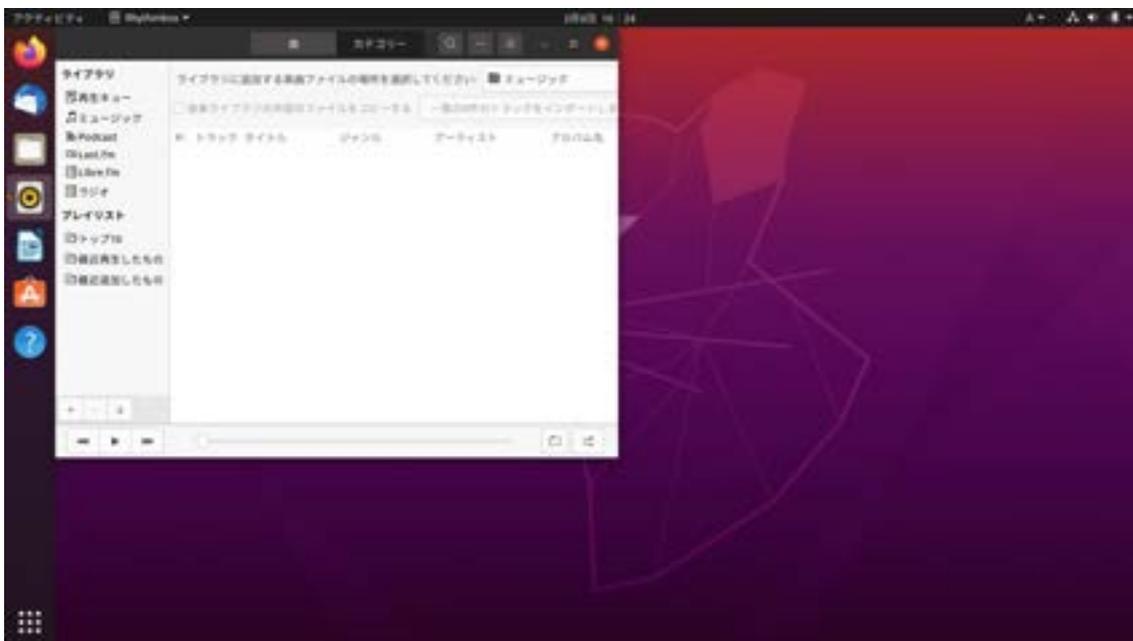
2番目のアイコンは、メールソフト「モジラ サンダーバード」です。



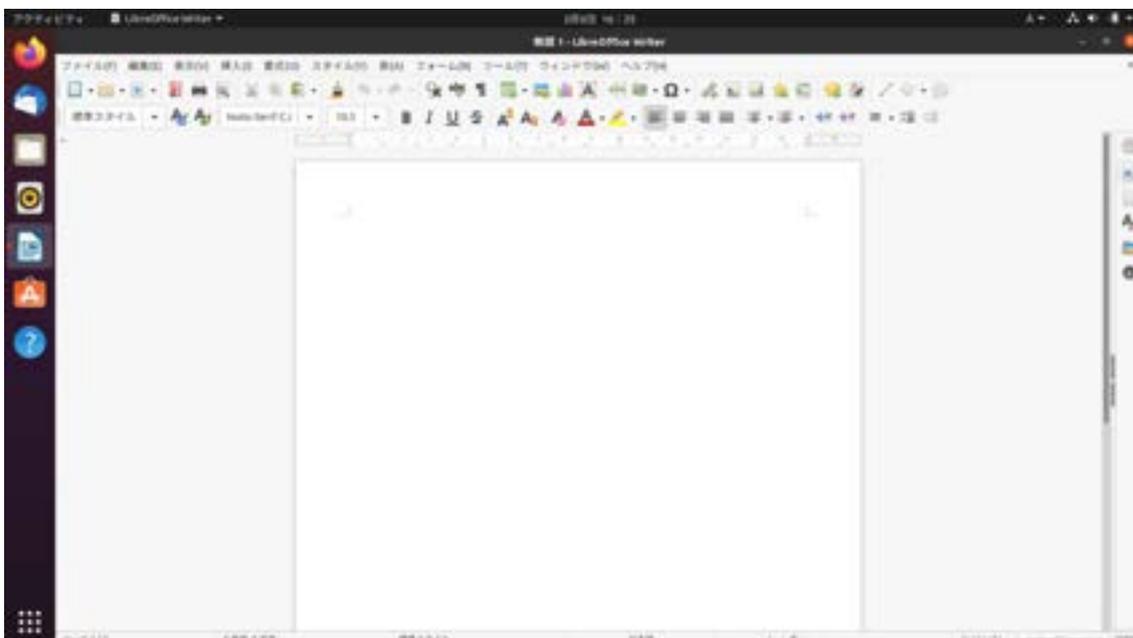
3番目のアイコンはユーザーフォルダーです。



4番目のアイコンは、音楽管理ソフトです。



5番目のアイコンは無料のエディタです。マイクロソフトのワードのようなものです。無料で使えます。ワードファイルとの互換性もあります。



6 番目のアイコンは、Ubuntu で使えるソフトウェアです。ダウンロードは無料です。



上のタブの「インストール済み」をクリックすると、標準でインストールされているソフト一覧がでます。



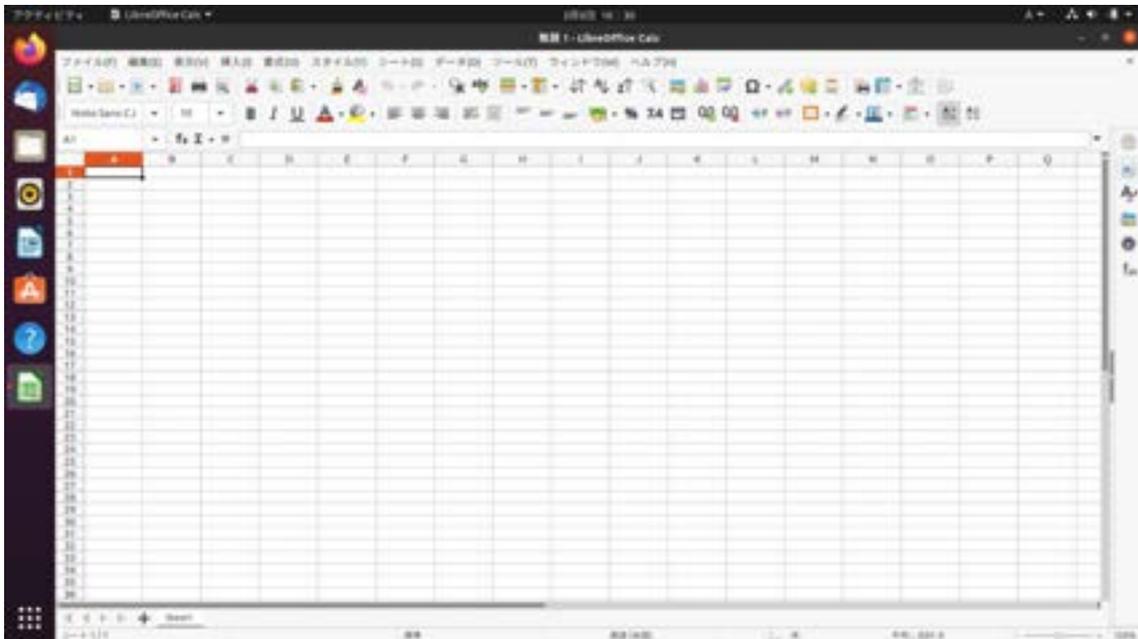
上のタブの「アップデート」をクリックすると、インストールされているソフトでアップデートがあるものができます。



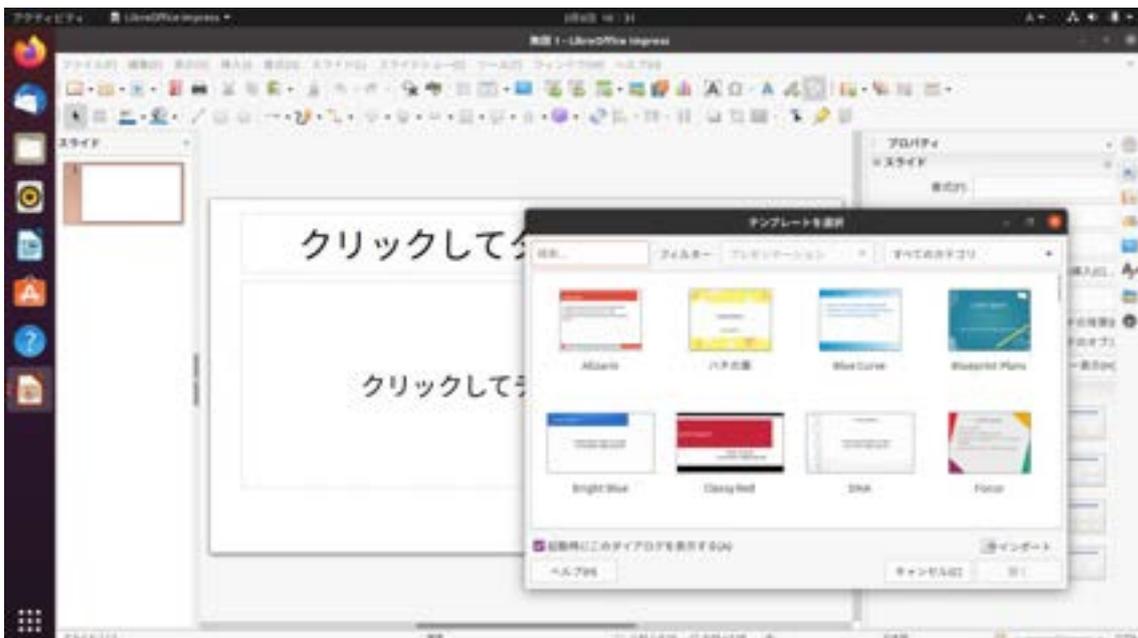
一番したの白い9つの点のアイコンを押すと、インストールされているソフト一覧が表示されます。



LibreOffice Calc を選ぶと、マイクロソフトのエクセルと同じような表計算ソフトが立ち上がります。エクセルファイルと互換性があります。



LibreOffice Impress を選ぶと、マイクロソフトのパワーポイントと同じようなプレゼンテーションソフトが立ち上がります。パワーポイントファイルと互換性があります。



OS だけでなく様々なソフトウェアが無料で使えるのが最大の特徴です。

コマンドライン

では、いよいよファイルサーバにするためのソフト、Samba をインストールします。メーカー製のNASでもLinuxとSambaは使用されています。

ここからは、動作が速いコマンドライン（CLI）を使ってみます。

GUIとCLI

PCは、マウスやキーボードやタッチパネル、音声などの入力装置（インターフェース）から入力された命令を受けて、内部で処理をして、ディスプレイや音声などでアウトプットします。

WindowsやMacOSなどでは、ディスプレイ上に、ウィンドウ、メニュー、ボタン、アイコンなどの画像（グラフィック）が表示され、ユーザーは必要な操作を行うために、画像（グラフィック）をマウスなどのポインティングデバイスやタッチパネルをクリックしています。これをGUI（Graphical User Interface）といいます。直感感覚的に操作ができますが、パソコンの性能がある程度高いものでないと動作が遅くなります。

一方、パソコンに直接命令をするCLI（Command Line Interface）という方法もあります。CLIは、処理速度が遅くメモリの少ないパソコンでも動作が軽いという特徴があります。特に新しいソフトウェアパッケージをインストールするときなどによく使います。

シェル（命令を書く画面）を開きます

※Linuxでは「シェル」、ウィンドウズでは「コマンドプロンプト」といいます。

左下の白い9つの点をクリックしてソフトウェア一覧を開きます。下にあるユーティリティの中の、「端末」を選びます。



そうすると、黒い画面が立ち上がります。これがシェルです。

カーソルがある部分に、

sudo apt update

と入力します。

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
htroghiro-linux:~$ sudo apt update
```

<コマンドの説明>

sudo (superuser do) コマンド：コンピュータの管理者権限でプログラムを実行するときに使います。

apt (Advanced Package Tool)：LinuxのOS管理システムをAPTライブラリといいます。

Update:このAPTライブラリのリストを、最新バージョンにアップデートします。Ubuntuに関する最新の情報を取得するということです。

次に、管理者のパスワードを聞かれます。

Ubuntu をインストールしたときのパスワードを入力します。入力しても何も変わらない場合は、もう一度入力してください。

パスワードが通ると、最新情報をネットから取得します。

```
hiroghiro-linux:~$ sudo apt update
[sudo] hiro のパスワード:
すみません、もう一度試してください。
[sudo] hiro のパスワード:
ヒット:1 http://archive.ubuntulinux.jp/ubuntu focal InRelease
ヒット:2 http://archive.ubuntulinux.jp/ubuntu-ja-non-free focal InRelease
ヒット:3 http://security.ubuntu.com/ubuntu focal-security InRelease
ヒット:4 http://jp.archive.ubuntu.com/ubuntu focal InRelease
ヒット:5 http://jp.archive.ubuntu.com/ubuntu focal-updates InRelease
取得:6 http://jp.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
108 kB を 7秒 で取得しました (14.6 kB/s)
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
アップグレードできるパッケージが 2 個あります。表示するには 'apt list --upgradab
le' を実行してください。
hiroghiro-linux:~$
```

次に、

sudo apt upgrade

と入力します。

```
hiro@hiro-linux:~$ sudo apt upgrade
```

この作業で、Ubuntu が最新のものになります。

```
hiro@hiro-linux:~$ sudo apt upgrade
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
アップグレードパッケージを検出しています... 完了
以下のパッケージが自動でインストールされましたが、もう必要とされていません:
  gir1.2-goa-1.0 libfprint-2-tod1 libfwupdplugin1 libllvnl0 libxmlb1
これを削除するには 'sudo apt autoremove' を利用してください。
以下のパッケージはアップグレードされます:
  update-notifier update-notifier-common
アップグレード: 2 個、新規インストール: 0 個、削除: 0 個、保留: 0 個。
230 kB 中 0 B のアーカイブを取得する必要があります。
この操作後に追加で 116 kB のディスク容量が消費されます。
続行しますか? [Y/n]
```

y を押してエンターキーを押して続行します。

インストールが終わります。

```
これを削除するには 'sudo apt autoremove' を利用してください。
以下のパッケージはアップグレードされます:
  update-notifier update-notifier-common
アップグレード: 2 個、新規インストール: 0 個、削除: 0 個、保留: 0 個。
230 kB 中 0 B のアーカイブを取得する必要があります。
この操作後に追加で 116 kB のディスク容量が消費されます。
続行しますか? [Y/n] y
(データベースを読み込んでいます ... 現在 184395 個のファイルとディレクトリがインストールされています。)
.../update-notifier_3.192.30.16_amd64.deb を展開する準備をしています ...
update-notifier (3.192.30.16) で (3.192.30 に) 上書き展開しています ...
.../update-notifier-common_3.192.30.16_all.deb を展開する準備をしています ...
update-notifier-common (3.192.30.16) で (3.192.30 に) 上書き展開しています ...
update-notifier-common (3.192.30.16) を設定しています ...
新バージョンの設定ファイル /etc/cron.weekly/update-notifier-common をインストールしています ...
update-notifier (3.192.30.16) を設定しています ...
新バージョンの設定ファイル /etc/xdg/autostart/update-notifier.desktop をインストールしています ...
libglib2.0-0:amd64 (2.64.6-1-ubuntu20.04.4) のトリガを処理しています ...
hicolor-icon-theme (0.17-2) のトリガを処理しています ...
hiro@hiro-linux:~$
```

Samba のインストール

ファイルサーバ用ソフト、Samba をインストールします。
以下のコマンドを入力します。

```
sudo apt install Samba
```

```
状態情報を読み取っています... 完了
以下のパッケージが自動でインストールされましたが、もう必要とされていません:
  gir1.2-goa-1.0 libfprint-2-tod1 libfwupdplugin1 libllvnl0 libxmlb1
これを削除するには 'sudo apt autoremove' を利用してください。
以下の追加パッケージがインストールされます:
  attr libverbs-providers libcephfs2 libibverbs1 librados2 librdnacd1
python3-crypto python3-dnspython python3-gpg python3-markdown
python3-packaging python3-pygments python3-pyparsing python3-samba
python3-tdb samba-common samba-common-bin samba-dsdb-modules
samba-vfs-modules tdb-tools
提案パッケージ:
  python-markdown-doc python-pygments-doc ttf-bitstream-vera
python-pyparsing-doc bind9 bind9utils ctdb ldb-tools ntp | chrony
smbldap-tools winbind heimdal-clients
以下のパッケージが新たにインストールされます:
  attr libverbs-providers libcephfs2 libibverbs1 librados2 librdnacd1
python3-crypto python3-dnspython python3-gpg python3-markdown
python3-packaging python3-pygments python3-pyparsing python3-samba
python3-tdb samba samba-common samba-common-bin samba-dsdb-modules
samba-vfs-modules tdb-tools
アップグレード: 0 個、新規インストール: 21 個、削除: 0 個、保留: 0 個。
10.3 MB のアーカイブを取得する必要があります。
この操作後に追加で 68.2 MB のディスク容量が消費されます。
続行しますか? [Y/n] 
```

y を押してエンターキーを押して続行します。

Samba が無事にインストールされたか確認をします。

```
hiro@hiro-linux:~$ systemctl status smb
```

Systemctl status smb

と入力します。

```
hiro@hiro-linux:~$ systemctl status smb
● smb.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smb.service; enabled; vendor preset:
   Active: active (running) since Tue 2023-02-07 08:26:36 JST; 23min ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
   Process: 1395 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile
   Main PID: 1401 (smbd)
    Status: "smbd: ready to serve connections..."
     Tasks: 4 (limit: 9125)
    Memory: 15.3M
    CGroup: /system.slice/smb.service
           └─1401 /usr/sbin/smbd --foreground --no-process-group
             └─1407 /usr/sbin/smbd --foreground --no-process-group
               └─1408 /usr/sbin/smbd --foreground --no-process-group
                 └─1409 /usr/sbin/smbd --foreground --no-process-group

2月 07 08:26:35 hiro-linux systemd[1]: Starting Samba SMB Daemon...
2月 07 08:26:36 hiro-linux systemd[1]: Started Samba SMB Daemon.
lines 1-19/19 (END)
```

smbd は、server message block です。ネットワーク上でファイル共有やプリンタ共有などを行うためのプロトコル（通信規約）に基づいた実行をします。Daemon（デーモン）とあるのでLinuxのバックグラウンドで動作するプログラムです。

緑色でアクティブ（ランニング）となっているので、動作しています。

ドキュメントはsmbd、Samba、smb.confがあります。

「Q」を押して、最終行のEND表示を消してください。

では、次に

```
hiro@hiro-linux:~$ systemctl status nmbd
```

Systemctl status nmbd

と入力します。

```
hiro@hiro-linux:~$ systemctl status nmbd
● nmbd.service - Samba NMB Daemon
   Loaded: loaded (/lib/systemd/system/nmbd.service; enabled; vendor preset:
   Active: active (running) since Tue 2023-02-07 08:26:35 JST; 31min ago
     Docs: man:nmbd(8)
           man:samba(7)
           man:smb.conf(5)
   Main PID: 1366 (nmbd)
   Status: "nmbd: ready to serve connections..."
     Tasks: 1 (limit: 9125)
    Memory: 11.6M
   CGroup: /system.slice/nmbd.service
           └─1366 /usr/sbin/nmbd --foreground --no-process-group

2月 07 08:26:35 hiro-linux systemd[1]: Starting Samba NMB Daemon...
2月 07 08:26:35 hiro-linux systemd[1]: Started Samba NMB Daemon.
lines 1-15/15 (END)
```

nmbd は、ホスト名（コンピューター名）でファイルサーバにアクセスするための名前サーバーです。これに回答するとウィンドウズのマイネットワークやネットワークコンピュータに表示されます。

このように、Samba は主に、「smbd」と、「nmbd」という二つのプログラムで構成されています。

これで Samba のインストールが終わりました。

Samba の設定

次に、Samba の設定を行います。

シェルを立ち上げ

```
Sudo mkdir /home/share/
```

と入力します。

```
htroghiro-linux:~$ sudo mkdir /home/share
```

Mkdir はディレクトリを新しく作るコマンドです。

次にこのディレクトリをフルアクセス（777）にします。書いたり読んだり消したりできることです。

```
Sudo chmod 777 /home/share/
```

```
htroghiro-linux:~$ sudo chmod 777 /home/share
```

次にファイルにアクセスした時間を記録します。(touch)
記録にフルアクセスできるようにします。

```
Sudo touch /etc/libuser.conf
```

```
Sudo chmod 777 /etc/libuser.conf
```

```
hiroghiro-linux:~$ sudo touch /etc/libuser.conf  
hiroghiro-linux:~$ sudo chmod 777 /etc/libuser.conf
```

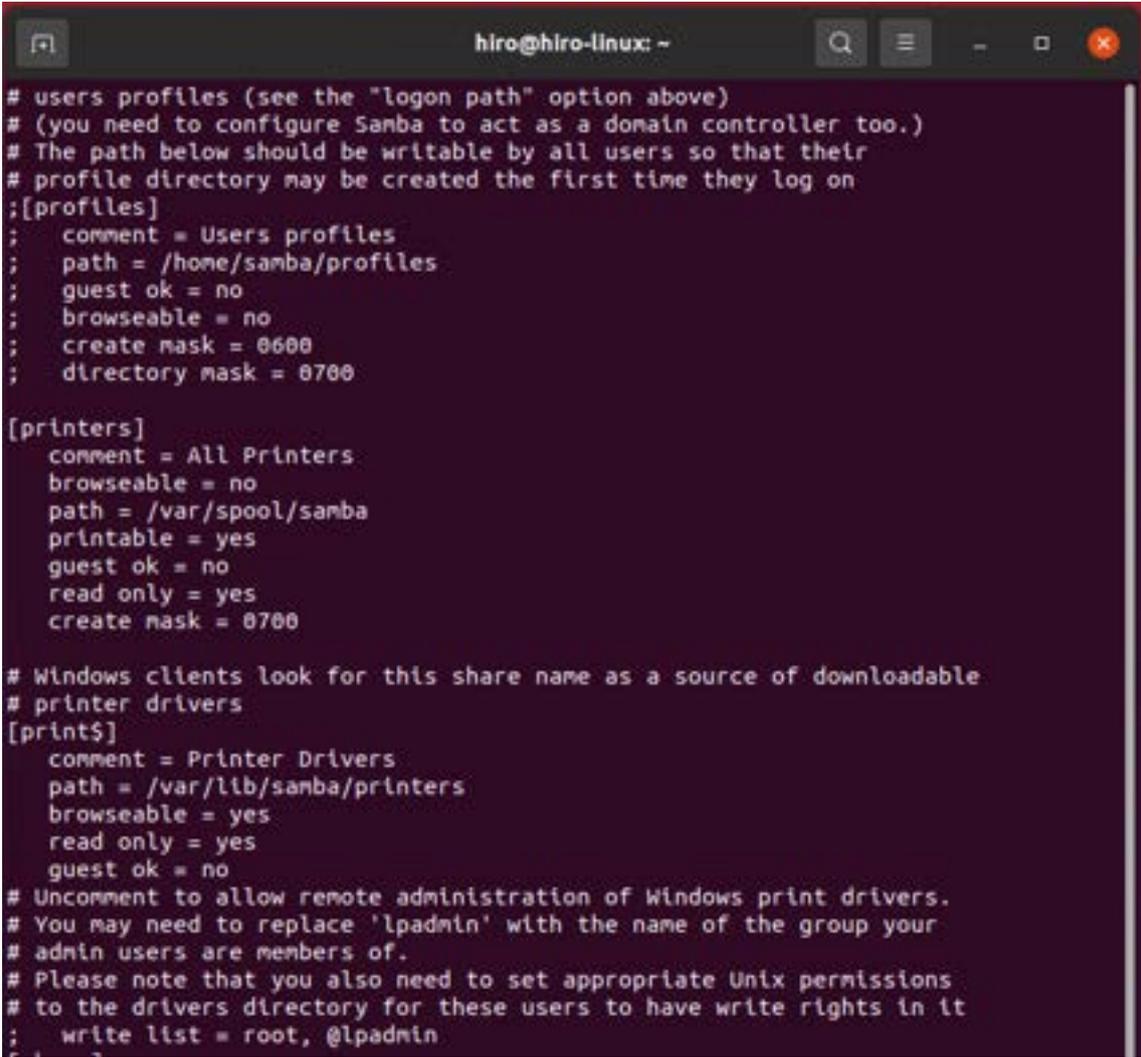
ファイルの設定が終わったので、サンバの詳細設定を行います。

Linux の標準的なテキストエディタ Vi (visual display editor)を使って詳細設定

Sudo vi /etc/Samba/smb.conf

入力すると、以下のような画面が出ます。もし出ない場合は、もう一度コマンドを入力してください。

Sudo vi /etc/Samba/smb.conf

A terminal window titled "hiro@hiro-linux: ~" showing the contents of the file /etc/Samba/smb.conf. The text is as follows:

```
# users profiles (see the "logon path" option above)
# (you need to configure Samba to act as a domain controller too.)
# The path below should be writable by all users so that their
# profile directory may be created the first time they log on
:[profiles]
;   comment = Users profiles
;   path = /home/samba/profiles
;   guest ok = no
;   browseable = no
;   create mask = 0600
;   directory mask = 0700

[printers]
comment = All Printers
browseable = no
path = /var/spool/samba
printable = yes
guest ok = no
read only = yes
create mask = 0700

# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
;   write list = root, @lpadmin
```

<vi について>

vi (visual display editor)は、Linux の標準的なテキストエディタです。vi はウィンドウベースではないため、各種の端末でさまざまなファイルの編集に使えます。

しかし、vi は、一般のテキストエディタのような手軽な方法で入力できないので少しコツがいります。

ファイルがすでに存在する場合、vi は既存のファイルを開きます。カーソルは、画面の左上隅に表示され、空白行は、チルダ(^)で示されます。

vi には、入力モードとコマンドモードという 2 種類の操作モードがあります。入力モードは、ファイルにテキストを入力するために使います。コマンドモードは、vi 特有の機能を実行するコマンドを入力するために使います。vi のデフォルトモードは、コマンドモードです。

ところが、vi はユーザーが現在どちらのモードにいるのかを示さないため、コマンドモードと入力モードの区別で混乱します。

vi を使って最初にファイルを開くと、常にコマンドモードになります。ファイルにテキストを入力するには、vi の入力コマンドを 1 つ入力します。入力コマンドには、現在のカーソル位置にテキストを挿入する「i(insert)」、現在のカーソル位置の後にテキストを追加する「a(append)」などがあります。

vi のコマンドモードに戻る場合は、Esc キーを押します。現在どちらのモードにいるのかよくわからない場合は、Esc キーを押してコマンドモードにした上で作業を続けます。vi がすでにコマンドモードのときに Esc キーを押すと、警告音が鳴ります。

「入力モード」

テキストを入力するには、vi の「挿入 (insert)」コマンドである「i」を入力します。このコマンドで入力モードに切り替わります。

ここで数行の短いテキストを入力し、各行の終わりで Return キーを押すと入力した文字はカーソルの左側に表示され、既存の文字は右側に押し出されます。入力途中ならば、Return キーを押す前に Back Space キーで後退して再入力することによって誤りを修正できます

テキストを入力し終わったら、Esc キーを押してコマンドモードに戻ります。このとき、カーソルは入力した最後の文字に移動します。これで次の vi のコマンドを入力できる状態になります。

「コマンドモード」

vi を使ってファイルを開くと、コマンドモードになります。このモードでは、広範囲にわたって機能を実行するさまざまなコマンドを入力できます。vi の大部分のコマンドは、1 文字か 2 文字と数値からなっています。

また、大文字のコマンドと小文字コマンドは異なる機能を持っています。たとえば、小文字の a はカーソルの右側にテキストを追加しますが、大文字の A はその行の最後にテキストを追加できます。

vi のコマンドの大部分は、Return キーを押さなくても実行されます。ただし、コロン (:) で始まるコマンドは、最後に Return キーを押す必要があります。

「ファイルの保存と vi の終了」

vi には、バッファ内容のファイル保存と vi の終了を制御する類似の意味を持つコマンドがあります。

「保存」

バッファの内容を保存する (バッファの内容をディスク上のファイルに書き込む) には、

:w

と入力して Return キーを押します。

「保存と終了」

:wq

と入力して Return キーを押します。

vi に以下の詳細設定を行います。

```
[share]      comment=Samba on Ubuntu
comm:      path=/home/share
path:      read only=no
read       writable=yes
brow:      browsable=yes
           create mask=0777
           directory mask=0777
```

カーソルが動かない場合は、コマンドモードになっているので、「ESC」キーを押して入力モードにしてください。
次に、「i」を押してカーソル位置に入力するインサートモードにしてください。

[share]をうちリターンキーを押します。
次に「tab」キーを押して

Comment=Samba on Ubuntu

リターンキーを押して、タブキーを押して

Path=/home/share

以下

```
read only =no
```

```
writable=yes
```

```
browsable=yes
```

```
create mask=0777
```

```
directory mask=0777
```

と打ちます。

途中カーソルが動かなくなったら「Esc」キーを押します。

文字が入力できなくなったら i を押します。

そして、打ち終えたら

```
:wq
```

と打って、エンターキーを押してください。

これで保存されます。

うまく編集ができない場合は、次の編集方法を試してください。

Linux の nano を使って詳細設定

Sudo nano /etc/Samba/smb.conf

```
hiro@hiro-linux:~$ sudo nano /etc/samba/smb.conf
```

以下の画面が表示されます。

```
GNU nano 4.8 /etc/samba/smb.conf
Sample configuration file for the Samba suite for Debian GNU/Linux.

This is the main Samba configuration file. You should read the
smb.conf(5) manual page in order to understand the options listed
here; Samba has a huge number of configurable options most of which
are not shown in this example

Some options that are often worth tuning have been included as
commented-out examples in this file.
- When such options are commented with ";*", the proposed setting
differs from the default Samba behaviour
- When commented with "#*", the proposed setting is the default
behaviour of Samba but the option is considered important
enough to be mentioned here

NOTE: Whenever you modify this file you should run the command
"testparm" to check that you have not made any basic syntactic
errors.

----- Global Settings -----

[global]

## Browsing/Identification ###

Change this to the workgroup/NT-domain name your Samba server will part of
workgroup = WORKGROUP

server string is the equivalent of the NT Description field
server string = %h server (Samba, Ubuntu)

### Networking ###

The specific set of interfaces / networks to bind to
This can be either the interface name or an IP address/netmask;
interface names are normally preferred
interfaces = 127.0.0.0/8 eth0

ヘルプ      C 書き込み      W 検索      ^M 切り取り      C 均等割付      C カーソル位置
終了      E 読み込み      A 置換      ^C 貼り付け      S スペル確認  O 行を指定
```

nano エディタは下のメニューに従ってキーを入力すると動作をするため、vi エディタのようにキーを覚えなくてもよいのが特徴です。



^G (F1) : ヘルプ

コントロールキーを押しながら G を押します。ファンクションキーの1でも同じ動作をします。

^X (F2) : 終了

コントロールキーを押しながら X を押します。F2 でも同じです。

^O (F3) : 書き込み

^J (F4) : テキスト整列

^R (F5) : 読み込み

^W (F6) : 検索

^Y (F7) : 前ページへ移動

^V (F8) : 次ページへ移動

^K (F9) : 1行カット

^U (F10) : 1行ペースト

^C (F11) : カーソル位置の確認

^T (F12) : スペルチェック

一番下の行に以下をうち、コントロールキー+Oで書き込み、名前がでたらそのままエンターキーを押して書き込み（保存）します。その後、コントロールキー+Xで終了します。

```
## /home/share ##  
  
[share]  
    comment=Samba on Ubuntu  
    path=/home/share  
    read only=no  
    writable=yes  
    browsable=yes  
    create mask=0777  
    directory mask=0777
```

Comment=Samba on Ubuntu

Path=/home/share

read only =no

writable=yes

browsable=yes

create mask=0777

directory mask=0777

と入力します。

入力ができたら、Samba を再起動します。

Sudo systemctl restart smbd

```
hiro@hiro-linux:~$ sudo systemctl restart smbd
```

Sudo systemctl restart nmbd

```
hiro@hiro-linux:~$ sudo systemctl restart nmbd
```

ユーザーの追加

ファイルにアクセスできるユーザーを追加します。

Samba にアクセスするためのユーザーを登録します。Samba のユーザーは、Ubuntu Server のユーザーでなくてはなりません。まずは、次のコマンドで Ubuntu Server にユーザーを登録します。Ubuntu Server のユーザー登録ができれば、最後に Samba のユーザー登録をします。

Sudo adduser --disabled-login animal

※ユーザー名を animal にしています。

```
hiro@hiro-linux:~$ sudo adduser --disabled-login animal
```

ログインしなくてよいので「-disabled-login」を指定しています。

```
hiro@hiro-linux:~$ sudo adduser --disabled-login animal
ユーザー 'animal' を追加しています...
新しいグループ 'animal' (1001) を追加しています...
新しいユーザー 'animal' (1001) をグループ 'animal' に追加しています...
ホームディレクトリ '/home/animal' を作成しています...
'/etc/skel' からファイルをコピーしています...
animal のユーザ情報を変更中
新しい値を入力してください。標準設定値を使うならリターンを押してください
フルネーム []: 
```

```
新しい値を入力してください。標準設定値を使うならリターンを押してください
フルネーム []: haru aki
部屋番号 []:
職場電話番号 []:
自宅電話番号 []:
その他 []:
以上で正しいですか? [Y/n] y□
```

※ここでは、「Full Name」（氏名）を、haru aki にしています。

「Room Number」（部屋番号）や電話番号など聞かれますが、何も書かずエンターキーを打ちます。

「以上で正しいですか? [Y/N] 」が表示されたら、yを押して [Enter] キーを押します。

最後に Samba のユーザーに登録します。

```
hiro@hiro-linux:~$ sudo pdbedit -a animal
```

Sudo pdbedit -a animal

※Ubuntu のユーザー名 animal を入力します。

パスワードを求められるので設定します。(例 1234)

```
hiro@hiro-linux:~$ sudo pdbedit -a animal
new password:
retype new password:
Unix username:      animal
NT username:
Account Flags:      [U          ]
User SID:           S-1-5-21-2615385677-1980255381-2131549654-1001
Primary Group SID: S-1-5-21-2615385677-1980255381-2131549654-513
Full Name:          haru aki
Home Directory:     \\HIRO-LINUX\animal
HomeDir Drive:
Logon Script:
Profile Path:       \\HIRO-LINUX\animal\profile
Domain:             HIRO-LINUX
Account desc:
Workstations:
Munged dial:
Logon time:         0
Logoff time:        木, 07 2月 2036 00:06:39 JST
Kickoff time:       木, 07 2月 2036 00:06:39 JST
Password last set:  火, 07 2月 2023 18:57:03 JST
Password can change: 火, 07 2月 2023 18:57:03 JST
Password must change: never
Last bad password   : 0
Bad password count  : 0
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
hiro@hiro-linux:~$
```

これで、ファイルサーバができました。

今回設定したファイルサーバはウィンドウズでもアンドロイドでもマック OS でも Linux でもファイルの共有ができるので非常に便利です。

Ubuntu のアップグレード

Ubuntu は定期的にバージョンアップをしています。同じバージョンは5年間はサポートしています。新しいバージョンになると膨大な数のパッケージの中には動作が不安定になるものがあります。バグ修正などが頻繁に行われるので、定期的にアップグレードをする必要があります。

今のバージョンを確認します。

`cat /etc/os-release`

```
hiro@hiro-linux:~$ cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.1 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.1 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
hiro@hiro-linux:~$
```

アップグレード可能なバージョンを確認します。

```
sudo do-release-upgrade -c
```

```
hiro@hiro-linux:~$ sudo do-release-upgrade -c
[sudo] hiro のパスワード:
新しい Ubuntu のリリースをチェックしています
入手可能な LTS の開発版はありません。
最新の非 LTS 開発リリースにアップグレードする
/etc/update-manager/release-upgrades に Prompt=normal とセットしてください。
```

アップグレードする前にインストールされているパッケージをアップデートしてくださいの内容が出た場合

利用可能なパッケージ一覧を更新します。

```
sudo apt update
```

```
hiro@hiro-linux:~$ sudo apt update
取得:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
ヒット:2 http://jp.archive.ubuntu.com/ubuntu jammy InRelease
ヒット:3 http://jp.archive.ubuntu.com/ubuntu jammy-updates InRelease
ヒット:4 http://jp.archive.ubuntu.com/ubuntu jammy-backports InRelease
取得:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [41.6 kB]
取得:6 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [13.3 kB]
165 kB を 2秒 で取得しました (71.9 kB/s)
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています... 完了
状態情報を読み取っています... 完了
パッケージはすべて最新です。
```

パッケージを更新します。

sudo apt upgrade

```
hiro@hiro-linux:~$ sudo apt upgrade
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています... 完了
状態情報を読み取っています... 完了
アップグレードパッケージを検出しています... 完了
以下のパッケージが自動でインストールされましたが、もう必要とされていません:
  app-install-data-partner bsdmainutils cpp-9 gcc-10-base gcc-9-base
  gir1.2-clutter-1.0 gir1.2-clutter-gst-3.0 gir1.2-cogl-1.0
  gir1.2-cogl-pango-1.0 gir1.2-gnomebluetooth-1.0 gir1.2-gtkclutter-1.0
  gnome-screenshot ippusbxd libamtk-5-0 libamtk-5-common libasn1-8-heimdal
  libboost-date-time1.71.0 libboost-filesystem1.71.0 libboost-iostreams1.71.0
  libboost-locale1.71.0 libboost-thread1.71.0 libbrlapi0.7 libcamel-1.2-62
  libcbor0.6 libcc1-0 libcdio18 libcmis-0.5-5v5 libdns-export1109 libdpkg-perl
  libedataserver-1.2-24 libedataserverui-1.2-2 libextutils-pkgconfig-perl
  libfile-fcntllock-perl libfuse2 libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.0-0
  libgssapi3-heimdal libgupnp-1.2-0 libhandy-0.0-0 libhcrypto4-heimdal
  libheimbase1-heimdal libheimntlm0-heimdal libhogweed5 libhx509-5-heimdal
  libicu66 libidn11 libisl22 libjson-c4 libjuh-java libjurt-java
  libkrb5-26-heimdal libldap-2.4-2 liblibreoffice-java libllvmlib2 libmozjs-68-0
  libmpdec2 libmysqlclient21 libneon27-gnutls libnettle7 libntfs-3g883
  liborcus-0.15-0 libperl5.30 libphonenumber7 libpoppler97 libprotobuf17
  libpython3.8 libpython3.8-minimal libpython3.8-stdlib libqpdf26 libraw19
  libreoffice-style-tango librid1-java libroken18-heimdal libsane libsnmp35
```

必要がなくなったパッケージを削除します。

```
sudo apt autoremove -y
```

```
gnome-screenshot ippusbxd libatk-5-0 libatk-5-common libasni-8-heimdal
libboost-date-time1.71.0 libboost-filesystem1.71.0 libboost-iostreams1.71.0
libboost-locale1.71.0 libboost-thread1.71.0 libbriapi0.7 libcamel-1.2-62
libcbor0.6 libcc1-0 libcdio18 libcmis-0.5-5v5 libdns-export1109 libdpkg-perl
libdataserver-1.2-24 libdataserverui-1.2-2 libextutils-pkgconfig-perl
libfile-fcntllock-perl libfuse2 libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.0-0
libgssapi3-heimdal libgupnp-1.2-0 libhandy-0.0-0 libhcrypto4-heimdal
libheibase1-heimdal libheimntlm0-heimdal libhogweed5 libhx509-5-heimdal
libicu66 libidn11 libisl22 libjson-c4 libjuh-java libjurt-java
libkrb5-26-heimdal libldap-2.4-2 liblibreoffice-java libllvm12 libmozjs-68-0
libmpdec2 libmysqlclient21 libneon27-gnutls libnettle7 libntfs-3g883
liborcus-0.15-0 libperl5.30 libphonenumbers7 libpoppler97 libprotobuf17
libpython3.8 libpython3.8-minimal libpython3.8-stdlib libqpdf26 libraw19
libreoffice-style-tango libridl-java libroken18-heimdal libsane libsnmp35
libtepl-4-0 libtracker-control-2.0-0 libtracker-miner-2.0-0
libtracker-sparql-2.0-0 libunoloader-java libvpx6 libwebp6 libwind0-heimdal
libwmf0.2-7 libzinnia0 linux-hwe-5.15-headers-5.15.0-58 ltrace lz4
mysql-common ncal perl-modules-5.30 pkg-config popularity-contest
python3-entrypoints python3-packaging python3-requests-unixsocket
python3-simplejson python3.8 python3.8-minimal syslinux syslinux-common
syslinux-legacy tegaki-zinnia-japanese ure-java vino xul-ext-ubufox
アップグレード: 0 個、新規インストール: 0 個、削除: 106 個、保留: 0 個。
この操作後に 412 MB のディスク容量が解放されます。
続行しますか? [Y/n]
```

```
python3-entrypoints (0.4-1) を削除しています ...
python3-packaging (21.3-1) を削除しています ...
python3-requests-unixsocket (0.2.0-2) を削除しています ...
python3-simplejson (3.17.6-1build1) を削除しています ...
syslinux (3:6.04-git20190206.bf6db5b4+dfsg1-3ubuntu1) を削除しています ...
syslinux-common (3:6.04-git20190206.bf6db5b4+dfsg1-3ubuntu1) を削除しています ..
:
syslinux-legacy (2:3.63+dfsg-2ubuntu9) を削除しています ...
tegaki-zinnia-japanese (0.3-2) を削除しています ...
vino (3.22.0-6ubuntu3) を削除しています ...
xul-ext-ubufox (3.4-0ubuntu1.17.10.1) を削除しています ...
libasni-8-heimdal:amd64 (7.7.0+dfsg-3ubuntu1) を削除しています ...
libroken18-heimdal:amd64 (7.7.0+dfsg-3ubuntu1) を削除しています ...
hicolor-icon-theme (0.17-2) のトリガを処理しています ...
gnome-menus (3.36.0-1ubuntu3) のトリガを処理しています ...
libglib2.0-0:amd64 (2.72.4-0ubuntu1) のトリガを処理しています ...
libe-bin (2.35-0ubuntu3.1) のトリガを処理しています ...
man-db (2.10.2-1) のトリガを処理しています ...
libreoffice-common (1:7.3.7-0ubuntu0.22.04.1) のトリガを処理しています ...
mailcap (3.70+nmu1ubuntu1) のトリガを処理しています ...
desktop-file-utils (0.26-1ubuntu3) のトリガを処理しています ...
hiro@hiro-linux:~$
```

再起動します。

`sudo reboot`

```
hiro@hiro-linux:~$ sudo reboot
```

アップグレードします。

`sudo do-release-upgrade`

この作業は、1時間程度かかります。途中何度かストップをして続けるかどうかを聞いてきますので、`y` を押してエンターキーを押して続けてください。

第3章 Web サーバの構築

サーバとは

サーバ(server)とは、サービス(service)をするものという意味です。どんなサービスがあるかということ、メールを送受信するもの、Webサイトを閲覧するもの、データを保存したり検索するものなど多くの種類があります。

サーバの種類によってインストールするソフトも違います。主なサーバとソフトと特徴を表にします。

サーバ	ソフト例	特徴
メール	Sendmail,plstfix	SMTP、POP,IMAP4 サーバなど
Web	Apache,nginx	Webサイトの機能
DNS	Bind	DNS機能
DHCP	Dhcpd	IPアドレスを自動的に振る
FTP	Proftpd,iis	FTPプロトコルでファイル送受信
プロキシ	Squid,nginx	通信の中継
データベース	Mysql,postgresql	データベース
ファイル	Samba	ファイル共有
認証	openldap	ユーザー認証

ここでは、webサーバソフトとしてApacheを使います。Apacheは、無料で処理速度が速い標準的なソフトです。

Apache のインストール

パッケージのインストール先の PPA (personal package archive) を Linux の倉庫 (リポジトリ) に追加します。

```
sudo add-apt-repository ppa:ondrej/apache2
```

```
hiro@hiro-linux:~$ sudo add-apt-repository ppa:ondrej/apache2
PPA publishes dbgshim, you may need to include 'main/debug' component
リポジトリ: 'deb https://ppa.launchpadcontent.net/ondrej/apache2/ubuntu/ jammy m
ain'
概要:
This branch follows latest Apache2 packages as maintained by the Debian Apache2
team with couple of compatibility patches on top.

It also includes some widely used Apache 2 modules (if you need some other feel
free to send me a request).

IMPORTANT: The <foo>-backports is now required on older Ubuntu releases.

BUGS&FEATURES: This PPA now has a issue tracker: https://deb.sury.org/#bug-repor
ting

PLEASE READ: If you like my work and want to give me a little motivation, please
consider donating: https://deb.sury.org/#donate
より詳しい情報: https://launchpad.net/~ondrej/+archive/ubuntu/apache2
リポジトリを追加しています。
続けるには「Enter」キーを、中止するにはCtrl-cを押してください。 █
```

エンターキーを押して進めてください。

```
続けるには「Enter」キーを、中止するにはCtrl-cを押してください。
Adding deb entry to /etc/apt/sources.list.d/ondrej-ubuntu-apache2-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ondrej-ubuntu-apache2-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ondrej-ubuntu-apache2.gpg with fingerprint
14AA40EC0831756756D7F66C4F4EA0AAE5267A6C
ヒット:1 http://jp.archive.ubuntu.com/ubuntu jammy InRelease
取得:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
取得:3 http://jp.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
取得:4 https://ppa.launchpadcontent.net/ondrej/apache2/ubuntu jammy InRelease [2
3.8 kB]
取得:5 http://jp.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
取得:6 http://jp.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packag
es [8,600 B]
取得:7 http://jp.archive.ubuntu.com/ubuntu jammy-updates/multiverse i386 Package
s [2,360 B]
取得:8 http://jp.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-
en [2,720 B]
取得:9 https://ppa.launchpadcontent.net/ondrej/apache2/ubuntu jammy/main i386 Pa
ckages [2,696 B]
取得:10 https://ppa.launchpadcontent.net/ondrej/apache2/ubuntu jammy/main amd64
Packages [2,700 B]
取得:11 https://ppa.launchpadcontent.net/ondrej/apache2/ubuntu jammy/main Transl
ation-en [2,220 B]
376 kB を 6秒 で取得しました (61.2 kB/s)
パッケージリストを読み込んでいます... 完了
hiro@hiro-linux:~$
```

追加ができれば、パッケージ情報を確認します。

sudo apt show apache2

```
hiro@hiro-linux:~$ sudo apt show apache2
[sudo] hiro のパスワード:
Package: apache2
Version: 2.4.55-1+ubuntu22.04.1+deb.sury.org+2
Priority: optional
Section: httpd
Maintainer: Debian Apache Maintainers <debian-apache@lists.debian.org>
Installed-Size: 524 kB
Provides: httpd, httpd-cgi
Depends: apache2-bin (= 2.4.55-1+ubuntu22.04.1+deb.sury.org+2), apache2-data (= 2.4.55-1+ubuntu22.04.1+deb.sury.org+2), apache2-utils (= 2.4.55-1+ubuntu22.04.1+deb.sury.org+2), lib-base, mime-support | media-types, perl:any, procps
Recommends: ssl-cert
Suggests: apache2-doc, apache2-suexec-pristine | apache2-suexec-custom, www-browser
Download-Size: 165 kB
APT-Manual-Installed: yes
APT-Sources: https://ppa.launchpadcontent.net/ondrej/apache2/ubuntu jammy/main amd64 Packages
Description: Apache HTTP サーバ
 Apache HTTP Server Project は、標準準拠のオープンソースソフトウェアとして、安全で、効率的で、拡張性のある HTTP サーバを構築することを目的としています。その成果は、インターネットで長い間ナンバーワンの web サーバとなっています。

 このパッケージをインストールすれば、設定ファイル、init スクリプト、補助スクリプトを含むすべてをインストールできます。

注: 追加レコードが 2 件あります。表示するには '-a' スイッチを付けてください。
hiro@hiro-linux:~$
```

2023年2月8日現在、versionが2.4.55になっています。これは、ubuntu22.04.1上でのバージョンになっています。もし、バージョンが低ければ、ubuntuのアップグレードをする必要があります。ubuntuのアップグレードのページを参照してアップグレードをして最初からやり直してください。

Apache をインストールします。

sudo apt-get update

```
hiro@hiro-linux:~$ sudo apt-get update
ヒット:1 http://jp.archive.ubuntu.com/ubuntu jammy InRelease
取得:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
ヒット:3 http://jp.archive.ubuntu.com/ubuntu jammy-updates InRelease
取得:4 http://jp.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
ヒット:5 https://ppa.launchpadcontent.net/ondrej/apache2/ubuntu jammy InRelease
217 kB を 2秒 で取得しました (90.4 kB/s)
パッケージリストを読み込んでいます... 完了
hiro@hiro-linux:~$
```

続いて、

sudo apt-get install -y apache2

```
hiro@hiro-linux:~$ sudo apt-get install -y apache2
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています... 完了
状態情報を読み取っています... 完了
以下の追加パッケージがインストールされます:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
提案パッケージ:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
以下のパッケージが新たにインストールされます:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
アップグレード: 0 個、新規インストール: 8 個、削除: 0 個、保留: 0 個。
2,131 kB のアーカイブを取得する必要があります。
この操作後に追加で 7,673 kB のディスク容量が消費されます。
取得:1 http://jp.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4 [92.4 kB]
取得:2 https://ppa.launchpadcontent.net/ondrej/apache2/ubuntu jammy/main amd64 libapr1 amd64 1.7.0-8+ubuntu22.04.1+deb.sury.org+1 [115 kB]
取得:3 http://jp.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4 [11.3 kB]
取得:4 http://jp.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4 [9,162 B]
取得:5 https://ppa.launchpadcontent.net/ondrej/apache2/ubuntu jammy/main amd64 apache2-bin amd64 2.4.55-1+ubuntu22.04.1+deb.sury.org+2 [1,416 kB]
44% [5 apache2-bin 688 kB/1,416 kB 49%] 68.2 kB/s 17秒
```

インストールが出来たら、インストールが成功し動作しているかを確認します。

sudo systemctl status apache2

```
hiro@hiro-linux:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Wed 2023-02-08 12:10:41 JST; 46s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 5377 (apache2)
    Tasks: 55 (limit: 9132)
   Memory: 5.2M
     CPU: 42ms
   CGroup: /system.slice/apache2.service
           └─5377 /usr/sbin/apache2 -k start
             └─5378 /usr/sbin/apache2 -k start
               └─5379 /usr/sbin/apache2 -k start

2月 08 12:10:41 hiro-linux systemd[1]: Starting The Apache HTTP Server...
2月 08 12:10:41 hiro-linux apachectl[5376]: AH00558: apache2: Could not reliably
2月 08 12:10:41 hiro-linux systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

緑色で active(running)とあるので、正常に動作していることが確認できました。

左下の「lines 1-16/16(END)」は「Q」キーを押すと消えます。

Apache のコマンド

以下に主に使うコマンドを上げます。シェルで動作を試してみてください。

```
systemctl start apache2
```

Apache のデーモンを起動します。

```
systemctl stop apache2
```

Apache のデーモンを停止します。

```
systemctl restart apache2
```

Apache デーモンを再起動します。デーモンが実行されていない場合、起動します。

```
systemctl reload apache2
```

設定ファイルを再読み込みします。

```
systemctl status apache2
```

ステータスを表示します。

設定ファイル

Apache の主な設定ファイルを上げておきます。それぞれのファイルの内容を見て、詳細設定をします。

場所は/etc/apache2 にあります。

apache2.conf

Apache サーバのメインの設定ファイルです。

envvars

環境設定のファイルです。

magic

ファイルの最初の数バイトの MIME タイプを決定します。

ports.conf

Apache が TCP ポートを決定する指示を格納しています。

Web サーバ動作確認

サーバとして動作しているかを確認するために、ブラウザからサーバーに接続してみます。

まず、ホストの IP アドレスを調べます。IP アドレスとはネットワーク上にある機器の住所です。

hostname -I

```
hiro@hiro-linux:~$ hostname -I
192.168.2.102 2001:a253:2941:f100:60dd:f6d0:7506:c1f4 2001:a253:2941:f100:199c:f561:3448:f3f5
```

この Linux マシンの住所（IP アドレス）は、192.168.2.102 になります。

Linux に入っている Firefox を立ち上げて、

<http://192.168.2.102/>

を入力してください。Apache のページが表示されます。



以下のファイルが表示され、ウェブサーバーとして機能していることが確認できます。



ポート・IP アドレス

ファイアウォールの設定をする前に簡単に用語の説明をします。

「ポート」

IP アドレスにある PC の、どのソフトで使用するかを指定するものです。
HTTP は 80 番ポート、HTTPS は 443 番ポートです。

「IPv4」

アドレスを 32 ビットのデータで表し約 43 億個あります。
「255.255.255.255」等で表されるもの。(2 の 8 乗) × 4 で 2 の 32 乗個になります。

「IPv6」

アドレスを 128 ビットのデータで表し約 2 の 128 乗個あります。320 兆 × 1 兆 × 1 兆個です。(3.4 × 10 の 38 乗)
「2001:0db8:bd05:01d3:355a:1fe0:0001:1eee」(16 進法で表す)

ファイアウォール設定

ファイアウォールのインストールを確認して有効にします。
そして、ポートを開いてファイアウォールの動作確認をします。

念のためファイアウォールをインストールします

sudo apt install ufw

```
hiro@hiro-linux:~$ sudo apt install ufw
```

ファイアウォールを有効にします。

sudo ufw enable

```
hiro@hiro-linux:~$ sudo ufw enable  
ファイアウォールはアクティブかつシステムの起動時に有効化されます。  
hiro@hiro-linux:~$ sudo ufw allow 80/tcp
```

ufw について

ufw は一度有効化すると、システム起動時に自動的に有効になります。

ufw の操作は、ターミナルから行います。[アプリケーション] → [アクセサリ] → [端末] を開いて操作を行ってください。ufw は基本的に、「デフォルトではあらゆる外部からの通信を拒否し、許可する通信を定義していく」形で設定を行います。（「sudo ufw default ALLOW」を実行することで、「デフォルトでは全て通すが、不要な通信のみを落とす」設定にすることも可能です。）

Windows 環境向けの統合セキュリティソフトが提供する「ソフトウェアファイアウォール」の多くは、「内側から外側に行く通信」（アウトバウンド）と「外側から内側に入ってくる通信」（インバウンド）の双方を制御することができますが、（今のところ）ufw はあくまでインバウンド通信の制御に特化しています。

ufw enable を実行して全ての通信を遮断したら、「必要なポートを開く」作業を行います。

SSH や IMAP、HTTPS の BASIC 認証などの、パスワードによる認証を用いるサービスでは、「でたらめなパスワードを打ち込み続け、偶然正しいものと一致することを期待する」(ブルートフォース) 攻撃への対策が必要です。

SSH では公開鍵による認証を行うことで、パスワードよりも強固な、事実上ブルートフォース攻撃では破れない設定にすることも可能ですが、鍵の管理が面倒など、パスワード認証に比べて不便な点もあり、すべての場合で利用できるわけでもありません。

すでに ufw と iptables の比較のところでも触れたように、このような場合は ufw の「接続頻度を制限する」機能が便利です。この指定は LIMIT キーワードを用いて行います。以下のように操作しておくことで、「執拗に接続してくるアクセス」を拒否することが可能です。

ポート開放

Apache のポートを開きます。

```
sudo ufw allow 80/tcp
```

```
hiro@hiro-linux:~$ sudo ufw allow 80/tcp  
ルールを追加しました  
ルールを追加しました (v6)
```

ssh のポートを開きます。

```
sudo ufw allow 22/tcp
```

```
hiro@hiro-linux:~$ sudo ufw allow 22/tcp  
ルールを追加しました  
ルールを追加しました (v6)
```

ファイアウォールを再読み込みします。

`sudo ufw reload`

```
hiro@hiro-linux:~$ sudo ufw reload
ファイアウォールを再読み込みしました
```

ファイアウォールの動作確認をします。

`sudo ufw status`

```
hiro@hiro-linux:~$ sudo ufw status
状態: アクティブ

To Action From
--
80/tcp ALLOW Anywhere
22/tcp ALLOW Anywhere
80/tcp (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)
```

HTTP サーバの動作確認

これでポートの開放とファイアウォールの設定が終わりました。サーバとして最低限必要な設定は終わりました。

実際にページを作って HTTP サーバとして動かしてみましよう。

ページを書き込むディレクトリに移動します。

```
cd /var/www/html/
```

```
hiro@hiro-linux:~$ cd /var/www/html/
```

`cd`はカレントディレクトリです。指定したディレクトリに移動します。

nano エディタを起動して記述をし、`test.html` という名前のファイルを作ります。

```
sudo nano test.html
```

```
hiro@hiro-linux:/var/www/html$ sudo nano test.html
```

エディタが開くので、次のソースを書いてください。

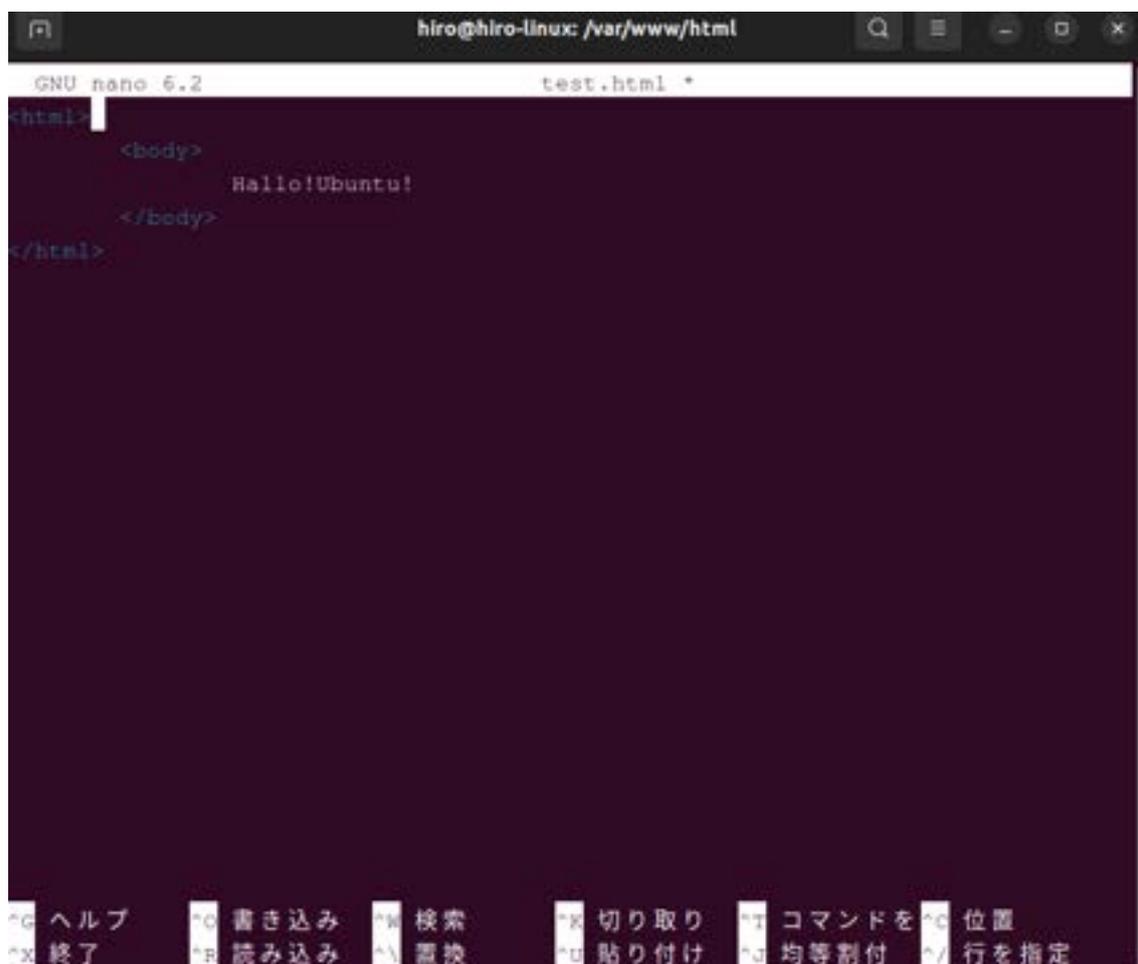
<HTML>

<body>

Hallo!Ubuntu!

</body>

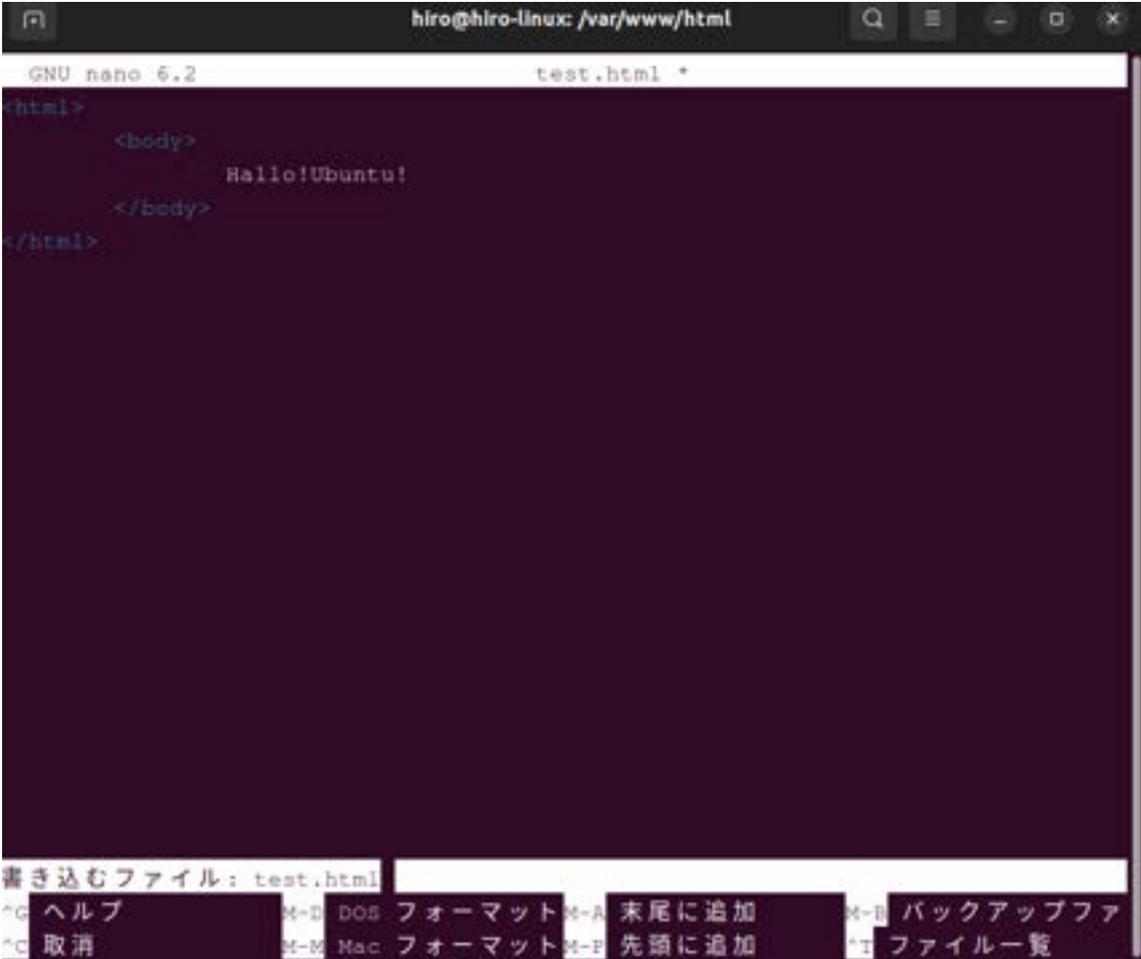
</HTML>



```
hiro@hiro-linux: /var/www/html
GNU nano 6.2 test.html *
<html>
  <body>
    Hallo!Ubuntu!
  </body>
</html>
```

ヘルプ 書き込み 検索 切り取り コマンドを 位置
終了 読み込み 置換 貼り付け 均等割付 行を指定

書いたら、コントロールキー+「O」を押します。



The screenshot shows a terminal window with the nano text editor open. The editor is editing a file named 'test.html'. The content of the file is as follows:

```
<html>
  <body>
    Hallo!Ubuntu!
  </body>
</html>
```

At the bottom of the terminal, there is a status bar with the following text:

書き込むファイル: test.html

ヘルプ	M-B	DOS	フォーマット	M-A	末尾に追加	M-B	バックアップファ
取消	M-M	Mac	フォーマット	M-E	先頭に追加	M-T	ファイル一覧

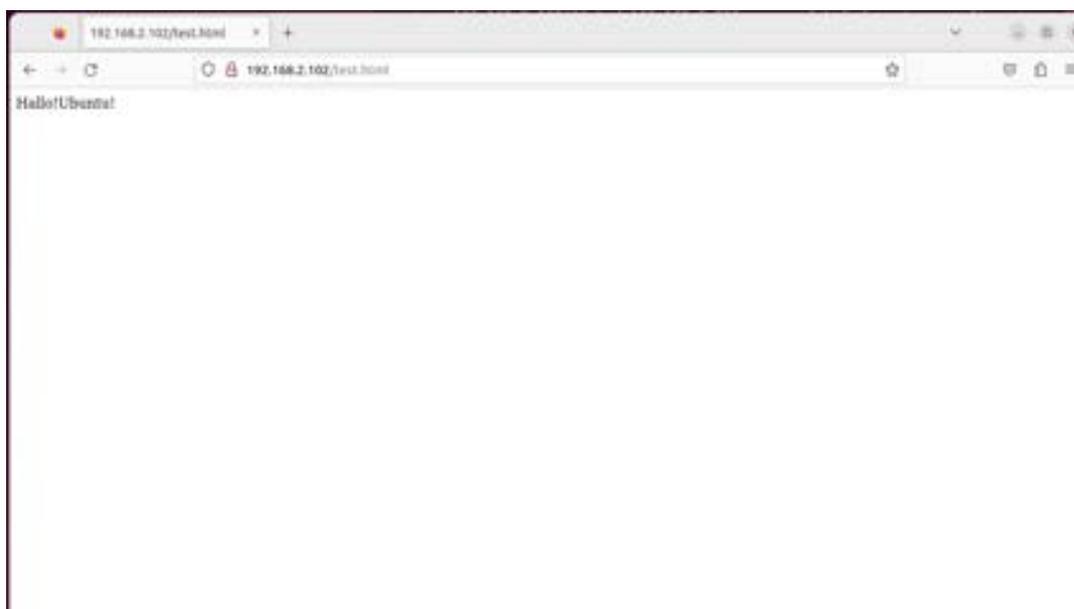
書き込むファイルを確認してくるので、エンターキーを押します。

ブラウザで確認します。

Firefox を起動して、

<http://192.168.2.102/test.html>

を入力してページを表示させます。



これで、web サーバが完成しました。wifi を接続して格納されたデータを閲覧することができます。

HTML ディレクトリ

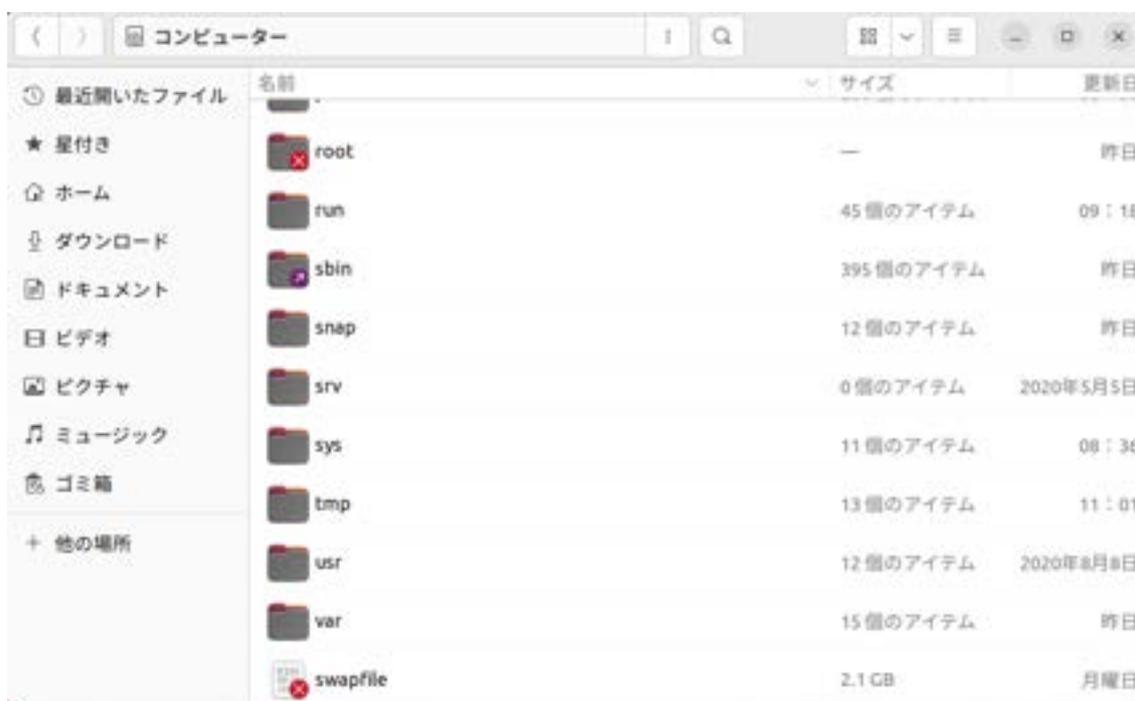
ホームをクリックします。



他の場所をクリックします



一番下の「var」ディレクトリをクリックします



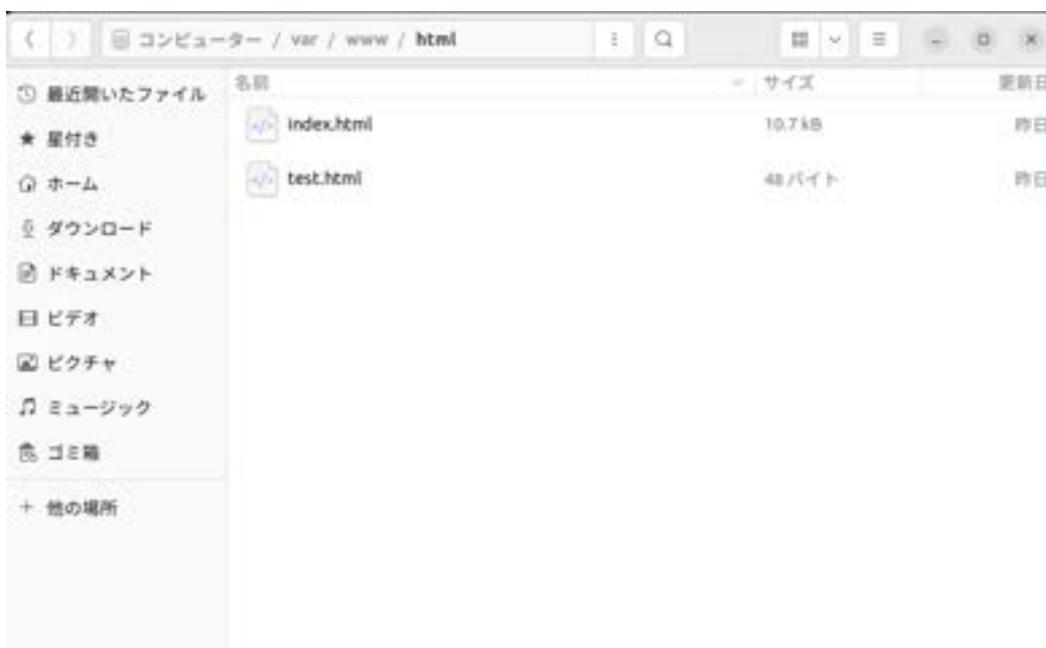
一番下の「www」をクリックします



「html」ディレクトリがあります。この中にhtmlファイルが入っています。

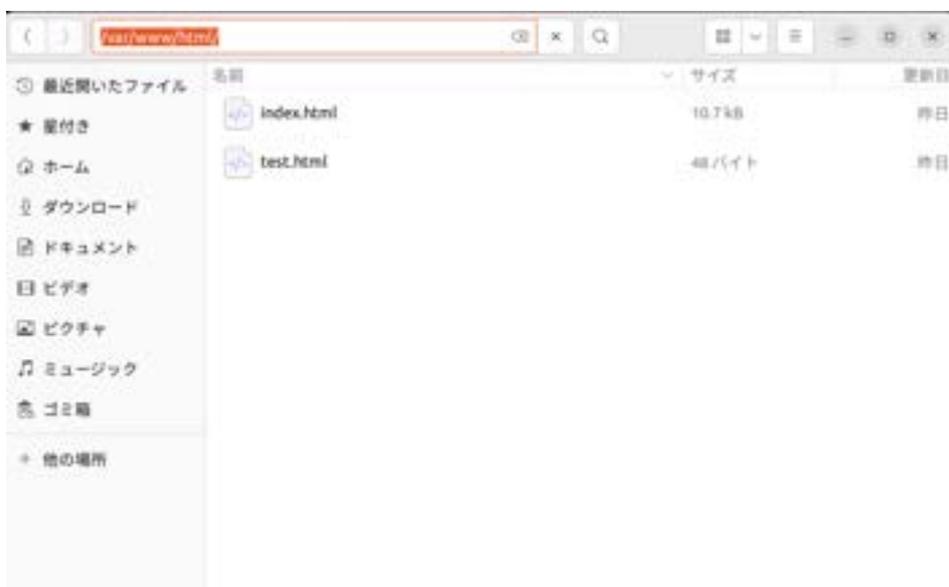


「html」ディレクトリをクリックします。

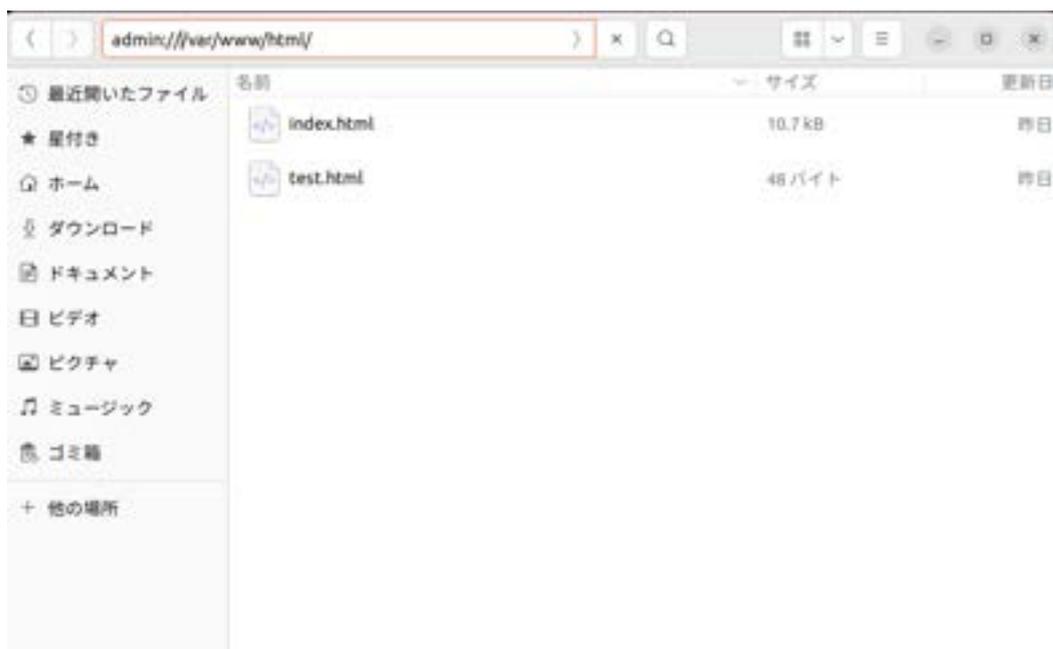


このディレクトリに画像を移動します。しかし、このままでは管理者として開いていないので、ディレクトリの操作はできません。

このディレクトリが表示されているとき、「コントロールキー」+「L」のキーを同時に押します。そうすると、上の窓にパスが表示されます。「varの中のwwwの中のhtmlの中」という意味です。



ここで、パスの前にカーソルを持って行って、
admin://
と入力します。



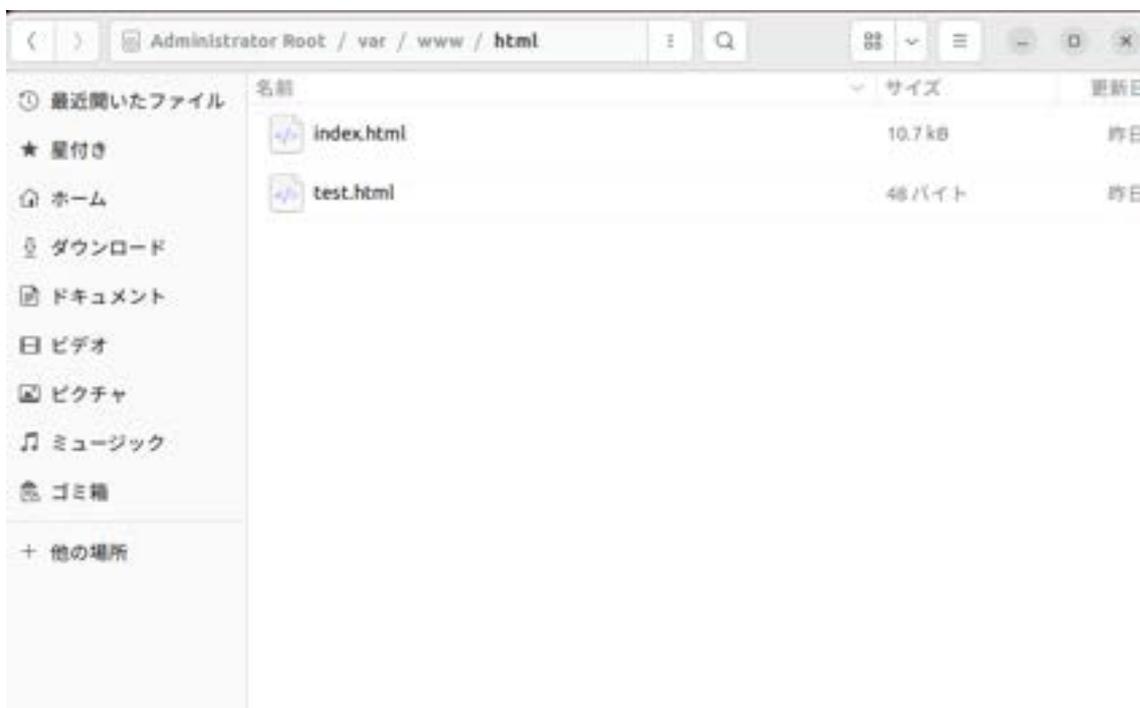
admin:///var/www/html/

※スラッシュが3本になっていることに注意してください。

確認できたら、エンターキーを押してください。



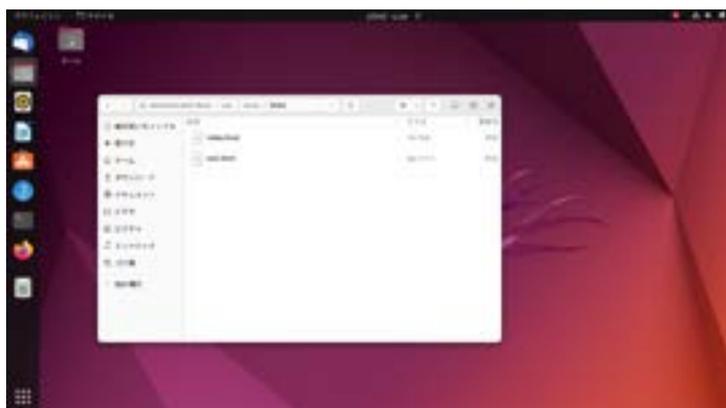
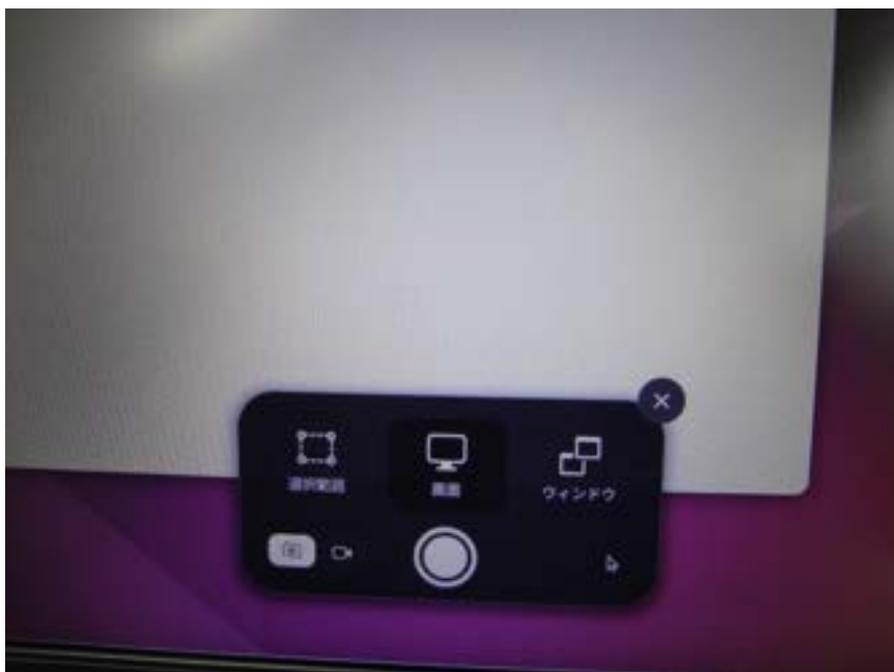
パスワードが求められますので、パソコンにログインするときのパスワードを入力してください。



これで管理者としての権限が反映されました。さて、ここにスクリーンショットの画像を移動して、FireFox で表示させてみましょう。

キーボードの右上に「PRT SCR」キーがあるので押してください。
※PRT SCR (Print Screen)。画面や範囲指定した範囲の画像を保存します。

このように表示されたら、中央上の「画面」を選んで、中央下の丸いボタンをクリックします。これで画面のスクリーンショットが保存されました。



スクリーンショットの保存場所を開いて、コピーします。

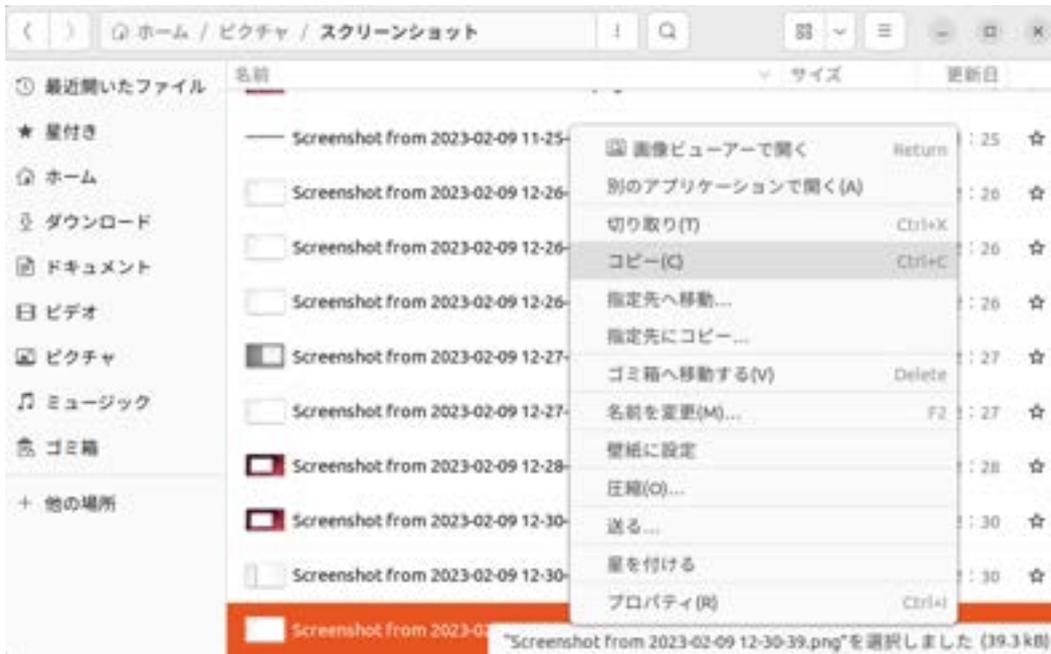
デスクトップのファイルをクリックします。下の画面になったら、ピクチャを開いてください。



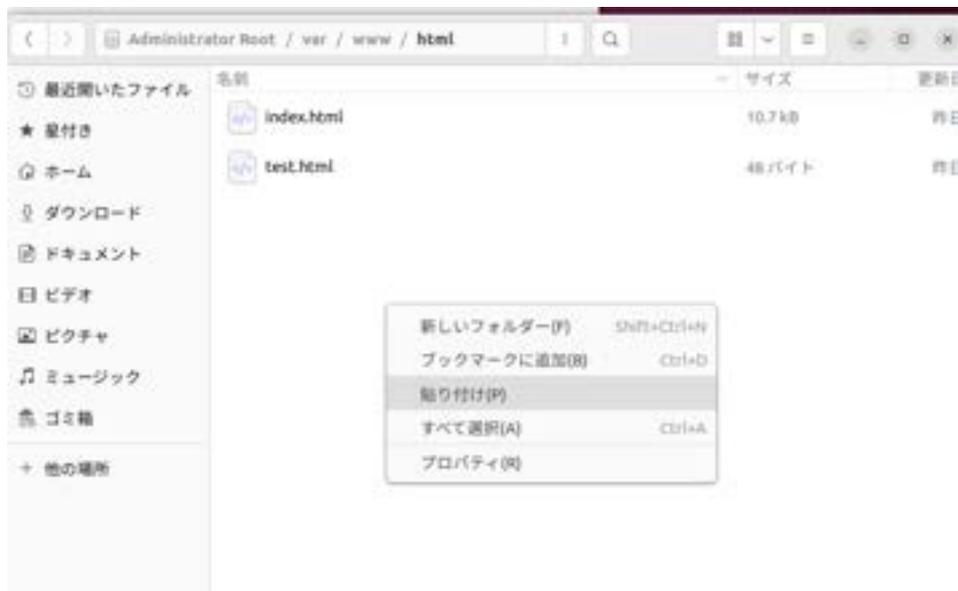
この中にスクリーンショットがあるので開きます。



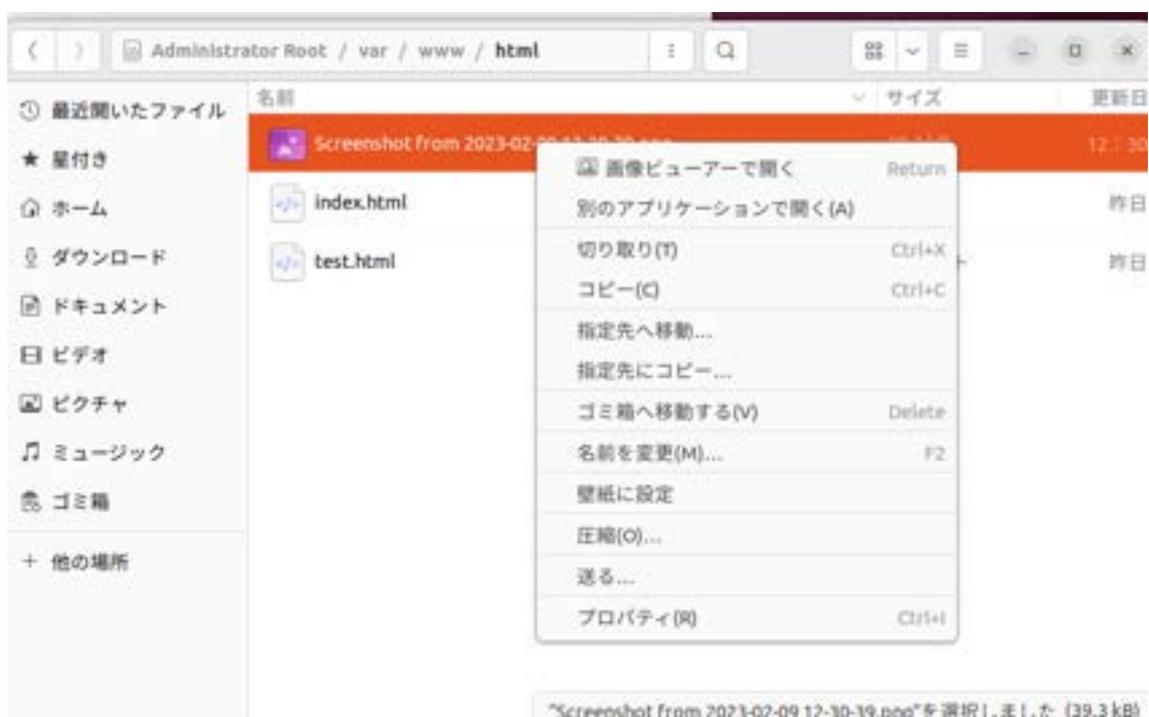
保存されているスクリーンショットを右クリックして、コピーします。



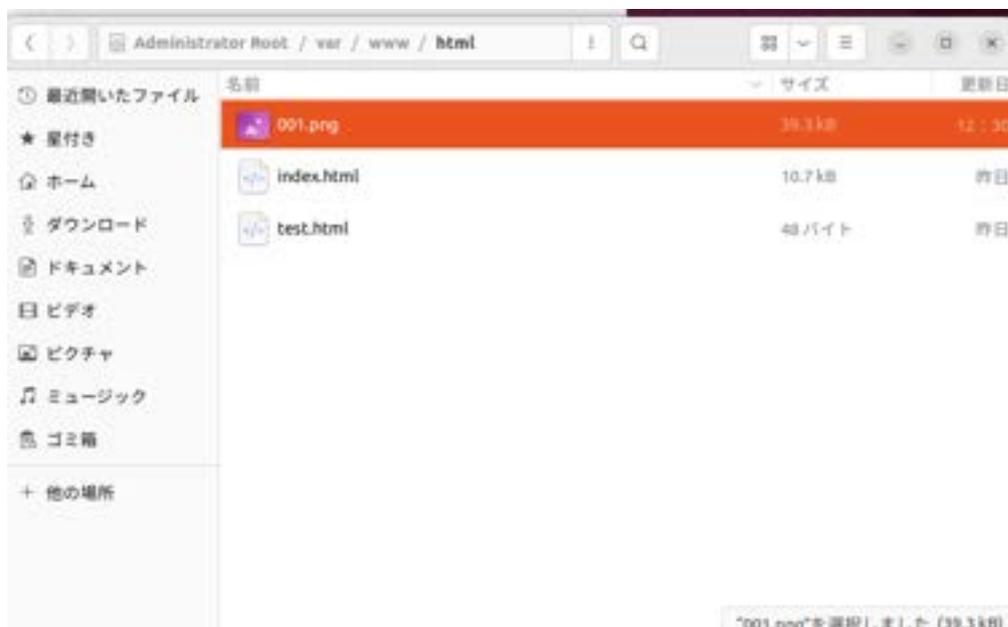
コピーしたファイルを先ほどの html の中に貼り付けます。



コピーしたファイルを右クリックして、名前を変更します。



名前を「001」にします。



では、ブラウザで確認します。

Firefox を立ち上げて、

例)

<http://192.168.2.102/001.png> のようにアドレスを打ちます。

192.168.2.102 はこのテキストで使っている PC の IP アドレスなので、
テキスト P.69 で使ったコマンド

hostname -I

で調べた皆さんの IP アドレスを入れてください。

<http://000.000.000.000/001.png>



入力できたら、エンターキーを押します。

このように保存したスクリーンショットが表示されたら成功です。



このやり方で画像や動画をこのディレクトリにアップしてください。

Web サーバをルータに接続するとインターネットと同じようにファイルの共有ができます。画像や動画もクローズのネットワークの中で視聴することができます。

サーバの仕組み

ここからは、これまで構築してきたサーバの仕組みについて解説をします

データの格納場所：ドキュメントルート

サーバのファイルやディレクトリ（ウィンドウズで言うところのフォルダ）が置かれている上位の親ディレクトリを「ルート」といいます。よく URL で使われるスラッシュ「/」です。

構築したサーバのドキュメントルートは、
「/var/www/html/」になっています。

ですので、var フォルダの中の www フォルダの中の html フォルダの中を参照していることになります。

サーバを運用する場合、見られたくないディレクトリやファイルが多くあります。そのため、ネットワークにする場合、公開するデータをアクセスできる場所に置き、それ以外のディレクトリはアクセスできないようになっています。

外部からこの公開するディレクトリに置く方法は2通りあります。外からペーストするなどする方法とサーバ上で作成する方法です。

外部からサーバにデータを転送する場合は、FTP を使います。この場合 FTP サーバの機能を持つソフトをサーバにインストールし、外部とファイルのやりとりができるようにします。

FTP より、セキュリティレベルが高いものとして SCP や SFTP があります。これは転送するときに暗号化をします。一般に web サーバでは SSH というソフトをインストールして使います。

以上、基礎的ではありますが本格的なサーバ構築の方法と解説を終わります。

