

Linux サーバー構築手順書

大阪公立大学工業高等専門学校

3 年 知能情報コース 5 番

ai10pro

2025 年 2 月 10 日

目次

0 はじめに	4
1 Linux について	5
1.1 Linux とは	5
1.2 Linux ディストリビューション	5
1.3 ファイルやディレクトリ構造について	5
2 Ubuntu インストール	6
2.1 Ubuntu とは	6
2.2 インストール環境	6
2.3 インストールメディアの準備	6
2.4 Boot Mode の変更	7
2.5 Ubuntu のインストール	7
2.6 インストール後の再起動	8
3 Ubuntu の各種設定・インストール	8
3.1 日本語入力	8
3.2 Vim のインストール	8
3.3 VSCode のインストール	9
3.4 sl のインストール (お遊び)	9
3.5 cmatrix のインストール (お遊び)	10
4 Linux の基本コマンド	11
4.1 よく使うコマンド一覧	11
4.2 シェルに備わっている機能	18
5 パーミッション	20
5.1 ファイルの所有者	20
5.2 アクセス権	20
5.3 パーミッションの変更	20
5.4 所有者と所有グループ	21
6 環境変数	21
7 テキストエディタ	22
7.1 vi(Vim) について	23
7.2 Vim の基本操作	23
7.3 「i」キー以外の入力モードへの移行	24
7.4 ファイルの保存や終了 + α	24
8 ネットワーク設定	25
8.1 ポート番号	25
8.2 IP アドレスの管理と固定化	25
8.3 Netplan によるネットワーク設定	26
9 ユーザー管理	26
9.1 root ユーザーのパスワードの変更	27

9.2	一般ユーザーの作成・削除とグループへの追加・削除	27
9.3	ユーザーの切り替え	27
9.4	sudo コマンド	28
10	SSH	28
10.1	SSH について	28
10.2	SSH のサーバー構築	29
11	基本的なサーバー管理	31
11.1	サーバー負荷	31
11.2	ディスクの使用状況	31
11.3	メモリとスワップ	31
11.4	プロセスの管理	32
11.5	サービス管理	32
12	Web サーバーの構築	33
12.1	サーバーソフトウェアのインストール	33
12.2	Apache HTTP server のインストール	33
12.3	Apache の設定	34
12.4	ファイアウォールの設定	37
12.5	パスワード認証	37
12.6	アクセスログ	38
12.7	エラーログ	39
13	参考リンク	40
14	終わりに	41

0 章 はじめに

この手順書は、Linux にて Web サーバーを構築するまでの手順を記載している。ターゲットとしてネットワークや html などの知識が少しあり、サーバー構築をしてみたいという人や、Linux というものを触ってみたいという人とする。

筆者の Linux などの歴としては、本手順書を書いている段階で 1 年程度である。そのため、初心者にある程度寄り添えるように心がけている。

次章から説明や手順を記載しているため実際に手を動かしてもらうと理解が深まると思う。

それでは楽しい Linux のお時間をお過ごしください。

本書の環境

■ PC : Gateway NE132-f14p

- CPU : Intel Celeron N3060
- メモリ : 4GB
- ストレージ : eMMC 64GB

■ インストールする OS : Ubuntu 22.04 LTS

本手順書の作成には Tex における編集補助として GitHubCopilot を使用している。

また、本書の作成には VScode を使用している。

2025 年 2 月 10 日

1 章 Linux について

1.1 Linux とは

Linux は、UNIX 系のオペレーティングシステムの一つであり、オープンソースである。UNIX 系のオペレーティングシステムは、多くのサーバーで使用されている。Linux は、サーバーだけでなく、組み込みシステムやスマートフォン、スーパーコンピュータなど、様々な機器で使用されている。例として Raspberry Pi や近年話題となっている Steam deck などのポータブルゲーミングデバイスなども Linux をベース OS にしている。

1.2 Linux ディストリビューション

Linux は、様々なディストリビューションが存在する。ディストリビューションとは、Linux カーネルに、シェル、コマンド、ライブラリ、デスクトップ環境、アプリケーションなどを組み合わせたものである。代表的なものとして、Ubuntu、Debian、CentOS、Raspberry Pi OS などがある。
本書では Ubuntu の環境を前提として説明を行う。

1.2.1 カーネル

Linux のカーネルは、Linux の中核となる部分であり、ハードウェアとソフトウェアの間のインターフェースを提供する。カーネルは、プロセス管理、メモリ管理、ファイルシステム、デバイスドライバなどの機能を提供する。

1.2.2 シェル

シェルは、ユーザーとカーネルの間のインターフェースを提供する。シェルは、ユーザーがコマンドを入力し、カーネルに命令を伝える。代表的なシェルとして、bash、zsh、fish などがある。

1.2.3 コマンド

シェルを通して実行するプログラムであり、Linux には数千にもなるコマンドが存在するが基本的に使用する物は限られているためサーバー運用においては覚えておくことで作業効率が向上する。

1.2.4 ライブラリ

ライブラリは、プログラムの共通部分をまとめたものであり、プログラムの開発を効率化する。ライブラリは、プログラムの実行時にリンクされる。

1.3 ファイルやディレクトリ構造について

Linux ではディレクトリ構造が Windows とは異なる。

1.3.1 ファイルの種類

Linux で扱われるファイルは以下のように分類される。

- ファイル：テキストファイルとバイナリファイル
- リンクファイル：シンボリックリンクとハードリンク
- 特殊ファイル：デバイスファイルや隠しファイル
- ディレクトリ：ファイルを集約するための Windows でのいうフォルダ

拡張子は Windows においてはアプリケーションの起動に紐づけられていたが、Linux では単なるファイル名の一部として認識される。

Linux ではデバイスファイルがあり、これはハードウェアをファイルとして扱うためのものである。

1.3.2 ディレクトリ構造

Linux のディレクトリ構造は/(ルート) を起点として以下のディレクトリが主に挙げられる。それらがある程度理解しておくことで各種設定などの変更の際に便利である。

表 1 に主要なディレクトリとその説明を示す。

ディレクトリ	説明
/root	システム管理者のホームディレクトリ
/home	一般ユーザーのホームディレクトリ
/bin	一般ユーザーでも実行可能なコマンドが含まれるディレクトリ
/sbin	sudo 権限または root 権限が必要なコマンドが含まれるディレクトリ
/media	USB メモリや外付け HDD などのデバイスがマウントされるディレクトリ
/etc	システムの設定ファイルが含まれるディレクトリ
/usr	システム全体で共有されるアプリケーション、ライブラリ、ドキュメントなどが保存されるディレクトリ
/lib	ライブラリが保存されるディレクトリ
/proc	カーネルやプロセスに関する情報が保存されるディレクトリ
/tmp	一時ファイルが保存されるディレクトリ
/var	変化するデータが保存されるディレクトリ、ログファイルやデータベースなど

表 1 主要なディレクトリとその説明

2 章 Ubuntu インストール

2.1 Ubuntu とは

Ubuntu は、Linux ディストリビューションの一つであり、デスクトップ環境を提供する。また、サーバー版も提供されており、サーバー運用にも使用される。Ubuntu は、Debian をベースにしており、Debian の安定性と Ubuntu の使いやすさを兼ね備えている。また、Ubuntu は、オープンソースであり、無料で利用することができる。

2.2 インストール環境

本書の環境を再度記載する。

■ PC：Gateway NE132-f14p

- CPU：Intel Celeron N3060
- メモリ：4GB
- ストレージ：eMMC 64GB

■インストールする OS：Ubuntu 22.04 LTS

i 最新版 Ubuntu

最新の Ubuntu のバージョンは 24.04 LTS である。しかし、本書では 22.04 LTS を使用する。理由として、ドキュメントが豊富であり、安定しているためである。最新バージョンと 22.04 LTS では一部 UI が違う部分もあるため、注意が必要である。

2.3 インストールメディアの準備

1. Ubuntu の ISO ファイルをダウンロード

情報

Ubuntu を入手する <https://releases.ubuntu.com/20.04/>

Ubuntu Desktop 日本語 Remix のダウンロード <https://www.ubuntulinux.jp/download/ja-remix>

2. Rufus をダウンロード

情報

Rufus を入手する <https://rufus.ie/ja/>

3. Rufus を起動し、デバイス、ブート選択、イメージ選択を行う

デバイスにはインストールメディアとするための USB を選択する。

ブート選択では、先ほどダウンロードした ISO ファイルを選択する。

パーティション構成は一般的には MBR でよい。ただし、Gateway の場合は GPT を選択する。その他の設定はデフォルトのままでよい。

注意

デバイスを選択する際は注意が必要である。間違えると PC や外部ストレージのデータが消える可能性があるため、注意が必要である。

4. スタートをクリックし、書き込みを開始する

2.4 Boot Mode の変更

1. 電源をつけ F2 を連打し BIOS 画面を表示

BIOS 画面の操作

BIOS 画面は基本的にキーボードで操作されることが多い。一部メーカーの場合はマウスで操作することもあるが、基本的にはキーボードでの操作が必要である。

2. Security に移動し Supervisor Password を設定

一度 Enter を押し、パスワードを入力し、再度 Enter を押し、もう一度パスワードを入力し、Enter を押す。

3. Boot に移動し Boot Mode を Legacy に変更

4. F10 で保存して再起動

5. F2 を連打し BIOS 画面を表示

6. Boot に移動し Boot Priority を変更しインストールメディアが入るものを選択

作成したインストールメディアを選択する。選択には F5 または F6 を使用し順番を変更する。

7. Boot Mode を UEFI に変更

8. SecureBoot を Disable に変更

9. F10 で保存して再起動

2.5 Ubuntu のインストール

1. 再起動後自動的に Boot Menu が表示される

2. Try or Install Ubuntu を選択

3. インストールの種類を選択
4. インストール先を選択
5. キーボードレイアウトを選択
6. ユーザー情報を入力
7. インストールが完了したら再起動

2.6 インストール後の再起動

1. 再起動時に Bios に入る
2. Boot に移動し Boot Priority を変更しインストールしたディスクを 1 番目にする
3. Secure を Enable に変更
4. Security に移動し Erase all Secure Boot Setting を選択し、Boot 設定を削除
5. Select an UEFI file as trusted for executing を選択
6. EMMC >EFI >Ubuntu >shimx64.efi を選択
7. Boot 名を入力し、Enter
8. Boot に移動し Secure Boot を Disable に変更
9. Security に移動し Supervisor Password を削除
Supervisor Password にて Enter を押し、今のパスワードを入力、Enter を押し、何も入力しないまま Enter を押す、もう一度行うことでパスワードが削除される。
10. F10 で保存して再起動
11. ログイン画面になったら成功

3 章 Ubuntu の各種設定・インストール

Ubuntu (Linux) は Windows などと同様に様々な設定が可能である。以下にはその一部を示すが、適宜自分にあった Ubuntu 環境を構築することが望ましい。練習ついでにいろいろな物を入れてみるとよい。

3.1 日本語入力

Ubuntu ではデフォルトでキーボードによる日本語入力は対応していない。そのため、以下の手順で日本語入力を可能にする。なお日本語入力には一般に「Mozc」が使用されることが多い。

1. ターミナルを開く
2. 以下のコマンドを実行

```
>_ terminal
$ sudo apt update
$ sudo apt install ibus-mozc
```

3. 設定 > キーボード > 入力ソース > 日本語 (Mozc) を追加
4. Windows キー + Space で日本語入力が可能になる
なお、右上にある「A」をクリックすることで入力ソースを切り替えることができる。

3.2 Vim のインストール

Vim は Linux 環境で使用されるテキストエディタであり、非常に高機能である。特徴として、ターミナル上で操作するため、マウスを使わずに編集を行う。本手順書では sudo 権限（後の章にて解説）の必要なファイルの編集などで使用する。標準で搭載されている場合もある。

1. ターミナルを開く
2. 以下のコマンドを実行

```
>_ terminal
$ sudo apt update
$ sudo apt install vim
```

3. インストールが完了したら、以下コマンドを実行しインストールされているか確認

```
>_ terminal
$ vim --version
```

3.3 VSCode のインストール

VSCode は Microsoft が提供するオープンソースのコードエディタであり、非常に高機能である。特徴として、拡張機能を追加することで、様々な言語に対応することができる。また、リモート開発機能を使用することで、リモートサーバー上での開発も可能である。

1. <https://code.visualstudio.com/>にアクセスし、ダウンロード
2. ダウンロードしたファイルを開く
3. インストールを開始
4. インストールが完了したら、VSCode を起動

3.4 sl のインストール（お遊び）

sl は、タイプミスでよく入力される「ls」コマンドを実行すると、蒸気機関車が走るアニメーションが表示されるコマンドである。インストールの練習として入れてみるとよい。

1. ターミナルを開く
2. 以下のコマンドを実行

```
>_ terminal
$ sudo apt update
$ sudo apt install sl
```

3. 以下のコマンドを実行

```
>_ terminal
$ sl
```

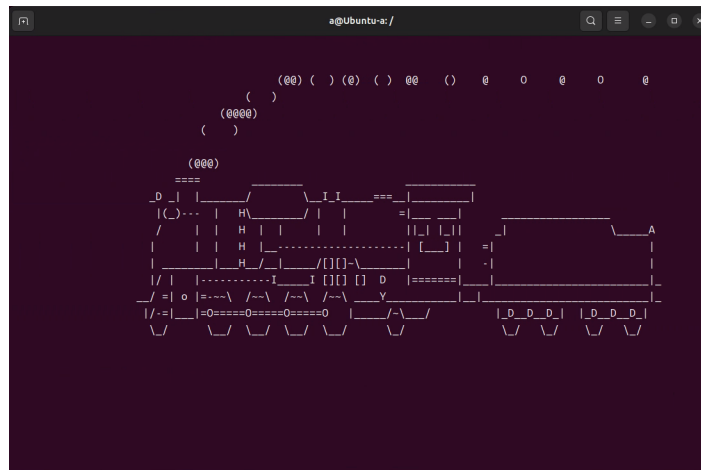


図1 slの実行結果

実行時にオプションがあるので、気が向いたら調べて遊んでみるとよい。

3.5 cmatrix のインストール (お遊び)

cmatrix は、映画「マトリックス」に登場するコードのような文字列が画面上を流れるアニメーションを表示するコマンドである。ハッカーみたいになれるのでなりたいたい人は入れるとよい。

1. ターミナルを開く
2. 以下のコマンドを実行

```
>_ terminal
$ sudo apt update
$ sudo apt install cmatrix
```

3. 以下のコマンドを実行

```
>_ terminal
$ cmatrix
```

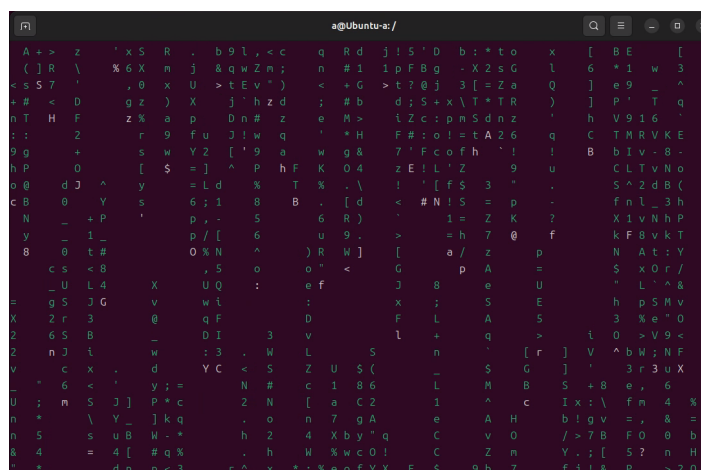


図2 cmatrixの実行結果

4 章 Linux の基本コマンド

サーバー上でのシステム操作は一般的にターミナルを用いてコマンド入力により操作を行うことが多い。この章では Linux の基本コマンドとして紹介するが、一部コマンドは Windows でも使用できる。ただし、Windows のコマンドは Linux とは異なるため注意が必要である。

コマンドは一般的に以下のように入力する。

>_ コマンドの入力方法

```
$ {コマンド名} {オプション} {引数}
```

4.1 よく使うコマンド一覧

4.1.1 ファイルやディレクトリの操作

- `ls` : ファイルやディレクトリの一覧を表示する
オプションなしの場合、カレントディレクトリのファイルやディレクトリの一覧を表示する。

```
>_ terminal
```

```
$ ls bin
```

■ `ls` コマンドのオプション

-a : 隠しファイルも表示する
-l : 詳細情報を表示する (パーミッション情報)
-h : ファイルサイズを見やすい形式で表示する

- `cd` : カレントディレクトリを移動する
ディレクトリ名の場所にカレントディレクトリを移動する。

```
>_ terminal
```

```
$ cd {ディレクトリ名}
```

```
# ルートディレクトリへの移動
```

```
$ cd /
```

- `pwd` : 現在のディレクトリのパスを表示する

```
>_ terminal
```

```
$ pwd
```

```
例: /home での実行結果
```

```
/home
```

- `mkdir` : ディレクトリを作成する
`path` に指定したディレクトリを作成する。

```
>_ terminal
```

```
$ mkdir {ディレクトリ名}
```

例：test ディレクトリを作成する

```
$ mkdir test
```

- rm：ファイルやディレクトリを削除する
一部ファイルは sudo 権限が必要である。

```
>_ terminal
```

```
$ rm {ファイル名}
```

例：test.txt を削除する

```
$ rm test.txt
```

- rmdir：ディレクトリを削除する
ディレクトリ内にファイルがある場合は削除できない。

```
>_ terminal
```

```
$ rmdir {ディレクトリ名}
```

例：test ディレクトリを削除する

```
$ rmdir test
```

- cp：ファイルやディレクトリをコピーする
copy であるため元のファイルは残る。

```
>_ terminal
```

```
$ cp {コピー元} {コピー先}
```

例：test.txt を test2.txt にコピーする

```
$ cp test.txt test2.txt
```

- mv：ファイルやディレクトリを移動する
move であるため元のファイルは消える。

```
>_ terminal
```

```
$ mv {移動元} {移動先}
```

例：test.txt を test2.txt に移動する

```
$ mv test.txt test2.txt
```

- find：ファイルを検索する

```
>_ terminal
```

```
$ find {検索先} -name {検索ファイル名}
```

- touch : ファイルを作成する

```
>_ terminal
```

```
$ touch {ファイル名}
```

例 : test.txt ファイルを作成する

```
$ touch test.txt
```

- ln: シンボリックリンクの作成

ファイルやディレクトリに対してシンボリックリンクを作成する。シンボリックリンクとは Windows におけるショートカットの役割を果たすものである。

```
>_ terminal
```

```
$ ln -s {リンク元} {リンク先}
```

例 : test.txt のシンボリックリンクをデスクトップに作成する

```
$ ln -s test.txt ~/Desktop/test.txt
```

4.1.2 テキスト操作

- cat : ファイルの内容を表示する

ファイルの内容をターミナル上でスクロールする形で表示する。

```
>_ terminal
```

```
$ cat {ファイル名}
```

- less : ファイルの内容をページ単位で表示するスペースキーで次のページ、q キーで終了する。

```
>_ terminal
```

```
$ less {ファイル名}
```

- head : ファイルの先頭部分を表示する

ファイルの先頭部分を表示する。

```
>_ terminal
```

```
$ head {ファイル名}
```

- tail : ファイルの末尾部分を表示する

ファイルの末尾部分を表示する。

```
>_ terminal
```

```
$ tail {ファイル名}
```

- `grep` : ファイル内の文字列を検索する
ファイル内の文字列を検索する。

```
>_ terminal
$ grep {検索文字列} {ファイル名}

例: test.txt 内に「test」が含まれているか検索する
$ grep test test.txt
```

- `vi` : Vim(テキストエディタ) の起動
Vim の使用方法については後述する。

```
>_ terminal
$ vi {ファイル名}

例: test.txt を編集する
$ vi test.txt
```

- `code` : VSCode(テキストエディタ) の起動
ファイル名の部分にはディレクトリも指定できる。その場合、そのディレクトリが VSCode で開かれる。

```
>_ terminal
$ code {ファイル名}

例: test.txt を編集する
$ code test.txt
```

4.1.3 パーミッションや所有者

- `chmod` : ファイルやディレクトリのパーミッションを変更する

```
>_ terminal
$ chmod {パーミッション} {ファイル名}

例: test.txt のパーミッションを rwxr-xr-- に変更する
$ chmod 754 test.txt
```

パーミッションは `rwxrwxrwx` の形式で表される。r は読み取り、w は書き込み、x は実行を表す。
例えば、`rwxr-xr--` は所有者に読み取り、書き込み、実行権限があり、グループに読み取り、実行権限があり、その他に読み取り権限があることを示す。

■ パーミッションの指定方法について

パーミッションは数字で指定することができる。
`rwX/rwX/rwX` という風に 3 桁ごとに区切って考える。r を 4, w を 2, x を 1 として、それぞれの権限を足し合わせる。したがって、`rwxr-xr--` は 754 と表すことができる。

※この数字は2進数で処理しており、rwxの時111となり、合計が7となる。

- chown : ファイルやディレクトリの所有者を変更する

```
>_ terminal
$ chown {所有者}:{グループ} {ファイル名}

例: test.txt の所有者を root に変更する
$ chown root test.txt
```

グループ名は省略することもできる。その場合、所有者と同じグループになる。

- chgrp : ファイルやディレクトリのグループを変更する

```
>_ terminal
$ chgrp {グループ} {ファイル名}

例: test.txt のグループを root に変更する
$ chgrp root test.txt
```

4.1.4 ネットワーク管理

- ping : ネットワークの接続確認を行う

```
>_ terminal
$ ping {IP アドレス}

例: 192.168.0.150 に ping を送信する
$ ping 192.168.0.150
```

📖 ping コマンドについて

ping コマンドはネットワーク内の機器の接続確認でよく使用される。Windows でも同様に使用できる。ただし、Windows は4回の送信で終了するが、Linux は Ctrl + C で終了するまで続く。

📖 ping コマンドのオプション

-c : 指定回数の送信を行う
-i : 送信間隔を指定する
-t : 継続的に送信を行う

- ifconfig : ネットワークインターフェースの情報を表示する

```
>_ terminal
$ ifconfig
```

出力例は以下の通りである。

wlp6s0:	flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500	概要
接続名	inet 192.168.0.12 netmask 255.255.255.0 broadcast 192.168.0.255 inet6 2001:ce8:115:7d:2aea:30a:aef3:1de1 prefixlen 64 scopeid 0x0<global> inet6 fe80::d5d4:adc6:91b0:a180 prefixlen 64 scopeid 0x20<link> inet6 2001:ce8:115:7d:6551:9937:a77a:32e prefixlen 64 scopeid 0x0<global> ether 2c:6e:85:09:22:20 txqueuelen 1000 (イーサネット)	IPアドレス関連
	RX packets 361 bytes 57347 (57.3 KB) RX errors 0 dropped 0 overruns 0 frame 0	受信系
	TX packets 168 bytes 31839 (31.8 KB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0	送信系

図 3 ifconfig の出力例

各種いろいろなネットワークアダプタに関する情報を取得できるがため調べて利用するとよい。ただし、IP アドレス関連にある inet アドレスは、IPv4 のアドレスでありよく使うため覚えておくとい。

- route : ルーティングテーブルの情報を表示する

i ルーティングとは

ルーティングとは、ネットワーク上のデータを送信する際に、どの経路を通して送信するかを決定することである。ルーティングテーブルは、どの経路を通して送信するかを示すものである。

```
>_ terminal
```

```
$ route
```

- netstat : ネットワークの接続状況を表示する

```
>_ terminal
```

```
$ netstat
```

例：全ての接続状況を表示する

```
$ netstat -a
```

netstat コマンドのオプション

-a : 全ての接続状況を表示する

-t : TCP 接続のみを表示する

-u : UDP 接続のみを表示する

-n : IP アドレスを表示する

- ssh : リモートサーバーに接続する
詳細については 10 章で解説する。

```
>_ terminal
```

```
$ ssh {ユーザー名}@{IP アドレス}
```

- ftp : ファイル転送プロトコルを使用してファイルを転送する


```
>_ terminal
$ ftp {IP アドレス}
```

4.1.5 システム管理

- top : システムのリソース使用状況を表示する

```
>_ terminal
$ top
```

■ top コマンドの操作方法

q : 終了する
P : CPU 使用率でソートする
M : メモリ使用率でソートする
T : 実行時間でソートする

- ps : 現在実行中のプロセスの情報を表示する

```
>_ terminal
$ ps

例：全てのプロセスを表示する
$ ps -ef
```

■ ps コマンドのオプション

-e : 全てのプロセスを表示する
-f : 詳細情報を表示する

- kill : プロセスを終了する

```
>_ terminal
$ kill {プロセス ID}

例：プロセス ID が 1234 のプロセスを終了する
$ kill 1234
```

- df : ディスクの使用状況を表示する

```
>_ terminal
$ df
```

- du : ディレクトリのディスク使用量を表示する

```
>_ terminal
```

```
$ du {ディレクトリ名}
```

例：test ディレクトリのディスク使用量を表示する

```
$ du test
```

- free : メモリの使用状況を表示する

```
>_ terminal
```

```
$ free
```

- uptime : システムの稼働時間を表示する

```
>_ terminal
```

```
$ uptime
```

- uname : システム情報を表示する

```
>_ terminal
```

```
$ uname
```

例：システム情報を表示する

```
$ uname -a
```

📌 uname コマンドのオプション

- a : 全ての情報を表示する
- s : カーネル名を表示する
- r : カーネルリリースを表示する
- v : カーネルバージョンを表示する
- m : マシンハードウェアを表示する

4.1.6 ユーザー管理

以下コマンドは sudo 権限が必要である。

各コマンドの詳細は後述する。

- useradd / adduser : ユーザーの新規作成
- passwd : パスワードの変更
- usermod / chfn : ユーザー情報の変更
- userdel : ユーザーの削除
- groupadd : グループの新規作成
- who : ログイン中のユーザーを表示する

4.2 シェルに備わっている機能

シェルには様々な機能が備わっており、これらを使用することで作業効率を向上させることができる。以下に代表的な機能を紹介する。

4.2.1 入力補完

入力補完は、コマンドやファイル名の入力時に Tab キーを押すことで、入力を補完する機能である。入力補完を使用することで、入力ミスを防ぐことができる。使用方法是入力中に Tab キーを押すだけである。ただし、複数の候補がある場合は Tab キーを押しても補完されない場合がある。その場合は Tab キーを 2 回押すことで候補が表示される。

4.2.2 コマンド履歴

コマンド履歴は、過去に入力したコマンドを表示する機能である。矢印キーを使用することで、過去に入力したコマンドを表示することができる。矢印キーの上を押すことで過去のコマンドを表示し、矢印キーの下を押すことで新しいコマンドを表示することができる。

■ インクリメンタル検索

Ctrl + R を押すことで、過去に入力したコマンドを検索することができる。過去に入力したコマンドを検索し下矢印キーを押すことで検索したコマンドを入力することができる。

4.2.3 パイプとリダイレクト

Linux でコマンド入力をする際複数コマンドを組み合わせたい時がある。それらを組み合わせる際に使用するのがパイプである。パイプは「|」を使用してコマンドをつなげることで、前のコマンドの出力を次のコマンドの入力として使用することができる。

```
>_ terminal
```

```
$ {コマンド 1} | {コマンド 2}
```

例：ls コマンドの出力を grep コマンドに渡して test を検索する

```
$ ls | grep test
```

また、実行したコマンドの出力をファイルに保存することができる。これをリダイレクトという。リダイレクトは「>」を使用して出力をファイルに保存することができる。

```
>_ terminal
```

```
$ {コマンド} > {ファイル名}
```

例：ls コマンドの出力を test.txt に保存する

```
$ ls > test.txt
```

■ 上書きと追記

リダイレクトする際、ファイルが存在する場合は上書きされる。ファイルの内容を保持したまま追記したい場合は「>>」を使用する。

5 章 パーミッション

5.1 ファイルの所有者

Linux ではファイルやディレクトリに所有者が設定されている。所有者はファイルやディレクトリを作成したユーザーであり、所有者はファイルやディレクトリに対して権限を持っている。所有者はユーザー名で表される。実際に確認するには、ls コマンドに「-l」オプションを付けて実行する。実行結果は以下のようになる。

```
tori@tori-PC-GN17CJSA6:~/test$ ls -l
合計 0
-rw-rw-r-- 1 tori tori 0 2月 8 01:53 test.txt
```

アクセス権 所有者 所有グループ

図 4 ls -l の実行結果

5.2 アクセス権

Linux ではファイルやディレクトリに対してアクセス権が設定されている。アクセス権は読み取り、書き込み、実行の 3 つの権限が設定されている。アクセス権は所有者、グループ、その他の 3 つに設定されている。

■ ファイルのアクセス権

- r : 読み取り権限 ファイルの内容を読み取ることができる
- w : 書き込み権限 ファイルの内容を書き込むことができる
- x : 実行権限 ファイルを実行することができる

■ ディレクトリのアクセス権

- r : 読み取り権限 ディレクトリ内のファイルを表示することができる
- w : 書き込み権限 ディレクトリ内にファイルを作成することができる
- x : 実行権限 ディレクトリ内に移動することができる

① ls -l で表示される種類について

ls -l で表示される先頭の文字はファイルの種類を示している。以下に一部を示す。

- - : 通常のファイル
- d : ディレクトリ
- l : シンボリックリンク

5.3 パーMISSIONの変更

アクセス権の変更には chmod コマンドを使用する。ただし、アクセス権の変更には sudo 権限が必要である。

➤ chmod コマンドの使用方法

```
chmod {パーミッション} {ファイル名}
chmod {パーミッション} {ディレクトリ名}
```

また、アクセス権の変更には記号を用いた方法と数値を用いた方法がある。数値を用いたものは前述しているため、ここでは記号を用いた方法について説明する。

表 2 chmod コマンドの記号について

対象	操作	権限
u: 所有者	+: 追加	r: 読み取り権限
g: 所有グループ	-: 削除	w: 書き込み権限
o: その他ユーザー	=: 指定	x: 実行権限
a: 全てのユーザー		

📌 chmod コマンドの使用例

- `chmod u+x test.txt` : test.txt に所有者に実行権限を追加する
- `chmod g-w test.txt` : test.txt に所有グループから書き込み権限を削除する
- `chmod o=r test.txt` : test.txt にその他ユーザーに読み取り権限を指定する
- `chmod a=rwx test.txt` : test.txt に全てのユーザーに読み取り、書き込み、実行権限を指定する

5.4 所有者と所有グループ

Linux におけるファイルやディレクトリは作成したユーザーが所有者・所有グループとなる。しかし、一定以上のサーバー管理となる場合、管理体系などの理由により管理者レベルの高いユーザーに所有権を譲渡する必要がある。その際に `chown` コマンドを使用する。

➤ chown コマンドの使用方法

```
chown {所有者} {ファイル名 (ディレクトリ名)}
chgrp {所有グループ} {ファイル名 (ディレクトリ名)}
```

⚠ chown コマンドの注意点

`chown` コマンドは `sudo` 権限が必要である。また、ファイルやディレクトリの所有者を変更する際は、そのファイルやディレクトリに対して所有者権限を持っている必要がある。

6 章 環境変数

環境変数は、システム全体で使用される変数である。環境変数はシェルが起動する際に設定され、シェルの終了時に破棄される。環境変数はシェルスクリプトやプログラムから参照することができる。主な環境変数は以下の通りである。

表 3 主な環境変数

環境変数	説明
HOME	ユーザーのホームディレクトリのパス
PATH	実行ファイルの検索パス
USER	ユーザー名
SHELL	シェルのパス
LANG	言語設定
PWD	カレントディレクトリのパス
HISTSIZE	コマンド履歴の最大値
HISTFILE	コマンド履歴の保存先

地齋に環境変数を表示するには、echo コマンドを使用する。この時、環境変数名の前に「\$」を付けることで環境変数の値を表示することができる。

➤ 環境変数の表示

```
$ echo ${環境変数名}
```

また、これらの環境変数の値などを一時的に変更したい場合は export コマンドを使用する。export コマンドを使用することで、環境変数の値を一時的に変更することができる。

➤ export コマンドの使用方法

```
export {環境変数名}={値}
```

例：HISTSIZE の値を 100 に変更する

```
$ export HISTSIZE=100
```

■ 環境変数の値を永続的に変更する

環境変数の値を永続的に変更するには、環境変数の設定を行うファイルに環境変数の設定を記述する。環境変数の設定を行うファイルは以下の通りである。変更には「.bash_profile」や「.bashrc」の末尾に export コマンドを追記する。

- /etc/profile
- /etc/bashrc
- ~/.bash_profile
- ~/.bashrc

❗ 環境変数の設定ファイルについて

「.bashrc」：システムのログイン時に読み込まれる隠しファイル。

「.bash_profile」：ユーザーがログインした際に読み込まれるファイル。

7 章 テキストエディタ

Linux におけるテキストエディタは、システムやサービスの変更などに使用される。特にシステムの設定変更においては Vim などのテキストエディタを使用することが多い。本手順書では簡単な Vim の使い方を紹介する。ただし、Vim の操作は独特であり、慣れるまで時間がかかるため、初心者には難しいかもしれない。

7.1 vi(Vim) について

Unix 系 OS において標準的なテキストエディタである Vim は、Vi の拡張版であり、Vi の機能に加えて様々な機能が追加されている。起動には vi コマンドを用いる。

>_ vi コマンドの使用方法

\$ vi {ファイル名}

7.2 Vim の基本操作

7.2.1 コマンドモードと入力モード

Vim にはコマンドモードと入力モードの 2 つのモードが存在する。コマンドモードでは、ファイルの編集や保存などの操作を行うことができる。入力モードでは、ファイルの編集を行うことができる。

vi エディタ (Vim) を起動すると、最初はコマンドモードになっている。入力モードに移行するには「i」を押す。入力モードからコマンドモードに戻るには「Esc」キーを押す。コマンドモードでの入力はコマンドとして処理される。一方、入力モードでは文字が入力される。すなわち困ってわからなくなったときひとまず「Esc」キーを押してコマンドモードに戻るとというのが基本である。

7.2.2 よく使うコマンド

(1) カーソル移動

表 4 カーソル移動のコマンド

コマンド	説明
h	左に移動する
j	下に移動する
k	上に移動する
l	右に移動する
0	行の先頭に移動する
\$	行の末尾に移動する
gg	ファイルの先頭に移動する
G	ファイルの末尾に移動する
: 行数	指定した行に移動する

(2) 切り取り/コピー/貼り付け

表 5 切り取り/コピー/貼り付けのコマンド

コマンド	説明
x	カーソルがある文字を削除する (Delete キー)
X	カーソルの左の文字を削除する (Backspace キー)
yy	カーソルがある行をコピーする
dd	カーソルがある行を切り取る
p	カーソルの下に貼り付ける
P	カーソルの上に貼り付ける

(3) 文字列の検索/置換

表 6 文字列の検索/置換のコマンド

コマンド	説明
/検索文字列	カーソル位置より下を対象に検索文字列を検索する
?検索文字列	カーソル位置より上を対象に検索文字列を検索する
n	次の検索結果に移動する
N	前の検索結果に移動する
:%s/置換前/置換後	置換前を置換後に置き換える
:%s/置換前/置換後/g	ファイル内の置換前を全て置換後に置き換え

7.3 「i」キー以外への入力モードへの移行

「i」キー以外にも入力モードに移行する方法がある。以下に代表的な方法を示す。

表 7 入力モードへの移行方法

コマンド	説明
i	カーソルの前から入力を開始
a	カーソルの後ろから入力を開始
I	行の先頭から入力を開始
A	行の末尾から入力を開始
o	カーソルの下に新しい行を挿入して入力を開始
O	カーソルの上に新しい行を挿入して入力を開始

7.4 ファイルの保存や終了+ α

ファイルの保存や終了はコマンドモードで行う。以下に代表的なコマンドを示す。

表 8 ファイルの保存や終了のコマンド

コマンド	説明
:w	ファイルを保存する
:wq	ファイルを保存して終了する
:q	ファイルを保存せずに終了する
:q!	ファイルを保存せずに強制終了する
:w ファイル名	別名でファイルを保存する

そのほかにも使えると便利なコマンドを以下に示す。

表 9 その他のコマンド

コマンド	説明
u	元に戻る (Undo)
Ctrl + r	やり直す (Redo)
.	直前のコマンドを繰り返す
:set number	行番号を表示する
:set nonumber	行番号を非表示にする
:set ignorecase	大文字小文字を区別しない検索を有効にする
:set noignorecase	大文字小文字を区別する検索を有効にする
:set hlsearch	検索結果をハイライト表示する
:set nohlsearch	ハイライト表示を解除する

8 章 ネットワーク設定

ここでは Linux におけるネットワーク設定について解説を行う。

⚠ ネットワーク知識について

ここでは IP アドレス、サブネットマスクなどネットワークに関する基礎知識があるものとしている。

8.1 ポート番号

通常サーバーでは複数のアプリケーションが動作している。クライアントとの通信時に IP アドレスしか情報を持たない場合、受信したデータがどのアプリケーションで利用されるのか不明である。そのため、通信時に IP アドレスと対象となるポート番号を指定することでアプリケーションに情報を伝達する。ポート番号は 0 から 65535 までの範囲で指定される。

■ ウェルノウンポート番号

0 から 1023 までのポート番号をウェルノウンポート番号と呼ぶ。これらのポート番号は一般的にすでに使用されているため、新たに使用することは推奨されない。

- 20 : FTP データ転送
- 21 : FTP 制御情報
- 22 : SSH
- 25 : メール (SMTP)
- 80 : HTTP
- 443 : HTTPS

また、使用済みポート番号は `/etc/services` ファイルに記載されている。

8.2 IP アドレスの管理と固定化

通常 Linux に限らず、機器をネットワークに接続すると DHCP により IP アドレスが自動で割り当てられる。しかし、サーバーにおいて IP アドレスが変わる可能性がある場合、IP が変わったことによるアクセス障害などにつながる場合がある。そのため、サーバーにおいては固定 IP アドレスを設定することが推奨される。

📖 DHCP

DHCP (Dynamic Host Configuration Protocol) は、ネットワークに接続された機器に IP アドレスなどのネットワーク設定を自動で割り当てるプロトコルである。DHCP により IP アドレスが自動で割り当てられるため、ネットワークの管理が容易になる。また、これにより割り当てられた IP アドレスを一般的に動的 IP アドレスと呼ぶ。

8.3 Netplan によるネットワーク設定

Ubuntu では Netplan を使用してネットワーク設定を行う。Netplan は YAML 形式でネットワーク設定を記述する。以下に Netplan の設定ファイルの例を示す。

⚠️ 設定ファイルのバックアップ

通常設定ファイルの変更の際には、設定ファイルのバックアップを取得することを推奨する。cp コマンドを使用し、設定ファイルのバックアップを作成しておく。

➤ 設定ファイルのバックアップ

```
$ cp /etc/netplan/01-netcfg.yaml /etc/netplan/01-netcfg.yaml.bak
```

➤ /etc/netplan/01-netcfg.yaml

```
network:
  version: 2
  renderer: networkd # ネットワークの設定方法
  ethernets:
    enp0s3: # インターフェース名
      dhcp4: no # DHCP を使用しない
      addresses:
        - 192.168.0.200/24 # 固定 IP アドレス
      gateway4: 192.168.0.1 # ゲートウェイ
      nameservers:
        addresses:
          - 8.8.8.8 # DNS サーバー
          - 8.8.4.4 # 代替 DNS サーバー
```

設定ファイルの変更後、変更を適用するためには以下のコマンドを実行する。

➤ ネットワーク設定の適用

```
$ sudo netplan apply
```

実行後接続を確認して終了となる。

9 章 ユーザー管理

Linux におけるサーバー作業は一般的に root ユーザーで行うことは推奨されない。そのため、一般ユーザーを作成し、sudo 権限を付与することが推奨される。一部 root 権限が必要な場合にのみ root ユーザーで作業を行うことが推

奨される。

現在ログインしているユーザーを見分けるにはターミナルに表示される最後の「\$」や「#」を確認する。一般ユーザーの場合は「\$」、root ユーザーの場合は「#」が表示される。

9.1 root ユーザーのパスワードの変更

初期状態では root ユーザーにパスワードは設定されていない。そのため、root ユーザーにパスワードを設定する必要がある。

➤_ root ユーザーのパスワードの変更

```
$ sudo passwd root
```

9.2 一般ユーザーの作成・削除とグループへの追加・削除

サーバー管理において管理する人が増えたりする。その際にユーザーの作成、削除、グループへの追加、削除などが必要になる。以下にそれらの操作方法を示す。

➤_ ユーザーの作成

```
$ sudo adduser {ユーザー名}
```

例：user1 というユーザーを作成する場合

```
$ sudo adduser user1
```

➤_ ユーザーの削除

```
$ sudo deluser {ユーザー名}
```

例：user1 というユーザーを削除する場合

```
$ sudo deluser user1
```

➤_ ユーザをグループへ追加 (usermod)

```
$ sudo usermod -aG {グループ名} {ユーザー名}
```

例：user1 を admin グループに追加する場合

```
$ sudo usermod -aG admin user1
```

➤_ ユーザをグループから削除 (gpasswd)

```
$ sudo gpasswd -d {ユーザー名} {グループ名}
```

例：user1 を admin グループから削除する場合

```
$ sudo gpasswd -d user1 admin
```

9.3 ユーザーの切り替え

Linux においては複数のユーザーで作業を行うことがある。その際にユーザーを切り替える方法を以下に示す。切り替えには su コマンドを使用する。

➤ ユーザーの切り替え

```
$ su {ユーザー名}
```

例：user1 に切り替える場合

```
$ su user1
```

📌 su コマンドを使って root ユーザーに切り替える

root ユーザーに切り替える場合は以下のように入力することで切り替えることができる。

➤ root ユーザーに切り替える

```
$ su #または
```

```
$ su -
```

パスワードを入力する

9.4 sudo コマンド

実際の作業においては、root ユーザーで作業を行うことは推奨されない。そのため、一般ユーザーに sudo 権限を付与し、必要な時にのみ root 権限で作業を行うことが推奨される。sudo コマンドを使用することで、一般ユーザーに root 権限を付与することができる。sudo 権限を付与するには、ユーザーを sudo グループに追加することで使用可能になる。

10 章 SSH

10.1 SSH について

SSH（Secure Shell）は、ネットワーク上で安全に通信を行うためのプロトコルである。SSH は、リモートホスト間の通信を安全に行うために使用される。

SSH はでは2つの認証が行われており、ホスト認証とユーザー認証が行われる。ホスト認証は、通信先のホストが正しいかどうかを確認するために行われる。SSH 接続時に、サーバーから固有のホスト認証鍵（公開鍵）がクライアントに送られ、クライアントはその鍵を使用してホスト認証を行う。ただし、初回接続時にはホストの認証鍵がないため、サーバーから送られてくるホスト認証鍵を登録する必要がある。

ホスト認証後にユーザー認証が行われる。ユーザー認証は、ユーザーが正しいかどうかを確認するために行われる。ユーザー認証にはパスワード認証や公開鍵認証などがある。パスワード認証は、ユーザー名とパスワードを使用して認証を行う。一方、公開鍵認証は、公開鍵と秘密鍵を使用して認証を行う。公開鍵認証はパスワード認証よりもセキュリティが高いため、推奨される。

表 10 SSH のユーザー認証方法

	パスワード認証	公開鍵認証
認証に必要な要素	ユーザー名とパスワード	ユーザー名、パスフレーズ、秘密鍵
セキュリティについて	総当たりに対して脆弱	秘密鍵の流出がない限り安全

近年ではこれら2つを組み合わせる二段階認証が推奨される。

10.2 SSH のサーバー構築

10.2.1 OpenSSH のインストール

Ubuntu での SSH サーバーの構築には OpenSSH を使用する。OpenSSH は SSH プロトコルを実装したオープンソースのソフトウェアである。OpenSSH は、SSH プロトコルの実装として最も広く使用されている。

OpenSSH のインストールは以下のコマンドを実行することで行うことができる。

```
>_ terminal
$ sudo apt install openssh-server
```

インストール後、systemctl コマンドを使用して SSH サーバーを起動・起動確認を行う。

```
>_ terminal
$ sudo systemctl start ssh
$ sudo systemctl enable ssh
$ sudo systemctl status ssh
```

以下のような表示がされれば SSH サーバーの起動に成功している。

```
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2025-02-08 23:29:54 JST; 1min 20s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 3384 (sshd)
    Tasks: 1 (limit: 4494)
   Memory: 1.7M
      CPU: 36ms
   CGroup: /system.slice/ssh.service
           └─3384 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

2月 08 23:29:54 tori-PC-GN17CJSA6 systemd[1]: Starting OpenBSD Secure Shell server...
2月 08 23:29:54 tori-PC-GN17CJSA6 sshd[3384]: Server listening on 0.0.0.0 port 22.
2月 08 23:29:54 tori-PC-GN17CJSA6 sshd[3384]: Server listening on :: port 22.
2月 08 23:29:54 tori-PC-GN17CJSA6 systemd[1]: Started OpenBSD Secure Shell server.
```

図 5 SSH サーバーの起動確認

SSH サーバーの起動

初期設定ではシステム起動時に SSH サーバーが自動で起動されるように設定されている。しかし、手動で SSH サーバーを起動する場合は「systemctl start ssh」コマンドを使用する。

10.2.2 ポート番号の設定

SSH サーバーはデフォルトで 22 番ポートを使用する。しかし、22 番ポートは前述している通りウェルノウンポート番号であるため、セキュリティ上のリスクがある。そのため、一般的にポート番号を変更することが推奨される。ポート番号の変更は sshd_config ファイルを編集することで行うことができる。

⚠ sshd_config ファイルのバックアップ

通常設定ファイルの変更の際には、設定ファイルのバックアップを取得することを推奨する。cp コマンドを使用し、設定ファイルのバックアップを作成しておく。

➤_ 設定ファイルのバックアップ

```
$ cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
```

ポート番号の変更は sshd_config ファイル内に「#Port22」と書かれている箇所のコメントアウトを外し、ポート番号を指定することで変更することができる。

➤_ sshd_config ファイルの編集

```
#Port 22  
Port 10022
```

このままではファイアーウォールで設定されているポート番号に対して許可されていないため、ファイアーウォールの設定を変更する必要がある。

➤_ ファイアーウォールの設定

```
$ sudo ufw allow 10022/tcp
```

i TCP と UDP?

TCP (Transmission Control Protocol) と UDP (User Datagram Protocol) は、インターネットで使用する通信プロトコルである。TCP は比較的安定した通信を行いたいときに使用される。一方、UDP は速度を重視する通信を行いたいときに使用される。

TCP は SSH など通信の安全を求める。一方、UDP は動画配信などの速度を重視する通信に使用される。

10.2.3 ログインの制限

SSH サーバーは外部から全ての権限がある root へのログインを制限することが推奨される。これにより、root ユーザーに対する攻撃を防ぐことができる。ログインの制限は sshd_config ファイルを編集することで行うことができる。

➤_ sshd_config ファイルの編集

```
PermitRootLogin no    # 追加する
```

また、特定のユーザーのみログインを許可する場合は AllowUsers を使用する。

➤_ sshd_config ファイルの編集

```
AllowUsers user1 user2    # 追加する
```

10.2.4 SSH ログイン

ここまでできたら一度 SSH 接続を行ってみる。SSH 接続の接続元 (SSH サーバーを立てた PC) をサーバー、接続する PC をクライアントと呼ぶ。接続は以下のコマンドをクライアントで使用する。なお、実行には Windows では PoserShell、Linux ではターミナルを使用する。

```
>_ terminal(クライアント PC)
```

```
$ ssh {ユーザー名}@{IP アドレス} -p {ポート番号}
```

コマンド実行後、パスワードを求められるので入力する。正しく入力された場合、SSH サーバーに接続される。

■ 初回接続

初回接続時にはホスト認証鍵がないため、接続時にホスト認証鍵を登録するかどうかを尋ねられる。yes を入力することでホスト認証鍵を登録することができる。

本書は基礎知識 + Web サーバーの構築を目的とするため二段階認証についての説明は省略する。ただし、近年実際のネットワークを介してアクセスをする場合は必須とされているので、調べ、できるようになっておくべきである。

11 章 基本的なサーバー管理

11.1 サーバー負荷

サーバー運用において負荷の理解は重要である。どの部分に負荷がかかっているのかを理解することでアルゴリズムの最適化やサーバースペックの見直しなどを行うことができる。サーバーの負荷を確認するには uptime コマンドを使用する。

>_ 負荷の確認

```
$ uptime
10:00:00 up 1 day, 1:00, 1 user, load average: 0.00, 0.01, 0.05
```

uptime コマンドの出力は稼働時間の他に load average が表示される。load average は 1 分、5 分、15 分の負荷平均を示している。ここでいう負荷とは CPU のコア数に対する平均負荷を示している。例えば、CPU が 4 コアの場合、load average が 4 を超えると CPU がフル稼働している状態となる。

11.2 ディスクの使用状況

ディスクの使用状況を確認するには df コマンドを使用する。

>_ ディスクの使用状況の確認

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
.....
```

df コマンドの出力はファイルシステム、容量、使用、空き、使用率、マウントポイントが表示される。使用率が 100% になるとディスクがいっぱいになっているため、ディスク容量の拡張や不要なファイルの削除などを行う必要がある。

11.3 メモリとスワップ

メモリとスワップの使用状況を確認するには free コマンドを使用する。

スワップとは？

スワップはメモリが不足した場合にディスクの一部を仮想メモリとして使用する機能である。スワップが発生するとディスクアクセスが発生するため、メモリよりも遅いアクセス速度となる。そのため、スワップが発生するとパフォーマンスが低下する。

➤_ メモリとスワップの使用状況の確認

```
$ free -h
```

	total	used	free	shared	buff/cache	available
Mem:	1.9G	1.1G	100M	100M	700M	500M
Swap:	2.0G	100M	1.9G			

free コマンドの出力はメモリとスワップの使用状況を表示している。メモリの使用率が高い場合はプロセスの終了やメモリの増設などを行う必要がある。

11.4 プロセスの管理

実行中のプロセスを確認する時には ps コマンドを使用する。実行中の全てのプロセスを確認するには以下のコマンドを実行する。

➤_ プロセスの確認

```
$ ps aux
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	10:00	?	00:00:00	/sbin/init
root	2	0	0	10:00	?	00:00:00	[kthreadd]

ps コマンドの出力はプロセスの情報を表示している。プロセスの終了やプロセスの強制終了などを行う場合は kill コマンドを使用する。実際はこれ以上のプロセスが実行されているので less コマンドや grep コマンドなどを使用して絞り込むことが多い。

11.5 サービス管理

サービスは、OS 本体から切り離し可能な特定の役割や機能を持ったサブシステムのことをいう。サービスにはログ管理やネットワーク、Web サーバーや SSH サーバーなどの各種サーバープログラムが含まれる。サービスの起動、停止、再起動は systemctl コマンドを使用する。

➤_ systemctl コマンド

```
$ sudo systemctl {サブコマンド} {サービス名}
```

systemctl のサブコマンド

- start : サービスの起動
- stop : サービスの停止
- restart : サービスの再起動
- status : サービスの状態を確認
- enable : サービスの自動起動を有効にする
- disable : サービスの自動起動を無効にする

12 章 Web サーバーの構築

ここまでサーバー操作に関する基礎を学んできた。ここからは Web サーバーの構築を行う。今回は Apache を使用した Web サーバーの構築を行う。

i Apache について

Apache HTTP Server(以下、Apache) は、Apache ソフトウェア財団が提供するオープンソースの Web サーバーソフトウェアである。1995 年に開発が始まった。主な特徴として、拡張性が高く、多くのモジュールが提供されている。また、多くの OS に対応している。他にも静的コンテンツだけでなく動的な Web アプリケーションのホスティングにも対応している。

12.1 サーバーソフトウェアのインストール

12.1.1 Web サーバーと Web ブラウザ

Web サーバーは HTTP プロトコルを使用してクライアントに Web ページを提供するサーバーソフトウェアである。一方、Web ブラウザは HTTP プロトコルを使用して Web サーバーから Web ページを取得するクライアントソフトウェアである。Web ブラウザとして現在有名なのが、Google Chrome、Firefox、Safari、Microsoft Edge などがある。

12.2 Apache HTTP server のインストール

Apache HTTP server のインストールは以下のコマンドを実行することで行うことができる。

```
>_ terminal
$ sudo apt update $ sudo apt -y install apache2
```

インストール後、systemctl コマンドを使用して Apache HTTP server を起動・起動確認を行う。

```
>_ terminal
$ sudo systemctl start apache2
$ sudo systemctl enable apache2
$ sudo systemctl status apache2
```

以下のような表示がされれば Apache HTTP server の起動に成功している。

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-02-09 01:00:16 JST; 49s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 6419 (apache2)
    Tasks: 55 (limit: 4494)
   Memory: 4.9M
      CPU: 56ms
   CGroup: /system.slice/apache2.service
           └─6419 /usr/sbin/apache2 -k start
             └─6420 /usr/sbin/apache2 -k start
               └─6421 /usr/sbin/apache2 -k start

2月 09 01:00:16 tori-PC-GN17CJSA6 systemd[1]: Starting The Apache HTTP Server...
2月 09 01:00:16 tori-PC-GN17CJSA6 apachectl[6418]: AH00558: apache2: Could not reliably determine the server's fully qualified
2月 09 01:00:16 tori-PC-GN17CJSA6 systemd[1]: Started The Apache HTTP Server.
```

図 6 Apache HTTP server の起動確認

また、ブラウザを使用して `http://localhost/` にアクセスすることで Apache のデフォルトページが表示される。

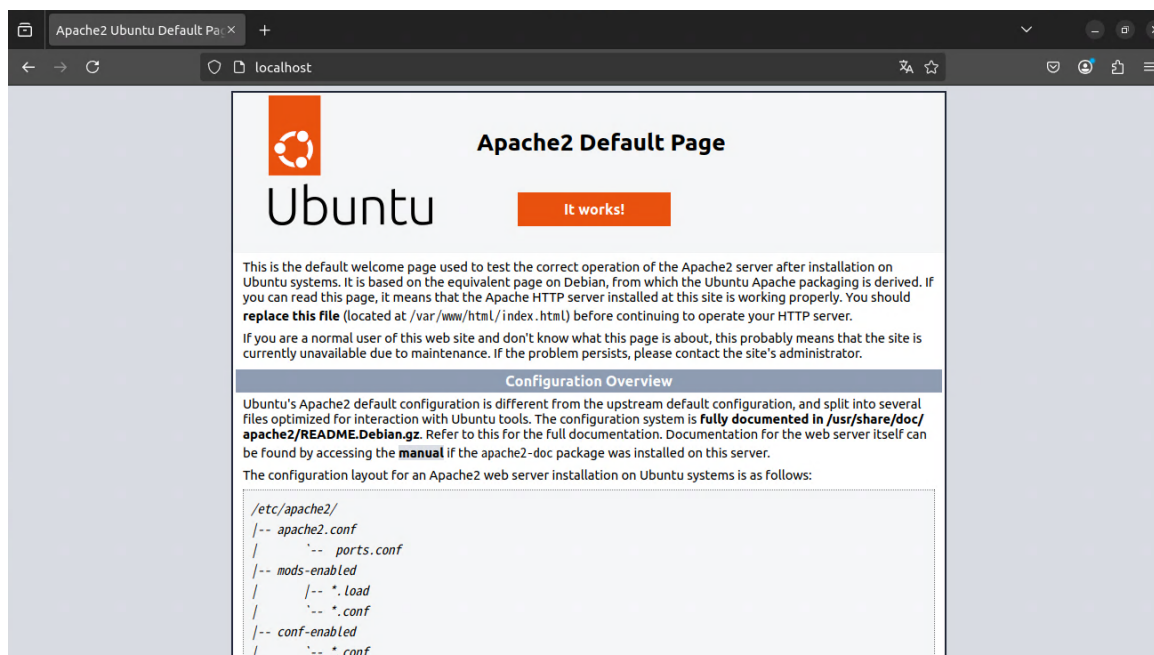


図 7 Apache のデフォルトページ

12.3 Apache の設定

Apache は Web サーバーとして当然ながら、様々な設定を行うことができる。それらを管理するのは「`/etc/apache2/apache2.conf`」や「`/etc/apache2/sites-available/000-default.conf`」などの設定ファイルである。ここではそれらに含まれる設定について解説する。

📄 apache2.conf ファイルについて

apache2.conf ファイルはサーバー全体に適用するためのグローバル設定ファイルである。このファイルには、サーバーの設定やモジュールの読み込みなどが記述されている。

📄 000-default.conf ファイルについて

000-default.conf ファイルはサイトごとの設定ファイルである。このファイルには、サイトの設定やドキュメントルートの設定などが記述されている。サイトごとの設定ファイルのため、このファイルを変更する、もしくは、新たに仮想ホストの設定ファイルを作成することで複数のサイトを運用することができる。

⚠️ 複数サイトの同時起動

Apache は前述の通り仮想ホストによりサーバー処理を行っている。そのため 1 つのサーバーから複数のサイトを運用することが可能である。しかし、ローカル上で複数サイトのアクセス処理のためには DNS サーバーによるポート転送が必要となるため今回は説明を省く。どうしても試してみたい場合は以下の参考サイトを参照するとよい。

77Lifework ベータ版, Apache を複数起動（複数インスタンス構成）する, <https://www.77-lifework.com/entry/apache-multi>

Apache の設定ファイルで記述可能な設定項目をディレクティブという。以下に主なディレクティブを示す。

- ServerRoot : 設定ファイルなどを配置するトップディレクトリ (`/etc/apache2`)
- Listen : Apache が受け付けるポート (80 番)

- User : Apache が実行されるユーザー (www-data)
- Group : Apache が実行されるグループ (www-data)
- ServerAdmin : Apache サーバーの管理者 (webmaster@localhost)
- DocumentRoot : Web ページを配置するディレクトリ (/var/www/html)
- DirectoryIndex : ディレクトリへのアクセス時に表示するファイル名 (index.html)

以降では一般的なサーバーにおける変更点として DocumentRoot、DirectoryIndex の変更を行う。また、Listen の変更を行う可能性もあるためそれについても説明する。

12.3.1 DocumentRoot と DirectoryIndex について

DocumentRoot は Web ページを配置するディレクトリを指定するディレクティブである。デフォルトでは /var/www/html が指定されている。ここで指定するディレクトリは「http://iサーバー IP:/」にアクセスした際に表示されるページが配置されるディレクトリである。このディレクトリの中にある DirectoryIndex で指定したファイルが「http://iサーバー IP:/」にアクセスした際に表示されるファイルとなる。DirectoryIndex はディレクトリへのアクセス時に表示するファイル名を指定するディレクティブである。デフォルトでは index.html が指定されている。このファイルが存在しない場合は index.php、index.cgi、index.pl、index.xhtml、index.htm の順に表示される。

12.3.2 仮想ホストの作成

Apache のインストール後は「000-default.conf」が仮想ホストとして設定されている。別の仮想ホストとして設定を行う。仮想ホストの設定は「/etc/apache2/sites-available/」ディレクトリ内に設定ファイルを作成する。ファイル名は任意であるが、一般的には「サイト名.conf」とする。ここでは「.conf」とする。

■ 仮想ホストの設定について

デフォルトで用意されているファイルを使用する場合は「000-default.conf」編集して使用してもよい。また、設定をゼロから記述するのが大変な場合は cp コマンドを使用する方法などもある。

➤ terminal: 仮想ホストの設定ファイルの作成

```
$ sudo touch /etc/apache2/sites-available/{ファイル名}.conf
```

作成したファイルに以下の最小構成の設定を記述する。DocumentRoot と DirectoryIndex は先ほど説明したとおりに設定を行う。

➤ /etc/apache2/sites-available/ファイル名.conf

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot {ディレクトリ名}
    DirectoryIndex {ファイル名}
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

■ APACHE_LOG_DIR について

APACHE_LOG_DIR は Apache のログファイルを保存するディレクトリを指定する環境変数である。デフォルトでは /var/log/apache2 に設定されている。この変数の変更には apache2.conf ファイルを編集する必要がある。

変更後、DocumentRoot に指定したディレクトリがない場合は作成し、DirectoryIndex を作成する。

>_ terminal: ディレクトリの作成

```
$ sudo mkdir {ディレクトリ名}
$ sudo touch {ディレクトリ名}/{ファイル名}
```

作成後、設定ファイルの有効化には a2ensite コマンドを使用し、仮想ホストの有効化を行う必要がある。

>_ terminal: 仮想ホストの有効化

```
$ sudo a2ensite {ファイル名}.conf
$ sudo systemctl restart apache2
```

Apache の再起動後 systemctl コマンドを使用して Apache の状態を確認する。(sudo systemctl status apache2)。起動が確認出来たら、ブラウザを使用して http://localhost/ にアクセスすることで新たに作成した仮想ホストのページが表示される。

⚠ 仮想ホストの無効化

仮想ホストの無効化を行わない場合、localhost にアクセスした際前述で作成したページにアクセスできない場合がある。また、サーバーへの負荷がかかってしまう。そのため、不要な仮想ホストの無効化を行うことが必要である。一方、仮想ホストを同時に立てる場合は別の手段となる。

>_ 仮想ホストの無効化

```
$ sudo a2dissite {ファイル名}.conf
$ sudo systemctl restart apache2
```



図 8 新たに作成した仮想ホストのページ

12.3.3 ポートの変更

ポートの変更には「Listen」ディレクティブを使用する。デフォルトでは 80 番ポートが指定されている。ポートの変更は以下のように行う。

```
>_ /etc/apache2/ports.conf
```

```
Listen 8080
```

変更後、Apache の再起動を行うことでポートの変更が適用される。

12.4 ファイアウォールの設定

SSH の設定の際さらっと出てきていたがここでもファイアウォールの設定を行う。

❗ ファイアウォールについて

ファイアウォールは、ネットワーク上で通信を制御するための装置やソフトウェアのことをいう。ファイアウォールは、外部からの不正アクセスや攻撃を防ぐために使用される。この機能は Windows や macOS などの OS にも搭載されている。今回は ufw を使用してファイアウォールの設定を行う。ufw は Uncomplicated Firewall の略で、比較的簡単に設定を行うことができる。

>_ ファイアウォールの有効化

```
$ sudo ufw enable
$ sudo ufw allow Apache
$ sudo ufw status
```

📌 ポート開放について

ファイアウォールの設定においては、ポートの開放が重要である。ポートの開放を行わない場合、外部からのアクセスが拒否されるため、Web サーバーにアクセスすることができない。ポートの開放は「sudo ufw allow ポート番号」コマンドを使用して行う。

>_ ポート開放:80 と 8080 の開放

```
$ sudo ufw allow 80/tcp
$ sudo ufw allow 8080/tcp
```

12.5 パスワード認証

通常はアクセスした人全員が Web ページの閲覧が可能である。しかし、特定のユーザーのみ閲覧可能なページの作成も可能となっている。それらの実現には一般的にユーザー名、パスワードによる認証が用いられる。その中で最も簡単な認証方法として基本認証がある。基本認証はユーザー名とパスワードを使用して認証を行う方法である。基本認証は Apache の設定ファイルに記述することで設定を行うことができる。

📌 apache2-utils について

認証を行うためのパッケージである apache2-utils は最新バージョンの場合インストールに含まれている。しかし、環境やバージョンなどによって含まれない場合もあるため念のためインストールをすることを推奨する。

➤_ apache2-utils のインストール

```
$ sudo apt install apache2-utils
```

12.5.1 htpasswd の作成

インストールした後、apache2-utils に含まれる htpasswd コマンドを使用してユーザー名とパスワードを記述したファイルを作成する。初回の登録時には「-c」オプションを使用する。

➤_ htpasswd の作成

```
$ sudo htpasswd -c /etc/apache2/.htpasswd {ユーザー名} # 初回のみ
$ sudo htpasswd /etc/apache2/.htpasswd {ユーザー名} # 2 回目以降
New password: {パスワード}
Re-type new password: {パスワード}
```

ユーザー登録の完了後、認証を利用する仮想ホストの設定ファイルに以下の設定を追加する。

➤_ /etc/apache2/sites-available/ファイル名.conf

```
<Directory {認証を利用したいディレクトリパス}>
  AuthType Basic
  AuthName "Restricted Content"
  AuthUserFile /etc/apache2/.htpasswd
  Require valid-user
</Directory>
```

設定後、Apache の再起動を行うことで設定が適用される。

⚠ 認証を利用したいディレクトリについて

指定したディレクトリ以下にあるもの全てに認証がかかるようになるため注意が必要である。

12.6 アクセスログ

アクセスログは Web サーバーにアクセスした際の情報を記録するログファイルである。これが読めるようになることでサーバーの状況を理解し、不正アクセスなどがあった際、おかしい部分として気づくことができる。Apache のアクセスログのデフォルトは/var/log/apache2/access.log である。アクセスログから読み取れるといいものをいかに挙げる。

- リクエストさいれた URL
- アクセス元の IP アドレス
- ユーザーエージェント
- ステータスコード
- リクエストの日時

アクセスログの設定は「/etc/apache2/apache2.conf」ファイルに記述されている。以下にアクセスログの設定例を示す。

➤ アクセスログの例

```
:::1 # アクセス元の IP アドレス (ローカルのため:::1)
- # リモートユーザー名
[10/Feb/2025:22:30:31 +0900] # リクエストの日時
"GET / HTTP/1.1" 200 2170 "-" "Mozilla/5.0 # リクエストの内容
(Windows NT 10.0; Win64; x64) # ユーザーエージェント
AppleWebKit/537.36 (KHTML, like Gecko) # ユーザーエージェント
Chrome/132.0.0.0 Safari/537.36" # アクセスブラウザ
```

アクセスログの例をまとめると次のようになる。

- アクセス元の IP アドレス (:::1)
- リモートユーザー名 (-)
- リクエストの日時 (10/Feb/2025:22:30:31 +0900)
- リクエストの内容 (GET / HTTP/1.1)
- ステータスコード (200)
- アクセスしたウェブブラウザ (Chrome)

❗ ステータスコード

いくつかのステータスコードを以下に示す。

- 200 : OK(リクエストが成功した)
- 401 : Unauthorized(認証が必要)
- 403 : Forbidden(アクセスが拒否された)
- 404 : Not Found(リクエストされたリソースが見つからない)
- 500 : Internal Server Error(サーバー内部のエラー)

12.7 エラーログ

サーバーを管理する上で重要なものとしてエラーログも挙げられる。これにはサーバー運用時に発生したエラーの情報が記録されている。Apache のエラーログのデフォルトは/var/log/apache2/error.log である。ただしエラーログについては様々なものがあるため、適宜対話型 AI などを用いて解決することがよいと思われる。そのためここでの説明は省略する。

13 章 参考リンク

Ubuntu について

RedHat, Linux とは, <https://www.redhat.com/ja/topics/linux/what-is-linux>

Canonical Ubuntu, Ubutu <https://jp.ubuntu.com>

エンジニアの入口, 【初心者でもわかる】Ubuntu のインストール方法まとめ, <https://eng-entrance.com/ubuntu-install>

金子邦彦研究室, Ubuntu 24.04 のインストールガイド <https://www.kkaneko.jp/tools/ubuntu/ubuntudesktop.html>

Quita, @yuta_931214(yuta kimura), 【初学者向け】Linux のファイル構造についてまとめてみた, https://qiita.com/yuta_931214/items/7cd5df422e05cf07c2f8

使用 PC

リコレ!, Gateway NE132 NE132-F14P [Windows 10], <https://used.sofmap.com/r/item/2133024911019>

便利機能インストール

Quita, @takuya66520126A(takuya), ubuntu で日本語入力に変更する方法, <https://qiita.com/takuya66520126/items/8bb760bf99c4e25364e3>

Quita, @tommy_g, Ubuntu に Vim をインストールする https://qiita.com/tommy_g/items/3ad4b26ccb4893f2760e

Quita, @to-fmak(Wenzhang), Linux の面白いコマンド 9 選 <https://qiita.com/to-fmak/items/ea345839f394db781bd0>

Zenn, koki, cmatrix コマンドでターミナルに文字を降らせる https://zenn.dev/kou_pg_0131/articles/cmatrix-introduction

ネットワーク関連

Quita, @pe-ta(ペータ), ifconfig の出力結果に書いてあること <https://qiita.com/pe-ta/items/aff8db72530c6baa11b2>

Quita, @kooohai(Kohei Yamada), 静的ルーティングの設定 - Linux <https://qiita.com/kooohai/items/b0931ae210911cc52adc>

ユーザー管理

Quita, @3062_in_zamud(3062.zamud), [chown] ファイルやディレクトリの所有者を変更する方法 https://qiita.com/3062_in_zamud/items/d93770016e7d931a5983

Apache

Quita, @Apache <https://qiita.com/tags/apache>

77Lifework ベータ版, Apache を複数起動 (複数インスタンスを構成) する, <https://www.77-lifework.com/entry/apache-multi>

JavaDrive, Listen ディレクティブ: リクエストを受け付けるポート番号, <https://www.javadrive.jp/apache/ini/index3.html> Apache, Apache HTTP サーバ バージョン 2.2 バーチャルホストの例, <https://httpd.apache.org/docs/2.2/ja/vhosts/examples.html>

InfoAcademy, Apache のログの場所は? ログの場所を設定する方法について詳しく解説【Linux】, <https://engineer-ninaritai.com/apache-log-directory/>

14 章 終わりに

サーバー構築お疲れさまでした。ここまで Linux の基礎的な部分から少し踏み込んだ部分も含めて解説をしました。筆者もそうですが 1 度するだけでは覚えていません。実際に同じように立てたい時が来た時にまた見返せばいいのです。

反省として画像をもっと入れるべきだと思っています。改訂版が出た暁にはさらに見やすいものを心がけるつもりです。

それでは今後とも楽しい Linux ライフを！