

# 【Linux】基礎

## 目次

- 全般
- UNIXコマンド
- Bashコマンド・Bashスクリプト
- Debian系限定のコマンド
- Red Hat系限定のコマンド
- Slackware系限定のコマンド
- サービス、ミドルウェア
  - サービス全般
  - SSHサーバ
  - Webサーバ
  - ファイルサーバ
  - cron
- ローカル開発環境の構築 – Windowsの場合

## 全般

### ■Linux ディストリビューション

- ▶ ※ ディストリビューション (distribution) は、流通、配布を意味する。本来「Linux」と言えば Linux Kernel を指す。しかし、カーネルはOSとしての基本的動作を実現しているものであり、それ単体では使いやすいものではない。このため、ユーザーインターフェイスやネットワークを制御する User space Packages、その実行や構築を支援するLibrariesなどを追加し Linux Kernel を中核として、パッケージ化したものが Linux ディストリビューションである。Linux Kernel だけでなく、Linux ディストリビューション全体のことを指して「（広義の）Linux」と呼ぶのも一般的になっている。
- ▶ ☆ なぜLinuxディストリビューションは豊富なのか。
- ▶ ☆ Debian系（Ubuntu、Raspberry Pi OS、MX Linux、Linux Mint など）
- ▶ ☆ Red Hat系（Fedora、Red Hat Enterprise Linux、Cent OS、Oracle Linux）
- ▶ ☆ Slackware系（Slackware、openSUSE など）
- ▶ ☆ 独立系（Manjaro など）
- ▶ ※ Linuxでは統一された1つのツリーでファイルを管理する。

# 【Linux】基礎

## 目次

- 全般
- UNIXコマンド
- Bashコマンド・Bashスクリプト
- Debian系限定のコマンド
- Red Hat系限定のコマンド
- Slackware系限定のコマンド
- サービス、ミドルウェア
  - サービス全般
  - SSHサーバ
  - Webサーバ
  - ファイルサーバ
  - cron
- ローカル開発環境の構築 – Windowsの場合

## 全般

### ■Linux ディストリビューション

- ▶ ※ ディストリビューション (distribution) は、流通、配布を意味する。本来「Linux」と言えば Linux Kernel を指す。しかし、カーネルはOSとしての基本的動作を実現しているものであり、それ単体では使いやすいものではない。このため、ユーザーインターフェイスやネットワークを制御する User space Packages、その実行や構築を支援するLibrariesなどを追加し Linux Kernel を中核として、パッケージ化したものが Linux ディストリビューションである。Linux Kernel だけでなく、Linux ディストリビューション全体のことを指して「（広義の）Linux」と呼ぶのも一般的になっている。
- ▶ ☆ なぜLinuxディストリビューションは豊富なのか。
- ▶ ☆ Debian系（Ubuntu、Raspberry Pi OS、MX Linux、Linux Mint など）
- ▶ ☆ Red Hat系（Fedora、Red Hat Enterprise Linux、Cent OS、Oracle Linux）
- ▶ ☆ Slackware系（Slackware、openSUSE など）
- ▶ ☆ 独立系（Manjaro など）
- ▶ ※ Linuxでは統一された1つのツリーでファイルを管理する。

- ▶ ※ <https://www.server-world.info/> にサーバ構築に役立つ情報がある。

## ■システム

- ▶ ※ systemd はLinuxカーネルによって最初に起動される**プログラム**（ゆえにそのプロセスIDは1）。サービスやデーモンの起動などを管理するプログラムであり、全デーモンを管理するデーモンである（全デーモンの親）。
- ▶ ※ systemctl は、systemd をコントロールする**コマンド**。サービスの起動・停止や自動起動の設定、サービス状態の確認などができる。

- ▶ カーネル情報
- ▶ ディストリビューション情報
- ▶ アーキテクチャの確認

## ■ネットワーク

- ▶ プライベートIPアドレスを調べる
- ▶ あるホストの

■日時

- ▶ タイムゾーンを日本に

## ■ソフトウェア

- ▶ ☆ Gitをインストール
- ▶ ☆ homebrewをインストール
- ▶ ☆ viをインストール

# UNIXコマンド

## ■シエルのショートカットキー

- ▶ コマンドを実行せずに新しいプロンプトに移る
- ▶ 今書いているコマンドを消す
- ▶ 上に書いてきたのを消す
- ▶ コマンドやファ、ディ名を途中まで書いて補完
- ▶ 以前のコマンドを使う

## ■ディレクトリの構成

- ▶ ☆ 各ディレクトリの説明
- ▶ ☆ 絶対PATH と 相対PATH

- ▶ ※ <https://www.server-world.info/> にサーバ構築に役立つ情報がある。

## ■システム

- ▶ ※ systemd はLinuxカーネルによって最初に起動される**プログラム**（ゆえにそのプロセスIDは1）。サービスやデーモンの起動などを管理するプログラムであり、全デーモンを管理するデーモンである（全デーモンの親）。
- ▶ ※ systemctl は、systemd をコントロールする**コマンド**。サービスの起動・停止や自動起動の設定、サービス状態の確認などができる。

- ▶ カーネル情報                   \$ cat /proc/version
- ▶ ディストリビューション情報   \$ cat /etc/os-release
- ▶ アーキテクチャの確認       \$ uname -m   か   \$ arch   か   \$ gcc -v

## ■ネットワーク

- ▶ プライベートIPアドレスを調べる    `$ ip a` か `$ ip addr` か `$ ip address` か `$ ifconfig`
- ▶ あるホストの    `"`    `$ host hostname`

■日時

- ▶ タイムゾーンを日本に `$ sudo timedatectl set-timezone Asia/Tokyo`

## ■ソフトウェア

- ▶ ☆ Gitをインストール
- ▶ ☆ homebrewをインストール
- ▶ ☆ viをインストール

# UNIXコマンド

## ■シエルのショートカットキー

- ▶ コマンドを実行せずに新しいプロンプトに移る {Ctrl}+{C}
- ▶ 今書いているコマンドを消す {Ctrl}+{U}
- ▶ 上に書いてきたのを消す {Ctrl}+{L} ※ `$ clear` でもいい
- ▶ コマンドやファ、ディ名を途中まで書いて補完 {Tab}
- ▶ 以前のコマンドを使う {↑} や {↓}

## ■ディレクトリの構成

- ▶ ☆ 各ディレクトリの説明
- ▶ ☆ 絶対PATH と 相対PATH

|           |  |
|-----------|--|
| ■用語       |  |
| ▶ ストリーム   |  |
| ▶ ディスクリプタ |  |

|              |  |
|--------------|--|
| ■基本操作        |  |
| ▶ カレントディを表示  |  |
| ▶ 他のディに移る    |  |
| ▶ 出力         |  |
| ▶ 現在のユーザ名を表示 |  |
| ▶ 文字列をトリムし出力 |  |

|   |  |
|---|--|
| ■コマンド実行全般   |  |
| ▶ コマの使い方を調べる  |  |
| ▶ 管理者権限で実行  |  |
| ▶ 出力せずファを上書き  |  |
| ▶ 出力せずファに追記   |  |
| ▶ コマの引数にファを   |  |
| ▶ 出力せず別コマ引数に  |  |
| ▶ ※ コマンドは実行すると、終了ステータス（またはリターンコード）と呼ばれる数値を返す。成功の場合 <code>0</code> 、失敗の場合 <code>1</code> またはその他の数値（ <code>0</code> を除く）。なお、直前に実行したコマンドの終了ステータスの値は特殊変数 <code>\$?</code> に格納されている。 |  |
| ▶ コマを立て続けに実行  |  |
| ▶ <code>"</code> （失敗したら次のコマは実行しない）  |  |
| ▶ <code>"</code> （失敗した場合のみ次のコマを実行）  |  |
| ▶ ※ <code>{0,0,0}</code> <code>{0..0}</code> でfor文のようにできる（ブレース展開）。  |  |
| ▶ 長いコマで改行したい  |  |
| ▶ ログを表示させない   |  |
| ▶ ※ オプションは複数付けられる。  |  |
| ▶ あるコマが出力するか  |  |

|                  |  |
|------------------|--|
| ■コマンドの履歴を活用      |  |
| ▶ コマの履歴を見る       |  |
| ▶ 履歴上の $n$ 番目を実行 |  |

|           |                                |
|-----------|--------------------------------|
| ■用語       |                                |
| ▶ ストリーム   | 何らかの入出力において、データ (バイト列) が流れる通り道 |
| ▶ ディスクリプタ | 各プロセス内でストリームを区別する識別子。実体は整数値。   |

|              |   |
|--------------|---|
| ■基本操作        |   |
| ▶ カレントディを表示  | <code>\$ pwd</code> ※ <code>print working directory</code> の略                   |
| ▶ 他のディに移る    | <code>\$ cd dirPath</code> ※ <code>change directory</code> の略                   |
| ▶ 出力         | <code>\$ echo 'str'</code>  |
| ▶ 現在のユーザ名を表示 | <code>\$ whoami</code> か <code>\$ echo \$USER</code>                            |
| ▶ 文字列をトリムし出力 | <code>\$ sed -r 's/^[[:space:]]* [[[:space:]]*\$//g' &lt;&lt;&lt; \$hoge</code> |

|   |   |
|---|---|
| ■コマンド実行全般   |   |
| ▶ コマの使い方を調べる  | <code>\$ man com</code> か <code>\$ help com</code> か <code>\$ com --help</code>         |
| ▶ 管理者権限で実行  | <code>\$ sudo commands</code> ※ ※ <code>!</code> も使える。                                  |
| ▶ 出力せずファを上書き  | <code>\$ commands &gt; filePath</code> : リダイレクション                                       |
| ▶ 出力せずファに追記   | <code>\$ commands &gt;&gt; filePath</code> : <code>"</code>                             |
| ▶ コマの引数にファを   | <code>\$ commands &lt; filePath</code> : <code>"</code>                                 |
| ▶ 出力せず別コマ引数に  | <code>\$ commands   anotherCommands</code> : パイプ  |
| ▶ ※ コマンドは実行すると、終了ステータス（またはリターンコード）と呼ばれる数値を返す。成功の場合 <code>0</code> 、失敗の場合 <code>1</code> またはその他の数値（ <code>0</code> を除く）。なお、直前に実行したコマンドの終了ステータスの値は特殊変数 <code>\$?</code> に格納されている。 |   |
| ▶ コマを立て続けに実行  | <code>\$ commands1 ; commands2 ; ...</code>   |
| ▶ <code>"</code> （失敗したら次のコマは実行しない）  | <code>\$ commands1 &amp;&amp; commands2 &amp;&amp; ...</code>                           |
| ▶ <code>"</code> （失敗した場合のみ次のコマを実行）  | <code>\$ commands1    commands2    ...</code>   |
| ▶ ※ <code>{0,0,0}</code> <code>{0..0}</code> でfor文のようにできる（ブレース展開）。  |   |
| ▶ 長いコマで改行したい  | <code>\</code> を入力後 Enter キーで改行可能   |
| ▶ ログを表示させない   | <code>\$ commands &gt; /dev/null 2&gt;&amp;1</code> ※インストール時などに                         |
| ▶ ※ オプションは複数付けられる。  |   |
| ▶ あるコマが出力するか  | <code>\$ [ -z \$(commands) ]; echo \$?</code> ※ <code>1</code> は出力アリ <code>0</code> はナシ |

|                  |                         |
|------------------|-------------------------|
| ■コマンドの履歴を活用      |                         |
| ▶ コマの履歴を見る       | <code>\$ history</code> |
| ▶ 履歴上の $n$ 番目を実行 | <code>\$ !n</code>      |

|                        |  |
|------------------------|--|
| ▶ 直前のコマを実行             |  |
| ▶ n個前のコマを実行            |  |
| ▶ 直前コマの最終引数            |  |
| ▶ <i>str</i> で始まる直近のコマ |  |
| ▶ コマを検索                |  |

■ディレクトリの情報を確認

|            |  |
|------------|--|
| ▶ ディの情報を確認 |  |
|------------|--|

■ディレクトリの中身を確認

|               |  |
|---------------|--|
| ▶ ディの中身を確認    |  |
| ▶ 隠しファ含めて "   |  |
| ▶ タイプ識別子付きで " |  |
| ▶ 1件1行で "     |  |
| ▶ 権限の情報付きで "  |  |
| ▶ 再帰的に "      |  |

■ファイルやディレクトリの操作

|              |  |
|--------------|--|
| ▶ ファの更新日時を更新 |  |
| ▶ ファを新規作成    |  |
| ▶ ディを新規作成    |  |
| ▶ リンクを作成     |  |
| ▶ ファを複製      |  |
| ▶ ディを複製      |  |
| ▶ 移動         |  |
| ▶ 名前の変更      |  |
| ▶ ファを削除      |  |
| ▶ ディの中身を空に   |  |
| ▶ 中身が空のディを削除 |  |
| ▶ 中身ごとディを削除  |  |

■ファイルの情報を確認

|            |  |
|------------|--|
| ▶ 権限を確認    |  |
| ▶ リンクのリンク先 |  |

|                        |   |
|------------------------|---|
| ▶ 直前のコマを実行             | \$ !!   |
| ▶ n個前のコマを実行            | \$ !-n  |
| ▶ 直前コマの最終引数            | !\$ で使える  |
| ▶ <i>str</i> で始まる直近のコマ | \$ ! <i>str</i> ※ <i>!str:p</i> とすれば実行はせず、その後 <i>!!</i> で実行       |
| ▶ コマを検索                | {Ctrl}+{R} → <i>str</i> 入力 → {Ctrl}+{R}で次候補へ → {Enter}か{Ctrl}+{C} |

■ディレクトリの情報を確認

|            |                          |
|------------|--------------------------|
| ▶ ディの情報を確認 | \$ ls -ld <i>dirPath</i> |
|------------|--------------------------|

■ディレクトリの中身を確認

|               |   |
|---------------|---|
| ▶ ディの中身を確認    | \$ ls <i>dirPath</i> : list ※ <i>dirPath</i> 省略ならカレディ       |
| ▶ 隠しファ含めて "   | \$ ls -a <i>dirPath</i>                                     |
| ▶ タイプ識別子付きで " | \$ ls -F <i>dirPath</i> ※ <i> * / = &gt; @  </i> のどれかが末尾につく |
| ▶ 1件1行で "     | \$ ls -l <i>dirPath</i>                                     |
| ▶ 権限の情報付きで "  | \$ ls -l <i>dirPath</i>                                     |
| ▶ 再帰的に "      | \$ ls -R <i>dirPath</i>                                     |

■ファイルやディレクトリの操作

|              |   |
|--------------|---|
| ▶ ファの更新日時を更新 | \$ touch <i>filePath</i> ※ファイルがない場合自動作成                 |
| ▶ ファを新規作成    | \$ touch <i>filePath</i> ※深いファならその祖先ディは既存の必要            |
| ▶ ディを新規作成    | \$ mkdir <i>childDirName</i> \$ mkdir -p <i>dirPath</i> |
| ▶ リンクを作成     | \$ ln -s <i>dirPath name</i> ※シンボリックリンクという              |
| ▶ ファを複製      | \$ cp <i>filePath newFilePath</i>                       |
| ▶ ディを複製      | \$ cp -r <i>dirPath newDirPath</i>                      |
| ▶ 移動         | \$ mv <i>path destinedDirPath</i>                       |
| ▶ 名前の変更      | \$ mv <i>path newPath</i>                               |
| ▶ ファを削除      | \$ rm <i>filePath</i>                                   |
| ▶ ディの中身を空に   | \$ rm -r <i>dirPath/*</i> ※とても危険なので要注意                  |
| ▶ 中身が空のディを削除 | \$ rmdir <i>dirPath</i>                                 |
| ▶ 中身ごとディを削除  | \$ rm -r <i>dirPath</i> ※とても危険なので要注意                    |


■ファイルの情報を確認

|            |                             |
|------------|-----------------------------|
| ▶ 権限を確認    | \$ ls -l <i>filePath</i>    |
| ▶ リンクのリンク先 | \$ readlink <i>filePath</i> |

■ファイルの中身を確認

|                      |  |
|----------------------|--|
| ▶ 一気にすべて確認           |  |
| ▶ ちょっとずつ確認<br>(ページャ) |  |
| ▶ 行数や単語数を確認          |  |
| ▶ 先頭・末尾の幾行か          |  |
| ▶ 随時追記されるたびに         |  |
| ▶ <i>n</i> 行目のみ      |  |
| ▶ 検索して該当行を抽出         |  |
| ▶ 文字列を置換して出力         |  |

■ファイルを編集 (viエディター)

|  |  |
|--|--|
| ▶ ※ ファイルを上書きや追記するくらいなら、viエディターを使わずリダイレクションを使えばいい。  |  |
| ▶ 起動   |  |
| ▶ ※ 起動直後はコマンドモードで、左下に  と出ている。 |  |
| ▶ 編集モードに   |  |
| ▶ コマンドモードに戻る   |  |
| ▶ 終了 (〃)   |  |
| ▶ 保存して終了 (〃)   |  |
| ▶ 保存せずに終了 (〃)  |  |

■ファイル実行

|                 |  |
|-----------------|--|
| ▶ ファイル実行        |  |
| ▶ ファ名だけで実行したいなら |  |
| ▶ コマのファの場所      |  |

■ファイルやディレクトリの検索



|            |  |
|------------|--|
| ▶ ファやディの検索 |  |
| ▶ ファだけ検索   |  |
| ▶ ディだけ検索   |  |

■ユーザやグループの管理



■ファイルの中身を確認

|                      |  |
|----------------------|--|
| ▶ 一気にすべて確認           | <code>\$ cat filePath</code>   |
| ▶ ちょっとずつ確認<br>(ページャ) | <ul style="list-style-type: none"><li>・ <code>\$ more filePath</code> ※{Space} で次頁へ、{Q} で終了。</li><li>・ <code>\$ less filePath</code> ※{t}{l}で行スクロール、{Q} で終了。</li></ul> |
| ▶ 行数や単語数を確認          | <code>\$ wc filePath</code>  |
| ▶ 先頭・末尾の幾行か          | <code>\$ head -n lineCount filePath</code> ・ <code>\$ tail -n lineCount filePath</code>  |
| ▶ 随時追記されるたびに         | <code>\$ tail -f filePath</code> ※追記されるたびに追加で表示する  |
| ▶ <i>n</i> 行目のみ      | <code>\$ sed -n np filePath</code> か <code>\$ cat filePath   head -n   tail -1</code>  |
| ▶ 検索して該当行を抽出         | <code>\$ grep 'str' filePath</code>  |
| ▶ 文字列を置換して出力         | <code>\$ sed s/what/replacement/g filePath</code> ※見つければ全て置換   |

■ファイルを編集 (viエディター)

|  |   |
|--|---|
| ▶ ※ ファイルを上書きや追記するくらいなら、viエディターを使わずリダイレクションを使えばいい。  |   |
| ▶ 起動   | <code>\$ vi filePath</code> ※    ※未存でも可   |
| ▶ ※ 起動直後はコマンドモードで、左下に  と出ている。 |   |
| ▶ 編集モードに   | { }    ※左下に  と出る |
| ▶ コマンドモードに戻る   | {Esc}   |
| ▶ 終了 (〃)   | :q  |
| ▶ 保存して終了 (〃)   | :wq   |
| ▶ 保存せずに終了 (〃)  | :q!   |

■ファイル実行

|                 |  |
|-----------------|--|
| ▶ ファイル実行        | <ul style="list-style-type: none"><li>・ <code>\$ filePath</code>※    ※カレディ中のファは  に続ける必要あり</li><li>・ <code>\$ \$SHELL filePath</code>※    ※  に続ける必要なし</li></ul> |
| ▶ ファ名だけで実行したいなら | <code>\$ export PATH=dirAbsPath:\$PATH</code> でパスを通しておく  |
| ▶ コマのファの場所      | <code>\$ which commandName</code>  |

■ファイルやディレクトリの検索

|            |  |
|------------|--|
| ▶ ファやディの検索 | <code>\$ find dirPath -name 'str'※</code> ※ワイルドカード可  |
| ▶ ファだけ検索   | <code>\$ find dirPath -name 'str' type -f</code> ※ 〃 |
| ▶ ディだけ検索   | <code>\$ find dirPath -name 'str' type -d</code> ※ 〃 |

■ユーザやグループの管理

|             |  |
|-------------|--|
| ▶ 全ユーザを表示   |  |
| ▶ 全グループを表示  |  |
| ▶ ユの属するグを確認 |  |

|                              |  |
|------------------------------|--|
| ▶ ユーザを新規作成                   |  |
| ▶ ユーザを削除                     |  |
| ▶ ユのパスワード変更                  |  |
| ▶ グループを新規作成                  |  |
| ▶ グループを削除                    |  |
| ▶ ユをグに追加                     |  |
| ▶ ユをグから排除                    |  |
| ▶ ☆ ユーザのデフォルトシェル（ログインシェル）を変更 |  |

## ■アクセス権限（パーミッション）

|                      |  |
|----------------------|--|
| ▶ ファやディのパーミ、所有者などを確認 |  |
| ▶ ファやディのパーミを変更       |  |
| ▶ ファやディの所有者を変更       |  |

## ■システム

|             |  |
|-------------|--|
| ▶ 環境変数の一覧   |  |
| ▶ 環境変数の編集   |  |
| ▶ 環境変数の値を出力 |  |
| ▶ 再起動       |  |
| ▶ 実行中プロセス一覧 |  |

## ■ネットワーク

|                       |  |
|-----------------------|--|
| ▶ HTTPアクセスをしてコンテンツを取得 |  |
| ▶ リダイレクトも処理して "       |  |
| ▶ 自己署名証明書を受け入れて "     |  |
| ▶ オンラインファイルをダウンロード    |  |
| ▶ グローバルIPアドレス         |  |

|             |                            |
|-------------|----------------------------|
| ▶ 全ユーザを表示   | \$ cat /etc/passwd         |
| ▶ 全グループを表示  | \$ cat /etc/group          |
| ▶ ユの属するグを確認 | \$ groups <i>user1</i> ... |

|                              |  |
|------------------------------|--|
| ▶ ユーザを新規作成                   | \$ sudo useradd <i>userName</i> ※ <code>-m</code> でホームディも自動作成      |
| ▶ ユーザを削除                     | \$ sudo userdel <i>user</i> ※ <code>-r</code> でホームディも削除            |
| ▶ ユのパスワード変更                  | 現ユーザなら \$ passwd 他人なら \$ sudo passwd <i>user</i>                   |
| ▶ グループを新規作成                  | \$ sudo groupadd <i>groupName</i>                                  |
| ▶ グループを削除                    | \$ sudo groupdel <i>group</i>                                      |
| ▶ ユをグに追加                     | \$ gpasswd -a <i>user group</i> か \$ usermod -aG <i>group user</i> |
| ▶ ユをグから排除                    | \$ gpasswd -d <i>user group</i>                                    |
| ▶ ☆ ユーザのデフォルトシェル（ログインシェル）を変更 |  |

## ■アクセス権限（パーミッション）

|                      |  |
|----------------------|--|
| ▶ ファやディのパーミ、所有者などを確認 | ファ： \$ ls -l <i>filePath</i><br>ディ： \$ ls -ld <i>dirPath</i> ※ <i>dirPath</i> 省略ならカレディ   |
| ▶ ファやディのパーミを変更       | ファ： \$ chmod <i>mode filePath</i><br>ディ： \$ chmod -R <i>mode dirPath</i>   |
| ▶ ファやディの所有者を変更       | ファ： \$ sudo <b>chown</b> <i>user:group filePath</i> ※ <code>.</code> <b>じゃなく</b> <code>:</code><br>ディ： \$ sudo <b>chown</b> -R <i>user:group dirPath</i> ※ " |

## ■システム

|             |                   |
|-------------|-------------------|
| ▶ 環境変数の一覧   | \$ env            |
| ▶ 環境変数の編集   | \$ export 環境変数名=値 |
| ▶ 環境変数の値を出力 | \$ echo \$環境変数名   |
| ▶ 再起動       | \$ sudo reboot    |
| ▶ 実行中プロセス一覧 | \$ ps -x          |

## ■ネットワーク

|                       |   |
|-----------------------|---|
| ▶ HTTPアクセスをしてコンテンツを取得 | \$ curl <i>URL</i>                                      |
| ▶ リダイレクトも処理して "       | \$ curl -L <i>URL</i> ※ <code>\$ wget URL</code> と同じっぽい |
| ▶ 自己署名証明書を受け入れて "     | \$ curl -k <i>URL</i> ※ <code>--insecure</code> でも可。    |
| ▶ オンラインファイルをダウンロード    | \$ curl -L -o <i>fileName URL</i>                       |
| ▶ グローバルIPアドレス         | \$ curl inet-ip.info など                                 |

|                   |  |
|-------------------|--|
| ▶ プライベートIPアドレス    |  |
| ▶ 別ホストまでのネットワーク経路 |  |
| ▶ 別ホストにパケットを飛ばす   |  |
| ▶ ホスト名            |  |

## ■別サーバにSSH接続

### 鍵作成

|                 |  |
|-----------------|--|
| ▶ 公開鍵／秘密鍵のペアを作る |  |
|-----------------|--|

### 普通のSSH接続

|                  |  |
|------------------|--|
| ▶ サーバに接続         |  |
| ▶ ログアウト (サーバと切断) |  |

### サーバとのファイルの授受 (SFTP)

|            |  |
|------------|--|
| ▶ SFTPを起動  |  |
| ▶ ファをD L ・ |  |
| ディをD L     |  |
| ▶ ファをU P ・ |  |
| ディをU P     |  |
| ▶ SFTPを終了  |  |

### サーバとのファイルの授受 (SCP)

|            |  |
|------------|--|
| ▶ ファをD L ・ |  |
| ディをD L     |  |
| ▶ ファをU P ・ |  |
| ディをU P     |  |

## ■日時

|            |  |
|------------|--|
| ▶ 日時       |  |
| ▶ 今月のカレンダー |  |

## Bashコマンド・Bashスクリプト

### ■はじめに

- ▶ ※ スクリプトの場合は冒頭にシェバンを書こう。

|                   |   |
|-------------------|---|
| ▶ プライベートIPアドレス    | \$ ifconfig   |
| ▶ 別ホストまでのネットワーク経路 | \$ sudo traceroute <i>targetIP</i> か \$ tracepath <i>targetIP</i> |
| ▶ 別ホストにパケットを飛ばす   | \$ ping <i>targetIP</i>   |
| ▶ ホスト名            | \$ hostname   |

## ■別サーバにSSH接続

### 鍵作成

- ▶ 公開鍵／秘密鍵のペアを作る \$ ssh-keygen -t 暗号化方式 -b ビット数 -C "コメント"

### 普通のSSH接続

|                  |                                   |
|------------------|-----------------------------------|
| ▶ サーバに接続         | \$ ssh <i>serverUser@serverIP</i> |
| ▶ ログアウト (サーバと切断) | exit (か {Ctrl} + {D} ?)           |

### サーバとのファイルの授受 (SFTP)

|            |   |
|------------|---|
| ▶ SFTPを起動  | \$ sftp <i>remoteUser@remoteIP</i>                                    |
| ▶ ファをD L ・ | \$ get <i>filePath</i> ※ <sup>1</sup> ・                               |
| ディをD L     | \$ get -r <i>dirPath</i> ※ <sup>1</sup> ※ <sup>1</sup> 相手サーバ上のパス      |
| ▶ ファをU P ・ | \$ put <i>filePath</i> ※ <sup>1</sup> ・                               |
| ディをU P     | \$ put -r <i>dirPath</i> ※ <sup>1</sup> ※ <sup>1</sup> 自身のWindows上のパス |
| ▶ SFTPを終了  | \$ exit quit bye {Ctrl} + {C} のいずれか                                   |

### サーバとのファイルの授受 (SCP)

|            |  |
|------------|--|
| ▶ ファをD L ・ | \$ scp <i>remoteUser@remoteIP:remoteSrcFilePath localDstPath</i> ・ |
| ディをD L     | \$ scp -r <i>remoteUser@remoteIP:remoteSrcDirPath localDstPath</i> |
| ▶ ファをU P ・ | \$ scp <i>localSrcFilePath remoteUser@remoteIP:remoteDstPath</i> ・ |
| ディをU P     | \$ scp -r <i>localSrcDirPath remoteUser@remoteIP:remoteDstPath</i> |

## ■日時

|            |         |
|------------|---------|
| ▶ 日時       | \$ date |
| ▶ 今月のカレンダー | \$ cal  |

## Bashコマンド・Bashスクリプト

### ■はじめに

- ▶ ※ スクリプトの場合は冒頭にシェバンを書こう。



- ▶ ※ Bashにたいしてユーザごとの個別の設定を行いたいなら `~/.bash_profile` や `~/.bashrc`、あるいは `~/.bash_logout` などにコマンドを書き込もう。
- ▶ ※ コチラに詳しい（英語）。

### ■コマンド実行全般

- ▶ バックグラウンド実行

### ■標準入出力

- ▶ コマンド引数の个数
- ▶ コマンド引数を参照

### ■スクリプト

- ▶ スクリプト自身のパス
- ▶ エラーしたら終了に

### ■変数

- ▶ 変数に入れた文字列の長さ

### ■関数

- ▶ 関数を定義
- ▶ 終了ステータスを返す
- ▶ ※ `return` コマンドがない場合、関数内で最後に処理されたコマンドの終了ステータスが関数の終了ステータスとなる。

## Debian系限定のコマンド

### ■パッケージ

- ▶ 全パッケを更新
- ▶ パッケをインストール
- ▶ ※ パッケージのインストールに失敗する場合、パッケージリストを更新しよう。
- ▶ パッケをアンインストール
- ▶ ※ ネットで検索していて `apt-get` と出てきたら → `apt` を代わりに使う

### ■ユーザやグループの管理

- ▶ ヌにsudo権限を付与

### ■ファイルを編集（nanoエディター）

- ▶ ※ Bashにたいしてユーザごとの個別の設定を行いたいなら `~/.bash_profile` や `~/.bashrc`、あるいは `~/.bash_logout` などにコマンドを書き込もう。
- ▶ ※ コチラに詳しい（英語）。

### ■コマンド実行全般

- ▶ バックグラウンド実行 `$ commands &`

### ■標準入出力

- ▶ コマンド引数の个数 `$#`
- ▶ コマンド引数を参照 `$n` ※ `$ /bin/bash oo.sh 第1引数 第2引数 ...`

### ■スクリプト

- ▶ スクリプト自身のパス `$0`
- ▶ エラーしたら終了に `set -eu` を先に書いておく

### ■変数

- ▶ 変数に入れた文字列の長さ `${#hoge}`

### ■関数

- ▶ 関数を定義 `hoge_hoge() { .. }`
- ▶ 終了ステータスを返す `return status`
- ▶ ※ `return` コマンドがない場合、関数内で最後に処理されたコマンドの終了ステータスが関数の終了ステータスとなる。

## Debian系限定のコマンド

### ■パッケージ

- ▶ 全パッケを更新 `$ sudo apt update` ※定期的に行おう
- ▶ パッケをインストール `$ sudo apt install pack1 ...` ※Yesが面倒なら `-y` を
- ▶ ※ パッケージのインストールに失敗する場合、パッケージリストを更新しよう。
- ▶ パッケをアンインストール `$ sudo apt remove pack`
- ▶ ※ ネットで検索していて `apt-get` と出てきたら → `apt` を代わりに使う


### ■ユーザやグループの管理

- ▶ ヌにsudo権限を付与 `$ sudo gpasswd -a user sudo`

### ■ファイルを編集（nanoエディター）



▶ ※ nano はとりあえず Ubuntu には標準で入っているらしい。

- |           |  |
|-----------|--|
| ▶ 起動      |  |
| ▶ 保存して終了  |  |
| ▶ 保存せずに終了 |  |
- ▶ ※ 起動するとショートカットキー一覧が下に表示される。そこにおいて、 は Ctrl キーを意味する。

## Red Hat系限定のコマンド

### ■パッケージ

- ▶ ※ Red Hat系では RPM Package Manager というパッケージ管理を用いる。
- |                |  |
|----------------|--|
| ▶ インストール済パッケー覧 |  |
| ▶ 更新可能なパッケー覧   |  |
| ▶ 全パッケを更新      |  |
| ▶ パッケをリポから検索   |  |
| ▶ パッケをインストール   |  |
| ▶ パッケを更新       |  |
| ▶ パッケをアンインストール |  |
| ▶ パッケの詳細を確認    |  |

## Slackware系限定のコマンド

### ■パッケージ


- |              |  |
|--------------|--|
| ▶ パッケをインストール |  |
|--------------|--|

## サービス、ミドルウェア

### ■サービス全般




- |              |  |
|--------------|--|
| ▶ 起動中のサービス一覧 |  |
| ▶ サービスの状態の確認 |  |

▶ ※ nano はとりあえず Ubuntu には標準で入っているらしい。

- |           |                              |        |
|-----------|------------------------------|--------|
| ▶ 起動      | \$ nano <i>filePath</i> ※    | ※未存でも可 |
| ▶ 保存して終了  | {Ctrl} + {X} → {Y} → {Enter} |        |
| ▶ 保存せずに終了 | {Ctrl} + {X} → {N}           |        |
- ▶ ※ 起動するとショートカットキー一覧が下に表示される。そこにおいて、 は Ctrl キーを意味する。

## Red Hat系限定のコマンド

### ■パッケージ

- ▶ ※ Red Hat系では RPM Package Manager というパッケージ管理を用いる。
- |                |  |   |
|----------------|--|---|
| ▶ インストール済パッケー覧 | \$ yum list installed  |   |
| ▶ 更新可能なパッケー覧   | \$ yum check-update  |   |
| ▶ 全パッケを更新      | \$ sudo yum update -y  |   |
| ▶ パッケをリポから検索   | \$ yum search <i>keyword</i> ※  に部分一致のを探す |   |
| ▶ パッケをインストール   | \$ sudo yum install <i>pack1</i> ...   | ※Yesが面倒なら  を |
| ▶ パッケを更新       | \$ sudo yum update <i>pack1</i> ...  | ※Yesが面倒なら  を |
| ▶ パッケをアンインストール | \$ sudo yum remove <i>pack1</i> ...  |   |
| ▶ パッケの詳細を確認    | \$ yum info <i>pack1</i> ...   |   |

## Slackware系限定のコマンド

### ■パッケージ

- |              |   |
|--------------|---|
| ▶ パッケをインストール | \$ sudo zypper install <i>pack1</i> ... |
|--------------|---|

## サービス、ミドルウェア

### ■サービス全般

- |              |                                    |              |
|--------------|------------------------------------|--------------|
| ▶ 起動中のサービス一覧 | \$ systemctl list-units -t service | ※ {Q} で抜けられる |
| ▶ サービスの状態の確認 | \$ systemctl status サービス           |              |

- ▶ サービスが起動中か確認
- ▶ サービスを起動する
- ▶ サービスを停止する
- ▶ サービスを再起動する
- ▶ サーバ起動時にサービスが自動起動するように
- ▶ サーバ起動時にサービスが自動起動しないように
- ▶ サーバ起動時のサービス自動起動の設定一覧

## ■SSHサーバ（sshデーモン、sshd）

### OpenSSHサーバ

- ▶ インストール (on Debian)
- ▶ インストール (on Red Hat)
- ▶ 各種設定を変更
- ▶ ※ 設定の変更を反映するには、SSHサーバを再起動する ( `$ sudo systemctl restart sshd .service` ) 必要がある。
- ▶ ☆ パスワード認証による接続を禁止する

## ■Webサーバ

### Apache HTTP Server (Apache)

- ▶ インストール (on Debian)
- ▶ ※ OSによっては自動的にドキュメントルートディレクトリに書き込み制限が設けられることがあるので、アクセス権限を変更する必要がある。

### Nginx

- ▶ インストール (on Debian)
- ▶ バージョン確認
- ▶ 設定ファを再読み込み
- ▶ 設定ファの構文チェック
- ▶ ※ Nginxでは1台で1つのドメインを運用する際にもバーチャルホスト技術を用いる。
- ▶ ☆ `/etc/nginx/` 以下の各種ファイル・ディレクトリの概要
- ▶ ☆ nginxにおける設定の構成単位は**モジュール**
- ▶ ☆ **ディレクティブ**とはnginxの設定ファイル内で使える命令のこと

- ▶ サービスが起動中か確認     `$ systemctl is-active サービス`
- ▶ サービスを起動する         `$ sudo systemctl start サービス`
- ▶ サービスを停止する         `$ sudo systemctl stop サービス`
- ▶ サービスを再起動する       `$ sudo systemctl restart サービス1 ...`
- ▶ サーバ起動時にサービスが自動起動するように     `$ sudo systemctl enable サービス`
- ▶ サーバ起動時にサービスが自動起動しないように    `$ sudo systemctl disable サービス`
- ▶ サーバ起動時のサービス自動起動の設定一覧       `$ systemctl list-unit-files -t service`

## ■SSHサーバ（sshデーモン、sshd）

### OpenSSHサーバ

- ▶ インストール (on Debian)    `$ sudo apt install openssh-server`
- ▶ インストール (on Red Hat)    `$ sudo yum install openssh-server`
- ▶ 各種設定を変更               `/etc/ssh/sshd_config` を開いて編集する
- ▶ ※ 設定の変更を反映するには、SSHサーバを再起動する ( `$ sudo systemctl restart sshd .service` ) 必要がある。
- ▶ ☆ パスワード認証による接続を禁止する

## ■Webサーバ

### Apache HTTP Server (Apache)

- ▶ インストール (on Debian)    `$ sudo apt install apache2`
- ▶ ※ OSによっては自動的にドキュメントルートディレクトリに書き込み制限が設けられることがあるので、アクセス権限を変更する必要がある。

### Nginx

- ▶ インストール (on Debian)    `$ sudo apt install nginx`
- ▶ バージョン確認               `$ nginx -v`
- ▶ 設定ファを再読み込み         `$ nginx -s reload`
- ▶ 設定ファの構文チェック       `$ nginx -t`
- ▶ ※ Nginxでは1台で1つのドメインを運用する際にもバーチャルホスト技術を用いる。
- ▶ ☆ `/etc/nginx/` 以下の各種ファイル・ディレクトリの概要
- ▶ ☆ nginxにおける設定の構成単位は**モジュール**
- ▶ ☆ **ディレクティブ**とはnginxの設定ファイル内で使える命令のこと

- ▶ ☆ **コンテキスト**とはモジュールあるいはディレクティブが作るスコープのこと
- ▶ ※ バーチャルホストごとに設定ファイルを分割すると管理しやすくなる。
- ▶ あるモジュールが有効か

## ■ファイルサーバ

### Samba

- ▶ 1. インストール (on Debian)
- ▶ 2. 共有ディを作成
- ▶ 3. 共有ディの所有者をなしに
- ▶ ☆ 4. 設定ファイルを編集して共有に関する設定を追加
- ▶ 5. サービスを再起動
- ▶ ☆ 6. ファイルを共有

## ■cron

- ▶ ※ Linuxディストリビューションであればインストールされている場合が多い。
- ▶ インストール (on Debian)
- ▶ インストール (on Red Hat)
- ▶ ※ インストールしたら通常は自動的に起動が始まる。
- ▶ 起動する
- ▶ ☆ 設定ファイルの書き方、設定ファイルの制約
- ▶ ☆ cron が動いていないと判明したら

## ■logrotate

- ▶ ※ Linuxディストリビューションであればインストールされている場合が多い。
- ▶ ※ 直近の rotate の日時はstateファイル（ステータスファイル）に記録されている。
- ▶ インストール (on Debian)
- ▶ ログローテーションを開始
- ▶ 別のstateファを指定して "

- ▶ ☆ **コンテキスト**とはモジュールあるいはディレクティブが作るスコープのこと
- ▶ ※ バーチャルホストごとに設定ファイルを分割すると管理しやすくなる。
- ▶ あるモジュールが有効か     \$ [ -z \$(nginx -V 2>&1 | grep -o -e '--with-モ') ]; echo \$?

## ■ファイルサーバ

### Samba

- ▶ 1. インストール (on Debian)     \$ sudo apt install samba
- ▶ 2. 共有ディを作成     \$ sudo mkdir -p /srv/samba/share/
- ▶ 3. 共有ディの所有者をなしに     \$ sudo chown nobody:nogroup /srv/samba/share/
- ▶ ☆ 4. 設定ファイルを編集して共有に関する設定を追加
- ▶ 5. サービスを再起動     \$ sudo systemctl restart smbd.service nmbd.service
- ▶ ☆ 6. ファイルを共有

## ■cron

- ▶ ※ Linuxディストリビューションであればインストールされている場合が多い。
- ▶ インストール (on Debian)     \$ sudo apt install cron
- ▶ インストール (on Red Hat)     \$ sudo dnf install -y crontabs
- ▶ ※ インストールしたら通常は自動的に起動が始まる。
- ▶ 起動する     ・ (on Debian)     \$ cron     ※ **-f** でフォアグラウンドに。  
                  ・ (on Red Hat)     \$ crond     ※ "
- ▶ ☆ 設定ファイルの書き方、設定ファイルの制約
- ▶ ☆ cron が動いていないと判明したら

## ■logrotate

- ▶ ※ Linuxディストリビューションであればインストールされている場合が多い。
- ▶ ※ 直近の rotate の日時はstateファイル（ステータスファイル）に記録されている。
- ▶ インストール (on Debian)     \$ sudo apt install logrotate
- ▶ ログローテーションを開始     \$ logrotate ※<sup>1</sup> *configFilePath*     ※<sup>1</sup>ここにオプを
- ▶ 別のstateファを指定して "     \$ .. -s *filePath* ..     ※デフォは **/var/lib/logrotate.status**

## 環境構築－Windowsの場合

## 環境構築－Windowsの場合

## ■全般

- ▶ ☆ VirtualBox と Docker の違い

## ■VirtualBoxで仮想マシンをつくる

### Vagrantを用いない場合

- ▶ ☆ 0. VirtualBoxのインストール
- ▶ ☆ 1. 新しく仮想マシンを作成し、起動
- ▶ ☆ 2. 仮想マシンに Linux ディストリビューションをインストール
- ▶ ☆ 3. Windows Terminal から仮想マシンにリモートログイン
- ▶ ☆ 4. 公開鍵認証を導入し、パスワード認証を禁止

### Vagrantを用いる場合

- ▶ ☆ 0. VirtualBoxのインストール・Vagrantのインストール
- ▶ ☆ 1. 新しく仮想マシンを作成し、起動 (Linux OSのインストールも込み)

## ■WSLでLinux環境をつくる

- ▶ ☆ 1. WSLをインストール
- ▶ ☆ 2. Linux ディストリビューション をインストールし、起動、初期化
- ▶ ☆ Linux ディストリビューションのディレクトリをエクスプローラーで開きたいなら
- ▶ ※ F. Linux ディストリビューションを終了したいなら普通に右上で閉じる。
- ▶ ※ R. 再び Linux ディストリビューションを起動したいなら、その名前を左下で検索。

## ■Dockerでコンテナをつくる

- ▶ ☆ 0. WSLでWindows OSと完全に統合されたLinuxディストリビューション環境をつくる
- ▶ ☆ 1. Dockerをインストール
- ▶ ☆ 2. Linux ディストリビューションからDockerを使えるように

## ■全般

- ▶ ☆ VirtualBox と Docker の違い

## ■VirtualBoxで仮想マシンをつくる

### Vagrantを用いない場合

- ▶ ☆ 0. VirtualBoxのインストール
- ▶ ☆ 1. 新しく仮想マシンを作成し、起動
- ▶ ☆ 2. 仮想マシンに Linux ディストリビューションをインストール
- ▶ ☆ 3. Windows Terminal から仮想マシンにリモートログイン
- ▶ ☆ 4. 公開鍵認証を導入し、パスワード認証を禁止

### Vagrantを用いる場合

- ▶ ☆ 0. VirtualBoxのインストール・Vagrantのインストール
- ▶ ☆ 1. 新しく仮想マシンを作成し、起動 (Linux OSのインストールも込み)

## ■WSLでLinux環境をつくる

- ▶ ☆ 1. WSLをインストール
- ▶ ☆ 2. Linux ディストリビューション をインストールし、起動、初期化
- ▶ ☆ Linux ディストリビューションのディレクトリをエクスプローラーで開きたいなら
- ▶ ※ F. Linux ディストリビューションを終了したいなら普通に右上で閉じる。
- ▶ ※ R. 再び Linux ディストリビューションを起動したいなら、その名前を左下で検索。

## ■Dockerでコンテナをつくる

- ▶ ☆ 0. WSLでWindows OSと完全に統合されたLinuxディストリビューション環境をつくる
- ▶ ☆ 1. Dockerをインストール
- ▶ ☆ 2. Linux ディストリビューションからDockerを使えるように