

第1章

イントロダクション

1.1 Linux とは

本書ではLinuxとよばれる ^{リナックス} OS（オペレーティングシステム）を学んでいく。世の中で有名な OS としては、Windows, macOS, Android, iOS などがある。これらは多くの人々が日常的に使う PC やスマートフォンなどで採用されているため、よく知られた存在であろう。それに比べると、Linux は非 IT エンジニアの人にはあまり聞き覚えのない OS かもしれない。

しかし、一般の人々からは見えないところで、Linux は幅広く利用されている。その用途は多岐にわたっているが、多くの IT エンジニアにとって特に関わりが深いのは、サーバ分野での利用である。我々が普段、24 時間いつでもインターネットに接続できて、いろいろな Web サービスを利用できるのは、Web サーバやデータベースサーバをはじめ、様々なネットワーク通信に関わるサーバが動作しているからである。そして、これらのサーバマシンの多くは、OS として Linux を採用している。

したがって、IT インフラ（ネットワークやサーバ）に関わる領域で活躍していくためには、Linux の知識は必須である。一方で、近年のソフトウェアやアプリケーションは、ネットワーク通信を行い、サーバと連携しながら処理を進めるものが多い。よって、プログラマや SE にとっても、Linux を学習しておく価値は大きい。本書は、これから IT エンジニアを目指す人が、Linux に関してまずは知っておくとよいだろうと思われる内容をまとめたものである。Linux の基本的な使い方はもちろん、簡単なサーバ構築・設定方法にも触れている。もしもあなたが IT 業界未経験の学生だとすると、就職先の企業の人に「Linux を使える新入社員が入ってきた」と思ってもらえる程度の内容をカバーしたつもりである。

ところで、Linux と言っていっても、実は様々な種類の Linux が存在する。この種類のことをディストリビューション（distribution）という。ディストリビューションが異なると、利用できる機能が一部異なる場合がある。

本書では、無料で利用可能な ^{セントオーエス}CentOS というディストリビューションを用いて学習する。CentOS は、企業での利用実績が豊富な有償のディストリビューションである RedHat Enterprise

Linux (RHEL) から、一部有償の機能などを取り除いたものであるが、大半の機能は共通である。そのため、CentOS を使って Linux を学んでおくと、RHEL もほとんど同じように使用できるだろう。また、CentOS 自体も、多くの企業で利用されている。

なお、本書では CentOS 7 の利用を想定している。実際に Linux を操作しながら学習を行うにあたり、本書が想定している環境と同じ実習環境を作りたい場合は、付録 A (p.187) を参照されたい。

1.2 GUI と CUI

Windows や macOS では通常、マウスでポインタを動かしてアイコンをクリックしたり、指で画面をタップするなどして、視覚的に操作を行う。このようなコンピュータの操作方法を **GUI** (Graphical User Interface) という。それに対して、Linux では、キーボードからコマンド (command) とよばれる様々な文字列を入力することでコンピュータに命令を伝える **CUI** (Character User Interface)*¹がよく使われる*²。

GUI では、コンピュータに詳しくない人でも、画面を見ながら「なんとなく」操作を行うことができる。現在、ここまで広く PC やスマートフォンが普及したのは、GUI のおかげであるといっても過言ではない。一方、CUI は前提知識としてコマンドの使い方を知らないと何もできない。その意味で、CUI は GUI に比べて敷居が高い。しかし、いろいろなコマンドを知り、使いこなしていくにつれて、CUI のほうが、複雑な処理をより手軽に実行できることがわかってくる。また、GUI と比べると、CUI は処理の負荷が軽いため、性能の低いマシンでも快適に動作する可能性が高いことも利点である。

本書では、Linux の基本的なコマンドを少しずつ学んでいく。最初のうちは、CUI の便利さがなかなかわからず、「GUI に比べて難しく面倒なだけだ」と思うかもしれない。しかし、本書を読み終える頃には、CUI に対する抵抗感が多少なりとも薄れ、その利点を感じられるようになってほしいと思う。

1.3 Linux に触れてみよう

それではさっそく、Linux のコマンド操作を体験してみよう。コマンドは端末 (terminal, ターミナル) とよばれるアプリケーション上で実行する。端末を起動すると、例えば次のようなプロンプト (prompt) とよばれる文字列が表示される*³。

*¹ **CLI** (Command Line Interface) ともいう。

*² 近年の Linux は GUI 機能も充実していることが多いが、サーバ用途で Linux が使われている場合は CUI 操作が主流である。IT 業界において「Linux が使える」とは「Linux をコマンドで操作ができる」と同義であるため、本書でも、Linux の CUI 操作に焦点を当てて説明していく。

*³ デフォルトのプロンプト表示は環境により異なるが、最後の 1 文字は \$ である可能性が高い。

12 第1章 イン트로ダクション

[ユーザ名@ホスト名 ~]\$

プロンプトは、「次のコマンドの入力を受け付けていること」を示す記号である。以降、本書ではプロンプト記号を \$ で表記する。

まずは、キーボードから `date` と入力して、[Enter] キーを押下しよう*4。

```
$ date      ←$より後の部分を入力し、[Enter] キーを押下（コマンドを実行）
2020 年   5 月 24 日 日曜日 13:54:27 JST      ←現在日時が表示される
$          ←次の入力を受け付けるプロンプトが表示される
```

この例では、`date` という文字列（`date` コマンド）を入力することで、Linux に「現在の日時を表示しなさい」と命令している。その結果として、現在日時が文字列で表示される。このように、文字で命令し、文字で結果が返されるのが CUI である。

他のコマンドも試してみよう。`cal` コマンド（CALENDAR）はカレンダーを表示するコマンドである。

```
$ cal      ← 今月のカレンダーを表示
          5 月 2020
日 月 火 水 木 金 土
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

`cal` コマンドは、コマンド名の後ろに^{ひきすう}引数（argument）を書くことで、表示するカレンダーの年月を指定できる*5。

```
$ cal 4 2020  ← 2020 年 4 月のカレンダーを表示
$ cal 2020    ← 2020 年 1 月～12 月のカレンダーを表示
```

コマンド名と引数のあいだは、半角空白文字で区切る必要がある。また、引数はコマンドライン引数とよばれることもある。一般に、コマンドの引数は、処理対象を指定するために使われることが多い。例えば、ファイルを削除するコマンドの場合、削除対象のファイル名を引数で指定する。

また、`cal` コマンドにはいくつかのオプション（option）も用意されている。ここでは、`-3` オプションと `-m` オプションを紹介する。コマンド名とオプションのあいだも、半角空白文字で区切る。

*4 コマンドの入力、およびその実行結果の表示をあわせて記載する場合、本書ではこの例のように枠で囲んで示すことにする。また、「←」以降は補足説明であり、実際に入力したり、画面に表示されたりするものではない。本書では、以降も必要に応じてこのような補足説明を入れていく。

*5 これまでの例と異なり、ここではコマンドを実行した結果の表示を省略している。このような場合、本書では枠で囲まずに示すことにする。

```
$ cal -3      ←先月, 今月, 来月の 3 ヶ月分のカレンダーを表示
$ cal -m      ←月曜 (Monday) 始まりで今月のカレンダーを表示
```

多くのコマンドにおいて、オプションは - (ハイフン) で始まる。引数が主に処理対象を指定するために用いられるものであるのに対し、オプションはコマンドの動作 (処理) を変えるために用いられることが多い。

複数のオプションを同時に指定したいときは次のように書く。

```
$ cal -3 -m    ← 2つのオプションを別々に記述
$ cal -m -3    ← オプションの順番を変えても同じ
$ cal -3m      ← 2つのオプションをまとめて記述
$ cal -m3      ← オプションの順番を変えても同じ
```

大抵の場合、オプションの指定順序は自由である。上記のいずれの書き方でも、先月、今月、来月の 3 ヶ月分の月曜始まりのカレンダーが表示される。

オプションと引数を組み合わせて指定することもできる。次のように実行すると、2020 年 4 月の月曜始まりのカレンダーが表示される。

```
$ cal -m 4 2020
```

原則として、オプションと引数は次の順で指定することを覚えておこう。

```
$ コマンド名 オプション 引数
```

また、Linux では、英字の大文字と小文字が厳格に区別される点に注意しよう。例えば、次のように実行しようとする、いずれもエラーとなる。

```
$ Date          ←コマンド名は Date ではなく date が正しい
$ cal -M 4 2020 ←オプションは-M ではなく -m が正しい
```

同様に、ファイル名の大文字と小文字も区別される。ファイル `file1` とファイル `File1` は別物である。

1.4 マニュアル閲覧

ここまで、`date` コマンドと `cal` コマンドを例にして、CUI とはどのようなものであるかを体験してきた。`cal` コマンドでは、オプションや引数の指定方法も扱った。コマンドの詳しい使い方を調べたいときは、**man** コマンド (MANual) を用いてマニュアルを閲覧するとよい^{*6}。

```
$ man コマンド名
```

^{*6} ただし、マニュアルはお世辞にもわかりやすいとは言い難い。読んでも意味がわからないときは、コマンドの使い方をわかりやすく解説してくれている Web サイトなどもあわせて参照するとよい。

14 第 1 章 イン트로ダクション

例えば，次のように実行すると，cal コマンドのマニュアルが表示される．

```
$ man cal
```

マニュアルは，矢印キーの上下でスクロールできる．閲覧を終了したいときは，キーボードの [Q] キーを押下する．

章末問題

[1] この章で扱ったコマンドのうち，次の説明に適するものを書け．

- (1) カレンダーを表示する．
- (2) マニュアルを表示する．
- (3) 現在日時を表示する．

[2] 空欄に適する語を，あとの選択肢から 1 つずつ選べ．

- CentOS は Linux の (①) の 1 つである．
- Windows のようにマウスやタッチ操作で視覚的に操作を行うものを (②) というのに対し，Linux の端末上での操作のように，コマンドとよばれる文字列により操作を行うものを (③) という．
- 次のコマンド入力を受け付けていることを示す記号を (④) という．

選択肢：

API CUI GUI オプション ディストリビューション 引数 プロンプト

[3] 2000 年 10 月のカレンダーを月曜始まりで表示するコマンドを，オプションや引数も含めて答えよ．