基礎から学べるLinux入門の決定版

Linux
標準教科書
Ver.4.0.1

LinuC
Linux技術者認定
https://linuc.org

open your NEXT future
LPI-JAPAN
Linux Professional Certification
https://lpi.or.jp

Linux標準教科書

2026年2月18日　Ver.4.0.1
LPI-Japan　発行

まえがき

特定非営利活動法人エルピーアイジャパンが、「Linux 標準教科書」の初版を公開したのは 2008 年のことでした。Linux 技術者教育に利用していただくことを目的として開発し、今回で 3 回目の改訂となりました。

今回の改訂では、標準教科書シリーズを独学自習で使っている方が多くなってきたことから、仮想マシンを使って実習環境を構築して実習を進める手順に変更しました。また、これまでの内容を全面的に見直し、実習中心の「動かして学ぶ」スタイルに書き改めました。

公開にあたっては、本教科書に添付されたライセンス（クリエイティブ・コモンズ・ライセンス）の下に公開されています。

本教科書は、最新の技術動向に対応するため、随時アップデートを行っていきます。

本教科書の最新情報は以下の Web ページをご参照ください。

`https://linuc.org/textbooks/linux/`

本教科書の目的

本教科書の目的は、基本的な Linux のコマンド操作と簡単なシステム管理に必要な知識を、実習を通して学習することにあります。

この教科書の位置づけはあくまで「Linux を初めて触る人のはじめの一歩」というものになっており、Linux 技術者として修得して欲しい技術すべてをカバーできていません。細々とした解説を行うことで冗長になることを避けるために、解説は必要最低限のものとしています。この教科書とあわせて『Linux サーバー構築教科書』および『Linux システム管理標準教科書』も学習してみてください。この 3 部作を一通り学習すれば、市販の Linux 関連書籍などを手に取っても、何が書いてあるのか分かるスキルレベルに到達できるでしょう。本教科書の内容を学習すれば、他の 2 冊の教科書の実習もスムーズに進められるようになります。

想定している実習環境

本教科書は、一人で独学自習できることを想定しています。実習環境として、以下の環境を構築しています。

仮想マシンを利用

仮想マシンを利用して学習環境を構築します。仮想マシンを利用すると、Windows や Linux、macOS 上の仮想マシンに Linux をインストールし、動作させることができます。

仮想マシン環境を実現するソフトウェアとして、以下のようなものがあげられます。

- VirtualBox（Windows、Linux、macOS）
- VMware Workstation（Windows）
- VMware Fusion（macOS）
- Parallels Desktop（macOS）
- UTM（macOS）
- Linux KVM（Linux）

本教科書では、VirtualBox を Windows 上で実行して解説を進めます。

OS

本教科書では、Linux ディストリビューションとして AlmaLinux のバージョン 9.4 を利用します。

実習例では Intel/AMD x86_64 アーキテクチャに対応したバージョンを利用していますが、ARM 版などその他のアーキテクチャに対応したバージョンでも実習を行うことができます。

ネットワーク

実習を行うネットワークはインターネットに接続できることを前提としています。インストール用の ISO イメージはサイズがとても大きいので、高速なインターネット接続でダウンロードするようにしてください。作業手順の説明の関係上、インストール直前に ISO イメージをダウンロードしていますが、時間がかかる場合があります。回線速度が遅い場合には、先にダウンロードを行っておくようにするとよいでしょう。

インターネットに接続できない場合には、インストール用の ISO イメージをどこかでダウンロードしてコピーする必要があります。

全体の流れ

本教科書では、以下の通りに実習を進めます。

1 章 Linux を学ぶ

Linux を学ぶとはどういうことなのか、本教科書で学習する内容の全体像を解説します。

2 章 VirtualBox のインストールと仮想マシンの作成

VirtualBox のインストール、仮想マシンの作成を行います。

3 章 Linux のインストールと設定

仮想マシンに Linux をインストールします。

4 章 Linux を操作してみよう

Linux をコマンドで操作することについてのイメージを掴むため、Web サーバーを動かしてみます。また、以後の実習を進めやすくするため、SSH によるリモートログインの方法も解説します。

5 章 基本的なコマンド

Linux を操作するための基本的なコマンドであるファイルやディレクトリの操作、その他基本コマンドを解説します。

6 章 標準入出力とフィルタコマンド

Linux のコマンド間のデータを連携する標準入出力と、標準入出力を活用するフィルタコマンドについて解説します。

7 章 vi エディタ

Linux の基本的なエディタである vi エディタの操作方法について解説します。

8 章 ユーザーとグループの管理

Linux のシステム制御の基本単位であるユーザーとグループの管理について解説します。

9 章 ファイルやディレクトリのアクセス制御

ファイルやディレクトリに対するアクセス制御について解説します。

10 章 ネットワークの設定と管理

Linux のネットワークの設定と管理について解説します。

11 章 プロセス管理

Linux 上で動作するプログラムはプロセスとして扱われます。プロセスの管理について解説します。

12 章 腕試しの課題とまとめ

本教科書の内容が理解できているか、Web サーバーの「ユーザーディレクトリ機能」を題材に腕試しの課題に挑戦してみましょう。

執筆者紹介

本教科書は、オープンなプロジェクト形式で開発を行っています。企画段階から意見交換を行い、事前の技術的な調査、執筆、レビューなどをプロジェクトのメンバーで分担して行っています。

宮原 徹（バージョン **1** 企画・バージョン **4** 執筆担当）

本教科書は、2008 年にバージョン 1 を企画、公開しました。16 年が経ち、現在では不要となっている技術も多くあるため、それらの項目を省きつつ、最新の LinuC レベル 1 の出題範囲を考慮して思い切って仮想マシンを使った実習環境構築から行う手順に変更しました。その他、実務で必要となる最低限の知識を意識して、短時間で実習を進められるように改訂しています。

Linux の基本的なコマンドの学習は、山や森全体から見れば 1 本 1 本の木のようで、なかなか全体像が掴めないかと思います。最初はまず実習を 1 つずつ確実に実行して、実行例と同じ結果が返ってくることを確認してみてください。一通り最後まで終わらせたら、再度それぞれの実行例の意味を考えながら、必要に応じて本教科書以外の書籍の解説も参照しながら実行してみてください。3 回目には、それぞれのコマンドの意味がわかってくるかと思います。1 回では覚えられない前提で、3 回は繰り返してみてもらえればと思います。

バージョン **4** の開発にご協力をいただいた方々（**50** 音順）

バージョン 4 の開発にあたり、企画から執筆、そして発行後の普及活用に多くの方のご協力をいただきました。

- 荒井 剛
- 板橋 章夫
- 井上 潮（東京電機大学）
- 河原木 忠司
- 鯨井 貴博（株式会社ゼウス・エンタープライズ）
- 竹本 季史（インターノウス株式会社）
- 中 翔
- 中谷 徹
- 福永 昭臣（株式会社ボールド）
- nkgr2016
- Sota

また、これまでバージョン 1 からバージョン 3 まで、沢山の執筆者、レビュアー、そして利用者の皆様からフィードバックをいただきました。厚く御礼申し上げます。

著作権

使用に関する権利

図 1: CC BY-NC-ND 4.0

表示

本教科書は、特定非営利活動法人エルピーアイジャパンに著作権が帰属するものであることを表示してください。

非営利

本教科書は、非営利目的で教材として自由に利用することができます。

商業上の利得や金銭的報酬を主な目的とした営利目的での利用は、特定非営利活動法人エルピーアイジャパンによる許諾が必要です。ただし、本教科書を利用した教育において、本教科書自体の対価を請求しない場合は、営利目的の教育であっても基本的に使用できます。その場合も含め、LPI-Japan 事務局までお気軽にお問い合わせください。

＊営利目的の利用とは以下のとおり規定しております。営利企業または非営利団体において、商業上の利得や金銭的報酬を目的に本教科書の印刷実費以上の対価を受講生に請求して本教科書の複製を用いた研修や講義を行うこと。

改変禁止

本教科書は、改変せず使用してください。本教科書に対する改変は、特定非営利活動法人エルピーアイジャパンまたは特定非営利活動法人エルピーアイジャパンが認める団体により行われています。

フィードバック

フィードバックは誰でも参加できる Slack で受け付けていますので、積極的にご参加ください。Slack 参加の詳細は以下の本教科書の Web ページを参照してください。

`https://linuc.org/textbooks/linux/`

図 2: https://linuc.org/textbooks/linux/

本教科書の使用に関するお問合せ先

特定非営利活動法人エルピーアイジャパン（LPI-Japan）事務局

お問い合わせ：`https://lpij.tayori.com/f/textbookinfo/`

図 3: https://lpij.tayori.com/f/textbookinfo/

Linux 標準教科書と LinuC レベル 1 の関係

Linux 標準教科書で Linux 操作の基礎を習得することはできますが、実際に業務で Linux サーバーを操作・管理をするためには Linux 標準教科書の内容に加えて LinuC レベル 1 で習得できるスキルも必要になることがあります。特に LinuC レベル 1 の出題範囲にある仮想マシンやコンテナ、セキュリティ、オープンソースについての理解などは、現在の IT エンジニアにとっては必須のスキルとなります。業務で Linux サーバーの操作と運用が行えるスキルの習得を目指す方は、Linux 技術者認定「LinuC レベル 1」の認定取得を目指してみてはいかがでしょうか。

Linux 技術者認定「LinuC（リナック）」のご紹介

Linux 技術者認定「LinuC（リナック）」とは、クラウド／ DX 時代の IT エンジニアに求められるシステム構築から運用管理に必要なスキルを証明できる技術者認定です。アーキテクチャ設計からシステム構築、運用管理までの技術領域を広くカバーしており、４つのレベルの認定取得を通じて一歩ずつ確実に求められるスキルを習得し、それを証明することができます。

LinuC の出題範囲策定や試験開発は、実際に現場で活躍しているハイレベルな IT エンジニアが参加するコミュニティによって行われています。そのため、グローバルで業界標準として利用されている技術領域をカバーし、システム開発や運用管理の現場で本当に必要とされる知識や実践的なスキルを問う内容になっています。その結果として従来型の Linux 領域にとどまった技術認定とは異なり、国内・海外を問わず活躍を目指す IT エンジニアにとっても十分役立つ技術者認定となりました。

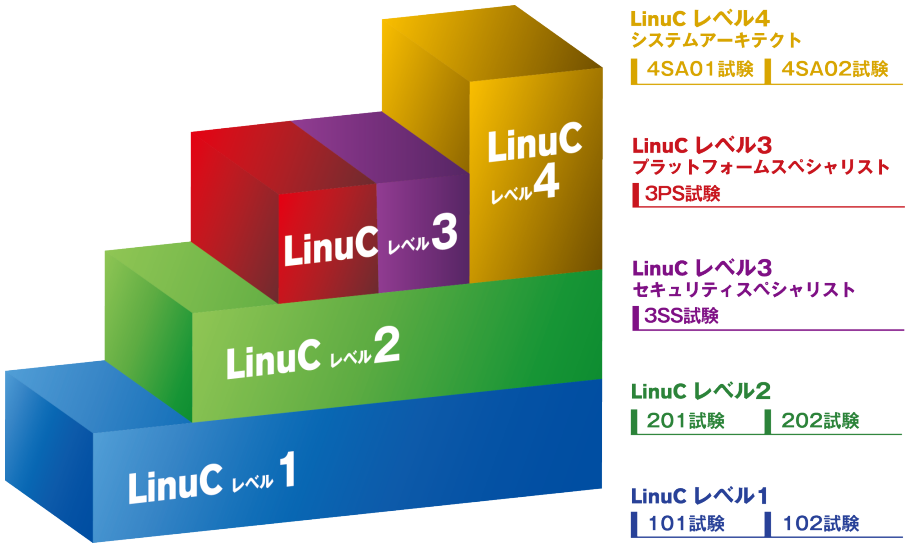

図 4: LinuC の体系図

LinuC レベル 1

コンピュータシステムを理解し、仮想環境を含む Linux システムの基本操作とシステム管理が行える即戦力エンジニアの証明（ITSS レベル 1）

LinuC レベル 2

仮想環境を含む Linux のシステム設計・ネットワーク構築において、アーキテクチャに基づいた設計・導入・保守・問題解決ができるエンジニアの証明（ITSS レベル 2）

LinuC レベル 3 プラットフォームスペシャリスト・セキュリティスペシャリスト

仮想化・自動化などにより柔軟性と可用性を備えた Linux プラットフォームの構築運用や、セキュア Linux システムのための OS からミドルウェアまでの各層堅牢化や認証認可基盤の構築および攻撃対策を実現できるスペシャリストの証明（ITSS レベル 3）

LinuC レベル 4 システムアーキテクト

オンプレ／クラウド、物理／仮想化を含むシステムのライフサイクル全体を俯瞰して最適なアーキテクチャを設計・構築ができる上級エンジニアの証明（ITSS レベル 4）

LinuC の詳細については、以下の Web サイトを参照してください。

https://linuc.org/about/01.html

図 5: https://linuc.org/about/01.html

LinuC の認定取得に向けた学習方法

LinuC の特徴の 1 つは学習環境が充実していることです。LinuC の認定取得に必要となるスキルや知識を身につけるには　体系的な学習として技術解説書を読むだけではなく、実際に手を動かして確認し問題集を繰り返し解いて理解を定着させる方法を推奨しています。

体系的にしっかりとした教育を実施している認定校や認定教材のほか、様々な学習コンテンツや無料で参加できるセミナーなどが豊富に用意されています。

詳しくは以下のリンク先「学習のすすめ方」をご覧ください。

https://linuc.org/measures/

図 6: https://linuc.org/measures/

LinuC の出題範囲を網羅した LPI-Japan 認定教材については以下のリンク先「認定教材」をご覧ください。

https://linuc.org/measures/textbook/

図 7: https://linuc.org/measures/textbook/

目次

1 Linux を学ぶ

この『Linux 標準教科書』は、Linux の基本的な操作について一通り学ぶための教科書です。Linux を操作するために必要な知識を修得するため、実習中心に実際に動かして学んでいきます。

本章では、実習に入る前に Linux を学ぶということはどういうことか、そしてこの教科書に出てくる様々な事柄の意味について解説します。

本章の内容

- Linux を学ぶということ
- Linux とは
- Linux でできること
- Linux を操作する方法
- コマンドを実行して Linux を操作する

1.1 Linux を学ぶということ

現在の IT システムにおいて、Linux は非常に重要な役割を担うようになっています。たとえば、スマートフォンの中で動いている Android は、Linux をベースにしたソフトウェアの集まりでできています。インターネットの向こう側で様々なサービスを提供しているシステムも、その多くが Linux で動作しています。テレビなどの家電製品にも、Linux が組み込まれて動作しています。数えきれないぐらいの機器の OS として Linux が採用されています。Linux を学ぶということは、これらの機器やシステムを作ったり、直したりする側になるための技術を学ぶということになります。

1.1.1 Linux の操作はコマンドで行う

様々な場所で動作している Linux ですが、その基本的な操作はコマンドと呼ばれる命令をキーボードで入力、実行して行います。結果の表示も画面に文字で表示される CUI（Character User Interface）のインターフェースになっています。Windows や macOS のようなグラフィカルな画面とマウスで操作する GUI（Graphical User Interface）でも Linux を操作できますが、本教科書では GUI の使い方については扱いません。直感的に使うことができるので、自分であれこれと試してみてください。

Linux が組み込まれているスマートフォンなどの機器も、グラフィックで表示したりタッチパネルで操作したりしますが、実際に Linux が動いているのは裏側の部分で、見た目や操作を実現しているのは Linux 上で動作する別のソフトウェアが担っています。これらは Linux を応用して実現していることなので、Linux の基本はあくまでキーボードでの入力や文字での出力ということになります。

1.1.2 Linux は地味？

本教科書で取り扱っている Linux の基本的な操作は、とても地味な作業です。また、一つ一つのコマンドや技術はとても小さなものなので、最初のうちはそれだけを見ると何のために覚えないといけないのかわからないでしょう。しかし、そのコマンド操作の結果の組み合わせによって、膨大な数の機器がインターネットに接続されて、今では欠かすことのできないシステムを構成しているわけです。いろいろな小さな技術の組み合わせが、インターネットのようなシステムを形作っているということを知ってください。

1.1.3 基礎の基礎の重要性

本教科書のうち、第 6 章までは Linux 操作の基礎の基礎という感じで、たとえれば文字の読み書きや四則演算など、小学校低学年で学ぶような事に位置づけられます。まず「習うより慣れろ」で、実習を通しで繰り返しやってみて、コマンド操作になれるようにしてください。実習内容は、実際に実務に就いた時に必要となる必要最小限の内容に絞り込んであります。また、説明は余計なことを考えないでもいいよう、説明不足ではないかというぐらい簡潔にしてあります。書かれていることが分かるようになれば、実務で困ることはないでしょう。ただし、思い切って省いてしまった部分にも沢山面白い技術が詰まっているので、是非本教科書以外の書籍やインターネット上の情報にあたって、知識の幅を拡げていってください。

1.2 Linux とは

本教科書で取り上げる Linux とは、狭い意味でいえば「カーネル」と呼ばれる OS の中核を成すソフトウェアです。ただし、カーネルだけではコンピューターを動かすことができないので、Linux カーネル上で動作する様々なソフトウェアと組み合わせて OS を形成しています。

1.2.1 OS とは

OS とは、コンピューターを動作させるための基本的なソフトウェアを指します。コンピューターをどのような目的で使うかによって OS の選択は変わります。一般的なパソコンでは、Windows や macOS のような OS が選択されますし、スマートフォンなどでは Android や iOS などが利用されています。

OS は、カーネルと呼ばれる OS の中核になるソフトウェアと、様々なデバイスドライバーやライブラリー、基本的なプログラムなどが組み合わされて動作しています。ここまでの組み合わせが OS と言えるような厳密な区分はありませんので、システムは「OS ＋アプリケーション」という組み合わせで出来ている、ということを理解しておけばよいでしょう。

1.2.2 ディストリビューションとは

Linux はカーネルと様々なソフトウェアの組み合わせでシステムを構成していますが、それらのソフトウェアをまとめてインストールできるようにしたのがディストリビューションです。ディストリビューションにはインストーラーや、各ソフトウェアを管理するパッケージ管理ツールが用意されており、自分の好きな組み合わせでシステムを構成できます。

本教科書ではディストリビューションとして AlmaLinux を使って実習を進めますが、Ubuntu などその他のディストリビューションでも実習を進めることができます。

1.3 Linux でできること

Linux は様々なソフトウェアを組み合わせてシステムを構成します。どのような目的のシステムがよく使われているのでしょうか。

1.3.1 サーバー

サーバーは、ネットワークで様々なサービスを提供するコンピューターです。その目的に応じたサーバー用ソフトウェアを実行します。Linux が大きく普及したのは、インターネットで様々なサービスを提供するためのサーバーとして Linux が使いやすかったことが理由として挙げられます。以下のようなサーバーが Linux で実現できます。

- Web サーバー
  Web ブラウザでアクセスして Web ページを表示するサーバーです。最近ではスマートフォンのアプリケーションも内部では Web サーバーにアクセスして情報を取得し、綺麗に画面に表示するといった動作をするので、幅広く必要とされるサーバーです。
- メールサーバー
  電子メールのやり取りを行うためのサーバーです。
- ファイルサーバー
  各種ファイルを保管し、複数の利用者で共有するサーバーです。
- データベースサーバー
  アプリケーションが利用するデータを保管し、必要に応じてデータを取り出して提供するサーバーです。

1.3.2 デスクトップやワークステーション

Linux は GUI で操作できるので、Windows や macOS 同様に Web ブラウザやワープロ、表計算ソフトなどを実行できます。

また、プログラミング環境やデータ分析用のツールなどが整っているため、プログラムの開発やデータ分析を行うワークステーションとして活用されることもあります。

1.3.3 スマートフォンや家電製品

スマートフォンやタブレットの多く採用されている Android は Linux をベースにした OS です。

また、各種家電製品にも Linux が搭載されています。

1.3.4 組み込み機器

IoT（Internet of Things）と呼ばれる、様々なものがインターネットに接続して通信を行うデバイスにも Linux が搭載されています。

Raspberry Pi のようなシングルボードコンピューターでも Linux が動作し、IoT に活用されています。

1.3.5 AI・機械学習

画像認識や生成系 AI など、AI・機械学習の分野で使われるソフトウェアの多くが Linux 上で動作します。

1.4 Linux を操作する方法

Linux を操作するには、Windows や macOS のような GUI での操作と、端末にコマンドを入力する CUI があります。

1.4.1 GUI

Linux 上で、X Window System や Wayland などの GUI を実現するソフトウェアを動作させることで、Linux を GUI（Graphical User Interface）で操作できます。

GUI での操作方法は、Linux が動作しているコンピューターを直接操作するローカルログインと、リモートデスクトップソフトウェアを使って遠隔地の Linux を操作するリモートログインの 2 つの方法があります。

1.4.2 CUI

端末に対してキーボードで文字を入力して、結果も文字で返ってくるのが CUI（Character User Interface）です。コマンドラインで操作するインターフェースということで CLI（Command Line Interface）とも呼ばれます。GUI に比べてシンプルな操作なので、Linux をサーバーとして動かす場合などには CUI で操作することが多くなります。また、操作に関わる入出力のデータ量が GUI に比べて少なくなるので、ネットワーク経由で遠隔地の Linux を操作する際に回線速度が遅くても快適に操作できるメリットがあります。

CUI で操作するには、以下の方法があります。

- GUI でログイン後、端末アプリを起動する
  手元にコンピューターがある場合には、最も簡単に操作できる方法です。
- 仮想コンソールを切り替える
  Linux は起動すると GUI で起動しますが、仮想コンソールを切り替えることで CUI で操作することができます。仮想コンソールは 1 番から 6 番まで用意されており、1 番が GUI、2 番から 6 番が CUI での操作になっています。仮想コンソールの切り替えは、Ctrl+Alt+1〜6 で対応する仮想コンソールに切り替えられます。GUI から CUI にするなら、たとえば Ctrl+Alt+2 を入力します。
- GUI 無しで Linux をインストールする
  Linux を GUI 無しでインストールすると、起動時に最初から CUI の仮想コンソールを表示して起動します。後から GUI を起動しないように設定することもできます。
- SSH でリモートログインする
  ネットワーク経由で遠隔地の Linux を操作する場合には、SSH プロトコルで接続してリモートログインできます。SSH クライアントは、Linux はもちろん、Windows や macOS でも動作するので、好きなコンピューターからリモートログインできます。

1.5 コマンドを実行して Linux を操作する

Linux にログインし、コマンドを実行できるようになったら、様々なコマンドを入力、実行して Linux を操作します。この操作には、コマンドを実行して結果を得る単純な操作と、サーバーのような常時実行するプログラムを操作するものに大別されます。

1.5.1 通常のコマンド実行

通常のコマンド実行では、ファイルなどからデータを読み込んで、結果を表示する動作を行います。

たとえば、どのようなファイルが存在するのか確認したり、ファイルの内容を表示するなどの操作がこれに当たります。実行されたコマンドは、結果を表示するか、終了の指示が与えられると終了します。

1.5.2 サーバーとして動作させる

Linux はサーバーとして使われることが多いので、常時実行し続けるプログラムを起動したり、停止したりすることも頻繁にあります。

サーバーのためのプログラムは、起動したまま処理要求を待機します。このようなプログラムを「デーモンプロセス」と呼ぶことがあります。デーモンは「daemon」と記述するので、デーモンプロセスは最後に「d」が付くものが多くあります。たとえば、Web サーバーは HTTP というプロトコルを扱うデーモンということで「httpd」という名前になっています。

2 VirtualBox のインストールと仮想マシンの作成

実習を進めるためには、Linux を実際に操作できる環境が必要となります。実習環境は様々なものがありますが、本教科書では手軽に準備できる仮想マシンを使用します。

本章では、実習環境の準備として VirtualBox をインストールして、仮想マシンを作成します。

本章の内容

- 仮想マシンとは
- VirtualBox のインストール
- VirtualBox の起動
- 仮想マシンの作成
- ネットワークの追加
- ホストキーによるホスト OS の操作への復帰

2.1 仮想マシンとは

仮想マシンは、ソフトウェアで仮想的なマシンを実行する仕組みです。仮想マシンにはゲスト OS として Linux をインストールし、操作することができます。

仮想マシンを使うのには、以下のようなメリットがあります。

2.1.1 学習用のハードウェアを別途用意する必要がない

仮想マシン用のソフトウェアは、普段使用している Windows などのホスト OS 上でアプリケーションとして実行されるので、別途 Linux 用のハードウェアを用意する必要がありません。

2.1.2 その他のアプリと同時実行できる

仮想マシンと同時にホスト OS 上でその他のアプリケーションも実行できます。たとえば、本教科書の PDF 版を表示しながら実習を進める、ということが簡単に行えます。

2.1.3 ホスト OS のアプリケーションからゲスト OS にネットワーク接続できる

ホスト OS とゲスト OS はネットワークで接続されています。ホスト OS 上で実行している Web ブラウザから、ゲスト OS で実行している Web サーバーに接続する、といったことができます。

2.1.4 メモリやストレージの容量に注意

仮想マシンに対する CPU やメモリ、ストレージなどのリソースの割り当ては自由に決められます。学習用に Linux を動かす仮想マシンにはそれほど沢山のリソースは必要ありませんが、大体以下のようなリソースの割り当てが必要になります。

リソース	割当量
CPU	1 から 2
メモリ	2GB
ストレージ	20GB

本教科書では、CPU の負荷が高い処理は行わないのであまり気にする必要はありません。メモリとストレージは実際の容量を消費するので、メモリが少ない場合には余計なアプリを停止し、ストレージは不要なデータを削除して空き領域を作る必要があります。

2.2　VirtualBox のインストール

本教科書の実習では、仮想マシンソフトウェアである「VirtualBox」上に Linux を導入します。

2.2.1　使用するコンピュータの仮想化支援技術の有効化

VirtualBox を実行するには、使用するコンピュータが搭載しているプロセッサの仮想化支援技術が有効になっている必要があります。仮想化支援技術は、Intel の CPU では Intel VT、AMD の CPU では AMD-V と呼ばれます。

仮想化支援技術を有効にするには、コンピュータの BIOS/UEFI 設定画面で設定を行います。設定方法の詳細は、使用するコンピュータの説明書などを確認してください。ほとんどの場合、電源を投入後、OS が起動する前にファンクションキーを押して設定画面を呼び出して設定を行います。

2.2.2　VirtualBox のダウンロード

VirutalBox は、次の URL からダウンロードできます。

```
https://www.virtualbox.org/
```

ホスト OS の種類に合わせてダウンロードを行えます。今回は Windows 環境にインストールします。上部のメニューから「Download」をクリックし、ダウンロードページを表示します。「VirtualBox Platform Packages」から「Windows hosts」をクリックして、インストーラーをダウンロードします。

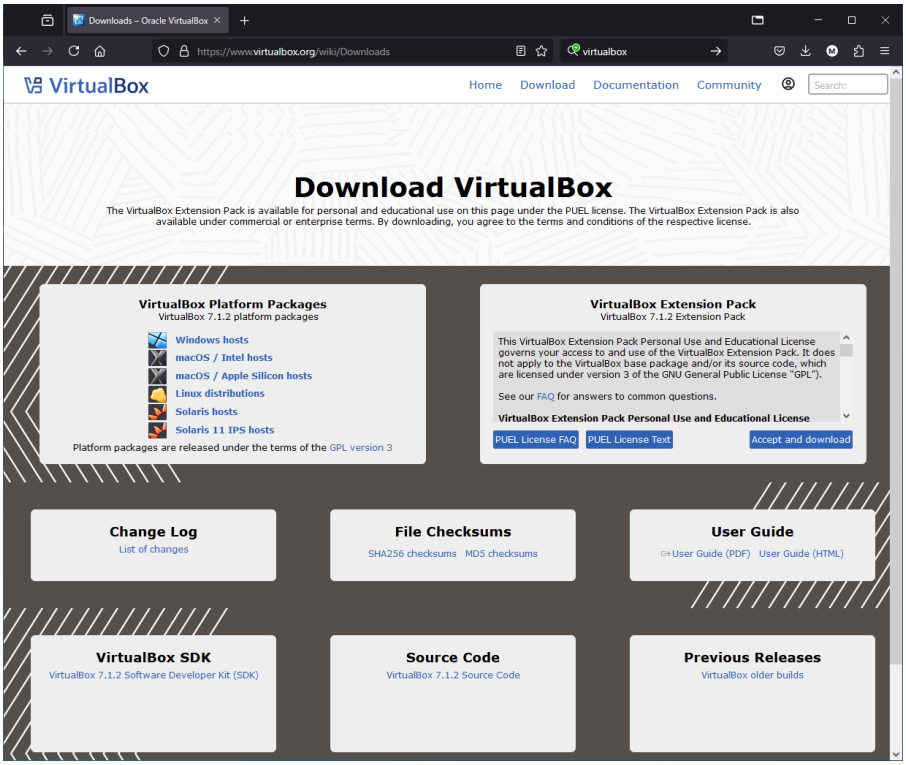

図 8: VirtualBox のダウンロードサイト

今回はバージョン 7.1.2 を使用しています。

2.2.3　VirtualBox のインストーラーの実行

VirtualBox のインストーラーをダウンロードが完了したら、インストーラーを実行してインストールします。

インストール時の特別な設定は行いませんので、インストーラーの指示に従ってインストールします。インストーラーが仮想ネットワークのインストールについて、また Python との連携について確認を取ってきますが、そのまま進めても問題ありません。

インストール終了後、Finish ボタンをクリックするとインストーラーが終了し、VirtualBox が起動します。

2.2.4 ホスト OS の再起動

通常、VirtualBox を新規インストールした後、起動して仮想マシンを動作させることができます。しかし、この後の作業で仮想マシンが正しく動作しない場合には、ホスト OS である Windows を再起動してみてください。

2.3 VirtualBox の起動

VirtualBox を起動します。

インストーラーが自動で起動した状態ではない場合には、デスクトップ上に追加された VirtualBox をダブルクリックして起動します。

VirtualBox が起動すると、VirtualBox マネージャーが表示されます。

図 9: VirtualBox マネージャーの画面

2.3.1 VirtualBox マネージャー

VirtualBox マネージャーは、VirtualBox による仮想マシン実行環境全体を管理するツールです。主に以下のことが行えます。

- VirtualBox 自身の各種設定
- 仮想マシンの管理
- メディア（ISO イメージ等）の管理
- 仮想ネットワークの管理

2.4 仮想マシンの作成

実習で使用する仮想マシンを作成します。

VirtualBox マネージャーの「ようこそ VirtualBox へ！」の画面で「新規 (N)」ボタンをクリックし、新しい仮想マシンの作成を開始します。画面が切り替わっている場合には、「ツール」右側のリストボタン（横 3 本線）をクリックし、「ようこそ」を選択します。

2.4.1 仮想マシンの名前と OS の設定

まず最初に、仮想マシンの名前と OS の種類などを設定します。

設定する項目と、設定する内容は以下の通りです。

項目	設定値
名前	AlmaLinux
タイプ	Linux
Subtype	Red Hat
バージョン	Red Hat 9.x (64-bit)

ISO イメージはインストール前に設定するので、ここでは設定しないでおきます。

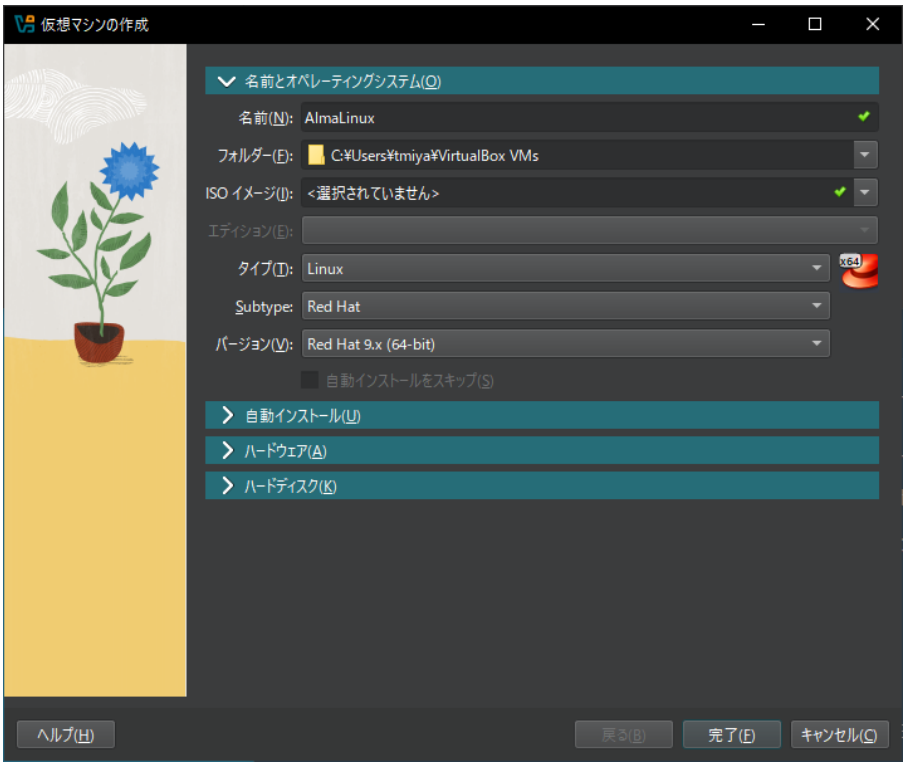

図 10: 仮想マシンの名前と OS 選択の画面

設定したら、「＞ハードウェア (A)」をクリックします。

2.4.2 仮想マシンのハードウェアの設定

仮想マシンのハードウェアの設定として、メインメモリーの容量とプロセッサの数を設定します。また、EFI の設定も行います。

項目	設定値
メインメモリー	2048MB
プロセッサー数	1
EFI を有効化 (一部の OS のみ)	チェックする

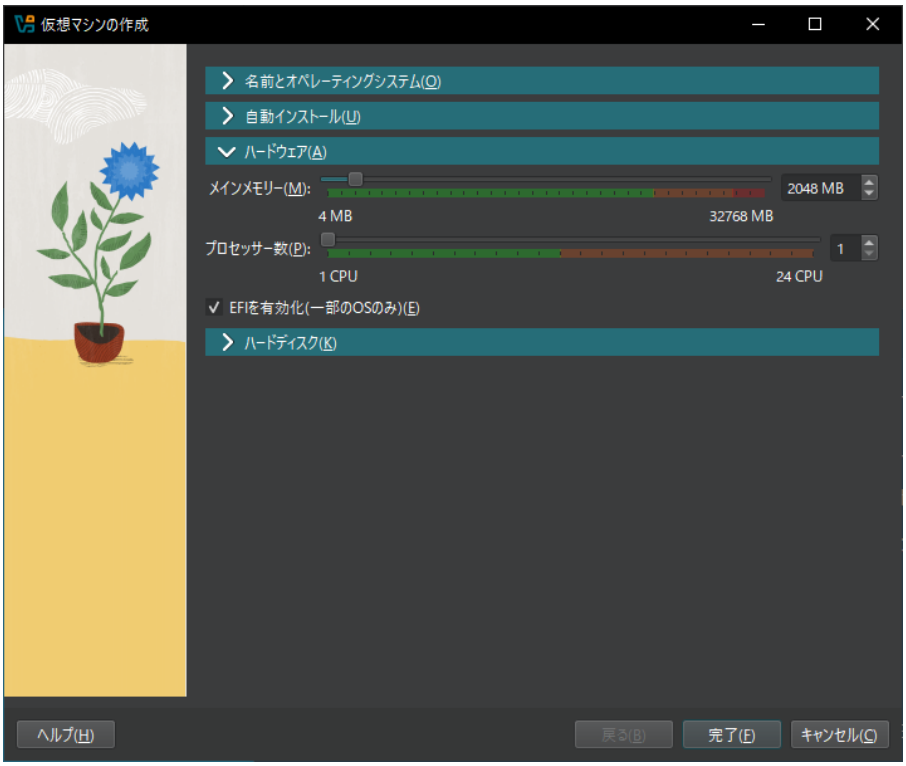

図 11: 仮想マシンのメモリや CPU 設定の画面

メインメモリーの容量は、使用しているコンピュータが搭載しているメモリ容量以下に制限されます。仮想マシンにある程度多めのメインメモリーを割り当てることで快適に動作させることができます。デフォルトの 2048MB でも動作しますが、余裕があれば 4096MB（4GB）割り当ててもよいでしょう。

プロセッサの数は、使用しているコンピュータが搭載しているプロセッサのコア数以下に制限されます。また、プロセッサによっては仮想 CPU 機能により見かけ上コア数が 2 倍になっている場合もあります。デフォルトの 1 つでも動作しますが、余裕があれば 2 つ割り当ててもよいでしょう。

「EFI を有効化」は、デフォルトではチェックされていません。OS のインストール画面のサイズが合わず一部の画面が見えなくなる問題が発生するため、チェックしておきます。

設定しましたら、「＞ハードディスク (K)」をクリックします。

2.4.3　仮想ハードディスクの設定

OS のインストール先となる仮想ハードディスクを作成します。

項目	設定値
ハードディスクファイルのサイズ	20.00GB

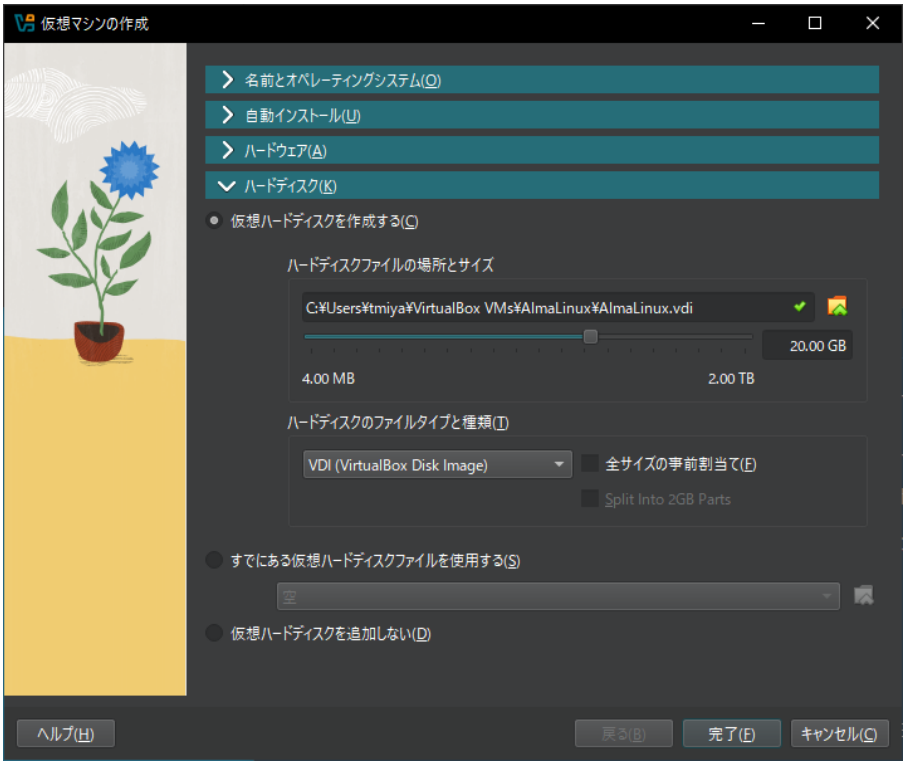

図 12: 仮想ハードディスクの設定画面

仮想ハードディスクの容量は、使用しているコンピュータが搭載しているストレージの容量以下に制限されます。また、「全サイズの事前割当て (F)」をチェックしない限り、仮想ハードディスクの容量はゲスト OS が使用した分だけしか消費しませんので、サイズの設定は最大消費容量を指定することになります。

設定が終わったら、「完了」ボタンをクリックします。

VirtualBox マネージャーに、新しく作成した仮想マシンが追加されたことを確認します。

図 13: 仮想マシン追加の確認

2.5　ネットワークの追加

ホスト OS と仮想マシン上のゲスト OS がネットワークで通信できるようにするため、以下の手順で「ホストオンリーアダプター」を追加します。

デフォルトでは NAT を使った外部ネットワークに接続するネットワークのみ設定されています。

2.5.1　仮想ネットワークの設定を確認

VirtualBox の仮想ネットワークの設定を確認します。

VirtualBox マネージャーの「ツール」右側のリストボタン（横 3 本線）をクリックし、「ネットワーク」を選択します。

「ホストオンリーネットワーク」タブを選択し、「VirtualBox Host-Only Ethernet Adapter」が存在していることを確認します。IP アドレスは、VirtualBox のインストール時に自動的に作成された場合には「192.168.56.1/24」が設定されています。この IP アドレスは、ホスト OS に作成された仮想ネットワークアダプターに割り当てられている IP アドレスです。

DHCP サーバーもデフォルトで有効になっており、インストールしたゲスト OS は自動的に IP アドレスが設定されます。

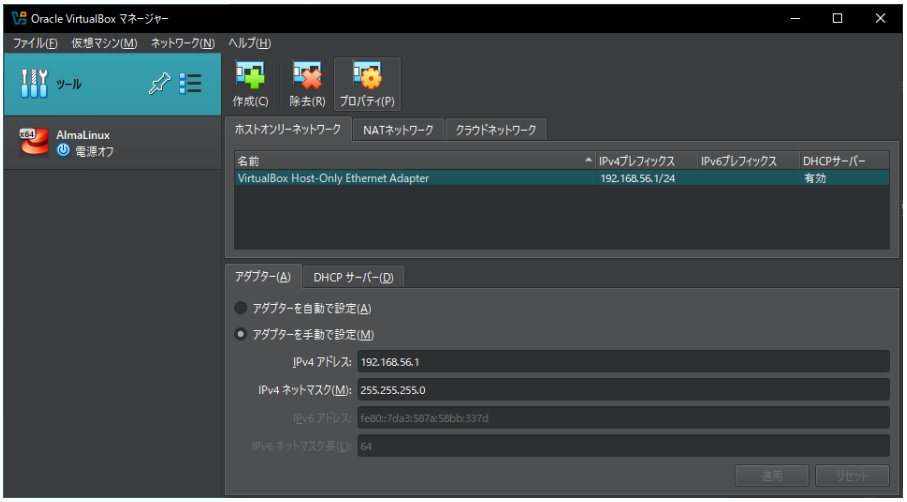

図 14: 仮想ネットワークの設定画面

2.5.2 仮想マシンにネットワークアダプターを追加

仮想マシンにネットワークアダプターを追加します。

VirtualBox マネージャーから仮想マシンを選択し、右側の各種設定情報から「ネットワーク」を選択します。

仮想マシンには最大 4 つのネットワークアダプターを設定できます。アダプター 1 は「NAT」が設定されており、外部のネットワークやその先にあるインターネットに接続できます。

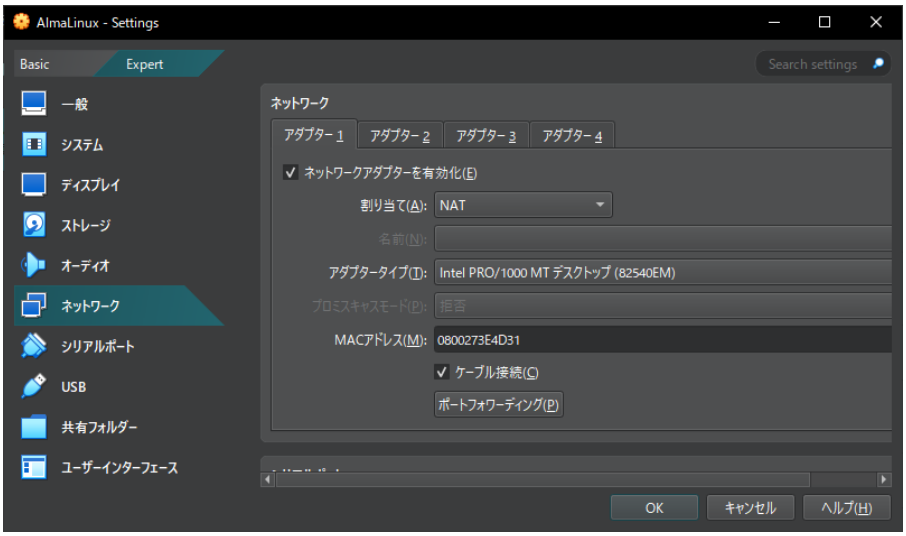

図 15: 仮想マシンのネットワークアダプター 1 の設定画面

アダプター 2 のタブを選択し、「ネットワークアダプターを有効化 (E)」をチェックします。「割り当て (A)」を「ホストオンリーアダプター」に設定すると、「名前 (N)」に「VirtualBox Host-Only Ethernet Adapter」が設定されます。これによって、この仮想マシンはホスト OS と通信が行えるようになります。

設定したら、「OK」をクリックします。

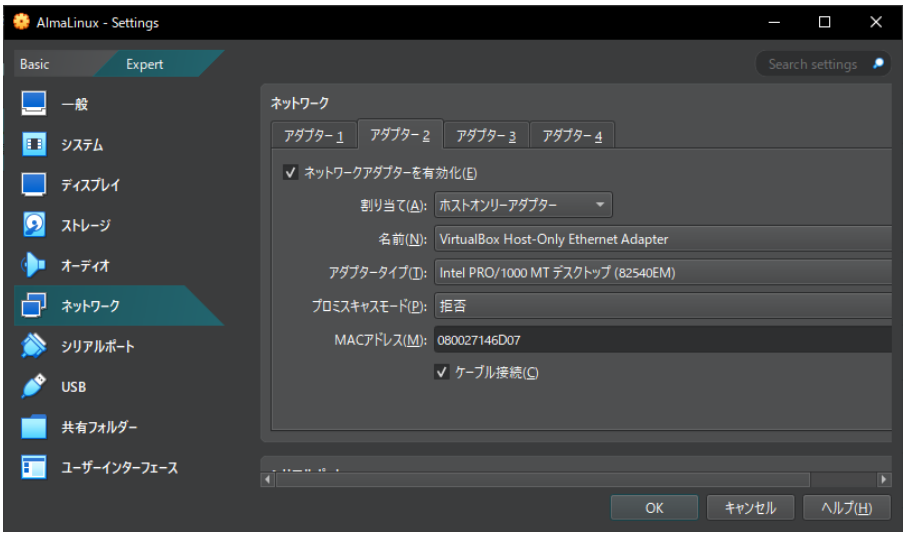

図 16: 仮想マシンのネットワークアダプター 2 の設定画面

以上で仮想マシンの作成と設定は完了です。

2.6 ホストキーによるホスト OS の操作への復帰

仮想マシンをマウスで操作している際に、ホスト OS にマウス操作を戻したくなった場合にはホストキーを押します。デフォルトではキーボード右側の Ctrl キーがホストキーに設定されています。ホストキーの設定は、仮想マシンのウインドウの右下にも表示されています。

2.6.1 ホストキーが押せない場合のホスト OS の操作への復帰

ノートパソコンなどキーボードによっては右 Ctrl キーが無いためホストキーが押せず、仮想マシンの操作から抜け出せなくなります。そのような状態になった場合には、Ctrl+Alt+Del キーを入力すると強制的にホスト OS である Windows 側に制御が移ります。「キャンセル」ボタンをクリックすれば、仮想マシンの操作から抜け出した状態でホスト OS の操作が可能になります。

2.6.2 ホストキーの変更

VirtualBox マネージャーの環境設定から、ホストキーを変更できます。

ホストキーの変更は、仮想マシンの作成の後でないと行えないので、初めて VirtualBox をインストールした場合には、仮想マシンの作成を行ってから設定変更を行ってください。

環境設定は、VirtualBox マネージャーの「ファイル (F)」メニューから「環境設定 (P)」を選択して呼び出します。「環境設定」ウインドウが表示されたら、左側の設定項目から「Expert」に切り替えて「入力」を選択し、「仮想マシン (M)」タブを選択します。一番最初にある「Host Key Combo」の「ショートカット」部分をクリックします。新たにホストキーにしたいキーを押すことで、ホストキーがそのキーに切り替わります。

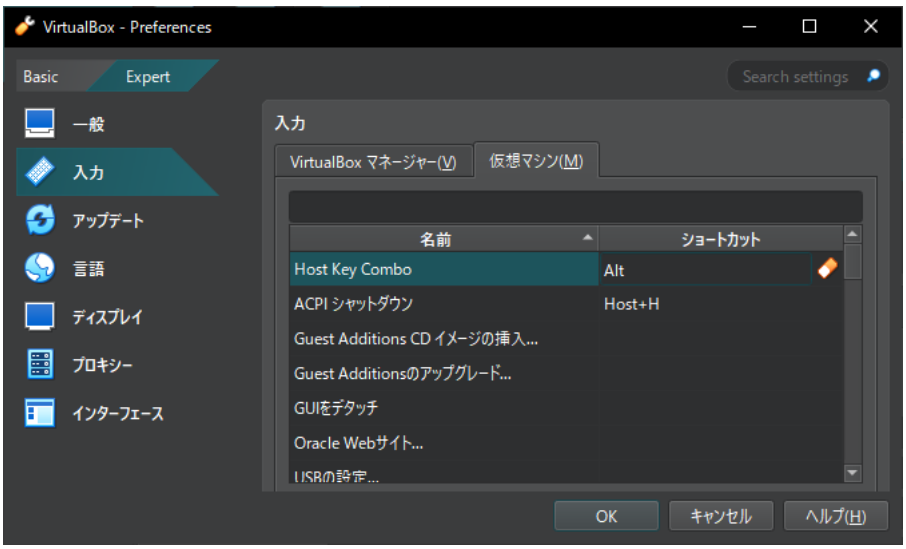

図 17: ホストキーの設定画面

ホストキーに設定できるのは Ctrl キー、Alt キー、Windows キー、Shift キー（単独設定不可）です。左の Ctrl キーをホストキーに設定してしまうと、コマンドライン操作などの際に重複してしまうので、避けた方がよいでしょう。複数のキーを同時に押す組み合わせをホストキーにすることもできるので、2 つないし 3 つのキーの組み合わせをホストキーとして設定するとよいでしょう。

3 Linux のインストールと設定

実習環境を用意するには、作成した仮想マシンに OS をインストールし、初期設定が必要です。

本章では、VirtualBox で作成した仮想マシンに AlmaLinux をインストールします。OS インストール後、初期設定とネットワーク接続の確認を行います。

本章の内容

- 利用する Linux のディストリビューション
- インストール用 ISO イメージの入手
- ISO イメージのファイル名
- ISO イメージを仮想光学ドライブで読み込む設定
- 仮想マシンの起動
- OS のインストール

3.1 利用する Linux のディストリビューション

本教科書では、AlmaLinux 9.4 の Intel/AMD x86_64 アーキテクチャに対応したバージョンを利用します。

```
https://almalinux.org/ja/
```

図 18: AlmaLinux の Web サイト

AlmaLinux は、商用ディストリビューションである Red Hat Enterprise Linux をベースにしたディストリビューションとして提供されています。利用に際し費用が発生しない、無償で提供されているディストリビューションです。

3.2 インストール用 ISO イメージの入手

AlmaLinux OS Foundation が配布している ISO イメージを、ダウンロードします。仮想マシンは、仮想光学ドライブに ISO イメージをセットすることでインストールが行えるので、インストール用の DVD/USB メモリを作成する必要がありません。

3.2.1 ダウンロード方法

ISO イメージをダウンロードするには、以下の URL にアクセスします。

```
https://mirrors.almalinux.org/isos/x86_64/9.4.html
```

この URL にアクセスすると、多くのミラーサイトが表示されます。例えば、IIJ が提供しているミラーサイトの URL であれば次のようになります。

```
http://ftp.iij.ad.jp/pub/linux/almalinux/9.4/isos/x86_64
```

3.2.2 URL が利用できない場合

URL はサイト構造の変更によって変わっている場合があります。その場合には、AlmaLinux の Web サイトからリンクを辿ってダウンロードサイトを探してください。

図 19: AlmaLinux のダウンロードサイト

3.3 ISO イメージのファイル名

ISO イメージは以下のようなファイル名になっています。

```
AlmaLinux-バージョン-アーキテクチャ-イメージの種類.iso
```

今回使用する ISO イメージのファイル名は「AlmaLinux-9.4-x86_64-dvd.iso」になります。

3.3.1 バージョン

本教科書では、本教科書の作成時点で最新であった AlmaLinux 9.4 を利用して解説しています。

今後のバージョンアップで、より新しいバージョンの AlmaLinux が入手可能になっているかもしれません。バージョン 9 系であればマイナーバージョンによる大きな差は無いと推測されますが、セキュリティ対応などの関係で動作が変わる場合があります。

初めて本教科書の内容を学習する際には、まずはバージョンを合わせて動作を確認し、その後異なるバージョンで同様に動作するか確認してみてください。バージョンによる動作の違いについては、本教科書の情報交換を行う Slack で情報共有を行い、本教科書の今後のバージョンアップで対応していく予定です。

3.3.2 アーキテクチャ

Linux カーネルは様々な種類の CPU アーキテクチャに対応しています。アーキテクチャによってバイナリが異なるため、アーキテクチャに合わせた ISO イメージを選択する必要があります。主なアーキテクチャには以下のものがあります。

- x86_64 Intel や AMD の CPU アーキテクチャです。64 ビット版になります。
- aarch64 ARM の CPU アーキテクチャです。64 ビット版になります。ARM を搭載した Windows や、Apple Silicon を搭載した Mac を使う場合にはこちらをダウンロードします。

3.3.3 ISO イメージの種類

ISO イメージには、いくつかの種類があります。インストールの目的によってイメージを選択します。AlmaLinux には以下の 3 種類の ISO イメージが用意されています。

- 起動のみ（boot.iso）
- 最低限（minimal.iso）
- フルパッケージ（dvd.iso）

本教科書の演習では、GUI 環境も使用するため、フルインストールが可能な ISO イメージ（dvd.iso）を使ってインストールします。

3.4 ISO イメージを仮想光学ドライブで読み込む設定

ダウンロードした OS インストール用の ISO イメージを、仮想マシンの仮想光学ドライブで読み込む設定をします。

VirtualBox マネージャーから仮想マシンを選択し、右側の各種設定情報から「ストレージ」を選択します。

仮想マシンには、IDE と SATA の 2 種類のストレージデバイスが接続されています。仮想光学ドライブは IDE に接続されています。読み込む設定が何もされていない場合、「空」と表示されているのが仮想光学ドライブですので、選択します。

「光学ドライブ:IDE セカンダリデバイス 0」の右側にある円形のボタンをクリックし、「Choose a Disk File…」を選択すると、ファイルダイアログが開きます。準備しておいた OS インストール用の ISO イメージを選択し、「開く」ボタンをクリックします。「空」の表示がファイル名に変わります。

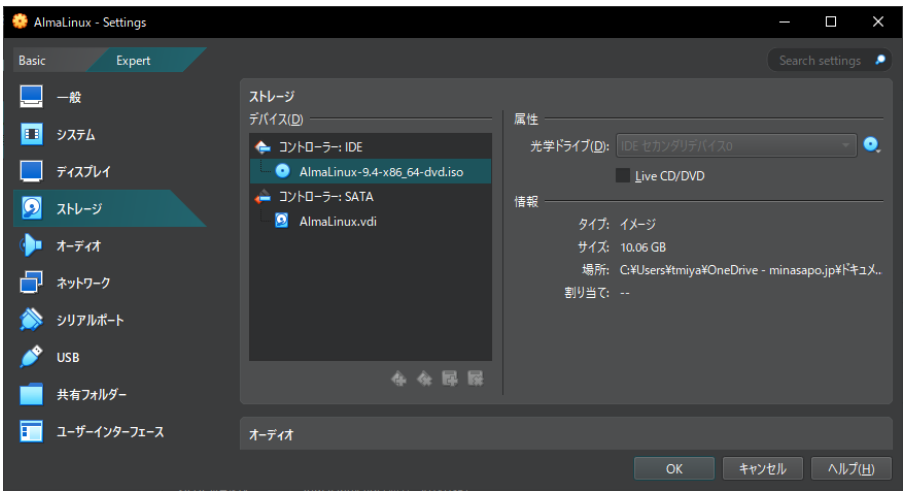

図 20: 仮想マシンの仮想光学ドライブの設定画面

3.5　仮想マシンの起動

VirtualBox マネージャーで仮想マシンを選択し、「起動」をクリックします。別ウインドウで仮想マシンが起動します。

ISO イメージからインストーラーが読み込まれて、OS のインストールを行えるようになります。

3.5.1　ISO イメージファイルのマウント

あらかじめ仮想マシンの仮想光学ドライブに ISO イメージを読み込ませる設定をしていない場合、起動用の ISO イメージを設定するダイアログが表示されます。

画面の指示に従って、ISO イメージのファイルを指定し、「マウントとブートのリトライ」ボタンをクリックして起動します。

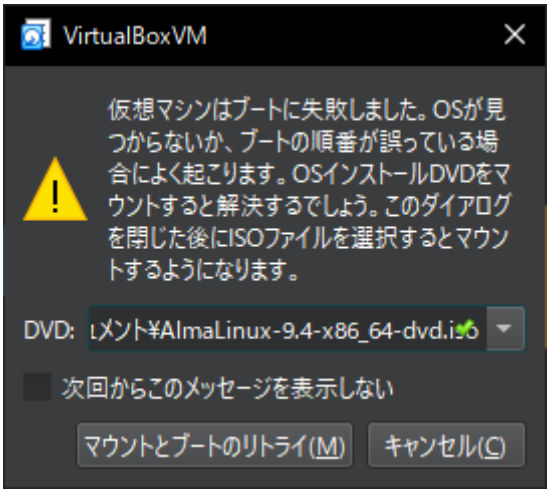

図 21: マウントとブートのリトライダイアログ画面

3.6　OS のインストール

以下の手順に従って、OS のインストールを行います。

3.6.1　インストーラー起動オプションの選択

OS インストール用の ISO イメージを読み込んで起動すると、最初に GRUB というブートローダーが起動します。ここでインストーラーの起動オプションを選択できます。

デフォルトの「Test this media & Install AlmaLinux 9.4」で起動し、ISO イメージが壊れていないかテストを行った上でインストールを行います。すでに選択されているので、Enter キーを押します。

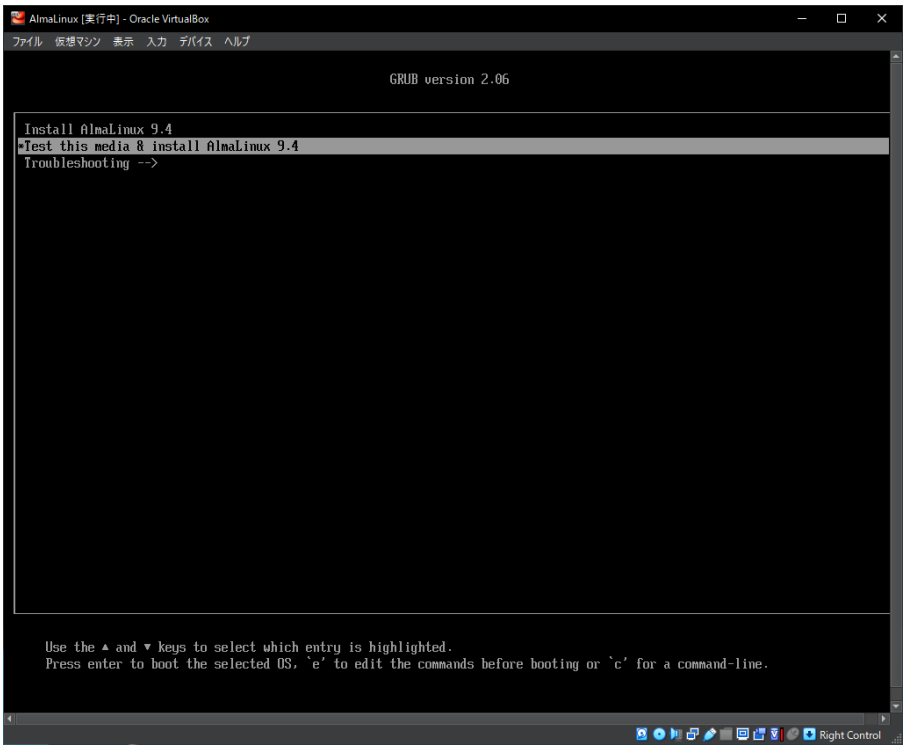

図 22: GRUB 画面

3.6.2　言語選択

言語の選択画面が表示されるので、左側のメニューから「日本語 Japanese」を選択します。画面右側のメニューに「日本語（日本）」が表示されます。

「続行」ボタンをクリックします。

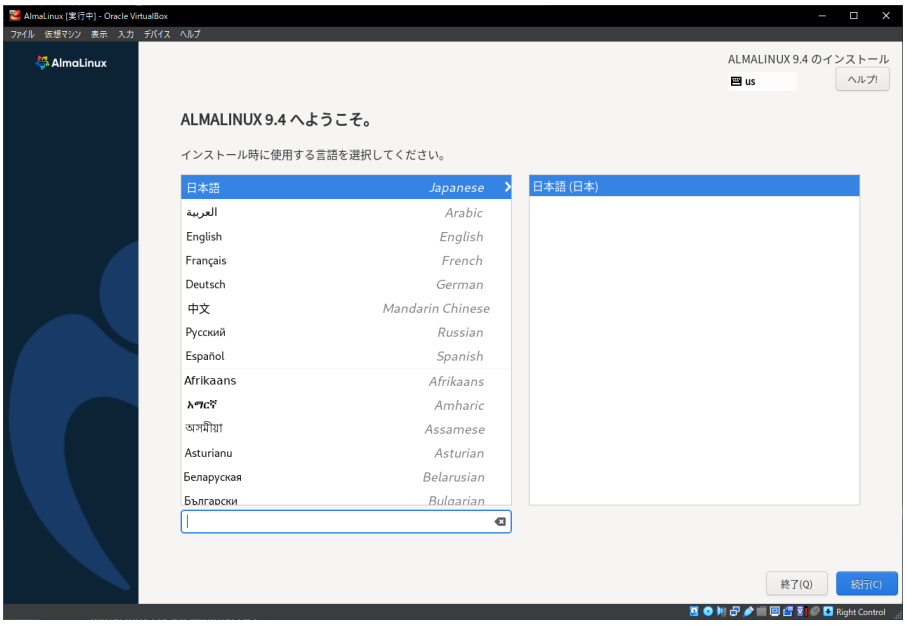

図 23: 言語選択画面

3.6.3　インストール概要

インストール概要の画面では、各種設定がまとめて表示されます。

「!」が付いた項目は、必ず設定が必要な項目です。

図 24: インストール概要画面

3.6.4　インストール先の設定

「インストール先」をクリックします。画面が切り替わり、「デバイスの選択」や「ストレージの設定」が行えます。

デフォルトで、仮想マシン作成時に新規作成した仮想ハードディスクがローカル標準ディスクとして選択されています。また、そのデバイスのパーティション設定は自動構成が選択されています。

今回はデフォルトのままインストール先を設定するので、そのまま「完了」ボタンをクリックします。

図 25: インストール先の設定画面

3.6.5 ソフトウェアの追加インストール

「ソフトウェアの選択 (S)」をクリックします。画面が切り替わり、大まかな用途を決める「ベース環境」と、追加でインストールするソフトウェアが選択できます。実習で利用するソフトウェアをあらかじめインストールしておきます。

ベース環境は「サーバー（GUI 使用）」を選択します。右側の「選択した環境用のその他のソフトウェア」リストから、インストールしたいソフトウェアとして「ベーシック Web サーバー」をチェックして追加でインストールします。

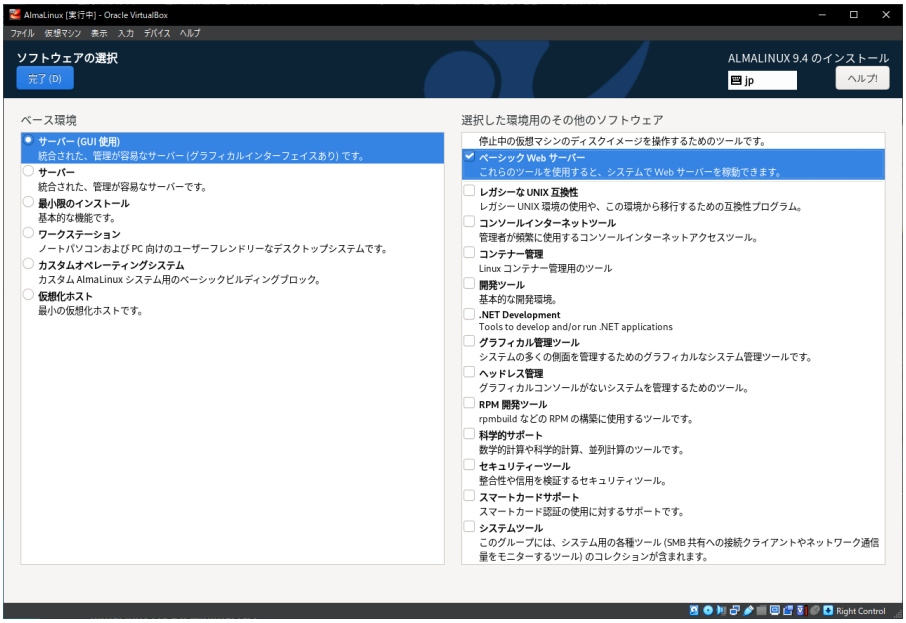

図 26: ソフトウェアの追加選択画面

3.6.6　ユーザーの作成

「ユーザーの作成 (U)」をクリックします。実習で使用するユーザーを作成します。

項目	設定値
フルネーム	LinuC
ユーザー名	linuc
このユーザーを管理者にする	チェックする
このアカウントを使用する場合にはパスワードを要求する	チェックする
パスワード	任意のパスワードを設定
パスワードの確認	設定したパスワードを再入力

パスワードは、入力間違えが無いように「パスワード (P)」と「パスワードの確認 (C)」に同じパスワードを 2 回入力します。長さや大文字小文字数字を混ぜるなど一定の要件を満たさないと脆弱なパスワードと判定されます。良好以上になるパスワードを入力するか、脆弱と判定された場合には「完了 (D)」ボタンを 2 回クリックする必要があります。

ユーザーの作成の設定が完了したら、「完了 (D)」ボタンをクリックします。

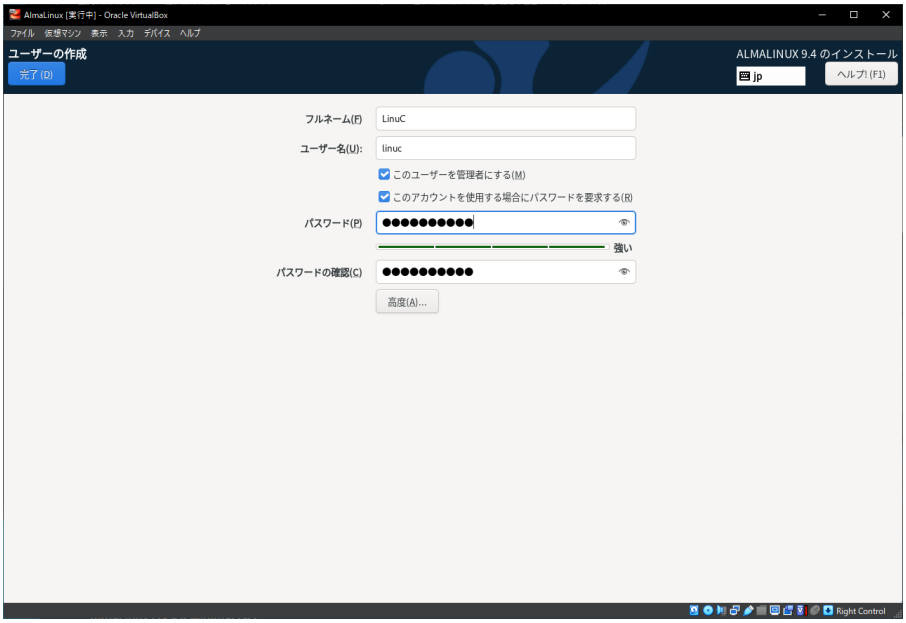

図 27: ユーザーの作成の設定画面

3.6.7　インストールの開始

必要な設定が終わったら、インストールを開始します。「インストールの開始 (B)」ボタンをクリックします。

図 28: インストール開始前の画面

インストールが始まり、「インストール状況の表示」画面が表示されます。

図 29: インストール終了の画面

インストールが終了したら、「システムの再起動」ボタンをクリックして、再起動します。

4 Linux を操作してみよう

Linux のインストールが終わったら、コマンド操作などについて学んで行く前に、簡単に操作をしてみましょう。

本章では、ログインからコマンド実行、Web サーバーを動かすための手順と、実習を行いやすくするための SSH によるリモートログインについて解説します。

本章の内容

- GUI によるログインとログアウト
- コマンド実行のための端末を起動
- Web サーバーを動かしてみる
- SSH でリモートログインする

4.1 GUI によるログインとログアウト

Linux を使い始めるにはログインを、使い終わったらログアウトを行います。

システムが起動したら、ログインとログアウトを試してみましょう。

4.1.1 GUI でログインする

ログインするには、ログインしたいユーザーをクリックし、パスワードを入力します。

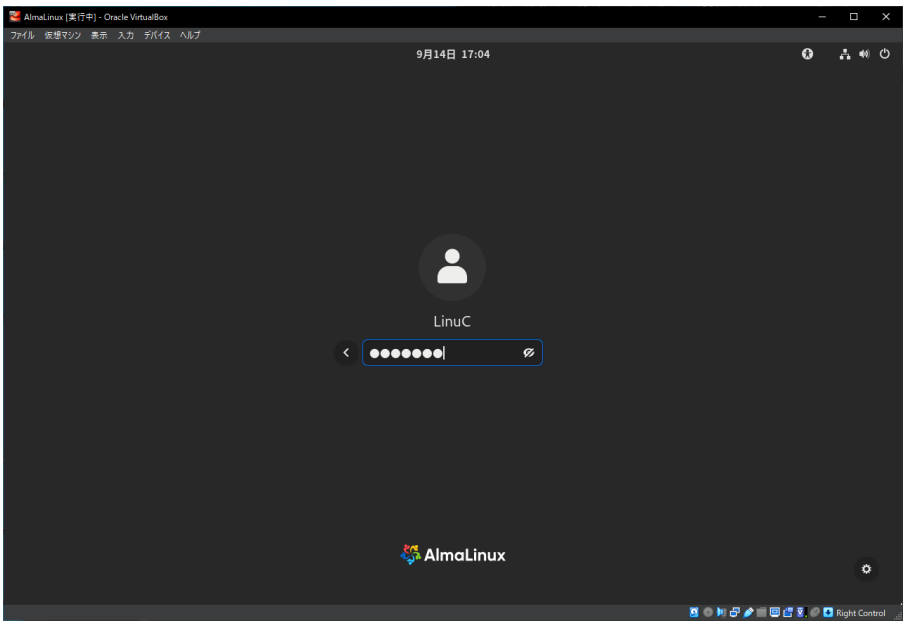

図 30: ログインの画面

4.1.2　初回ログイン時のツアー

初回ログイン時には「AlmaLinux へようこそ」と表示され、操作方法を確認するツアーを開始できます。ツアーを見たい場合には「ツアーをチェックする」ボタンをクリックします。はじめて AlmaLinux の GUI を操作するのであれば、短いツアーですのでチェックしてみてください。

図 31: AlmaLinux へようこその画面

4.1.3　GUI でログアウトする方法

ログアウトするには、画面右上にあるメニューバーの電源アイコンをクリックし、「電源オフ/ログアウト」をクリックすると表示される「ログアウト」をクリックします。

ログアウトを確認するダイアログが表示されるので、「ログアウト」をクリックします。

4.2　コマンド実行のための端末を起動

コマンドを実行して Linux を操作するために「端末」アプリを起動します。

再度ログインを行い、画面左上の「アクティビティ」をクリックし、画面下に表示されるアイコンから「端末」アプリのアイコンをクリックします。

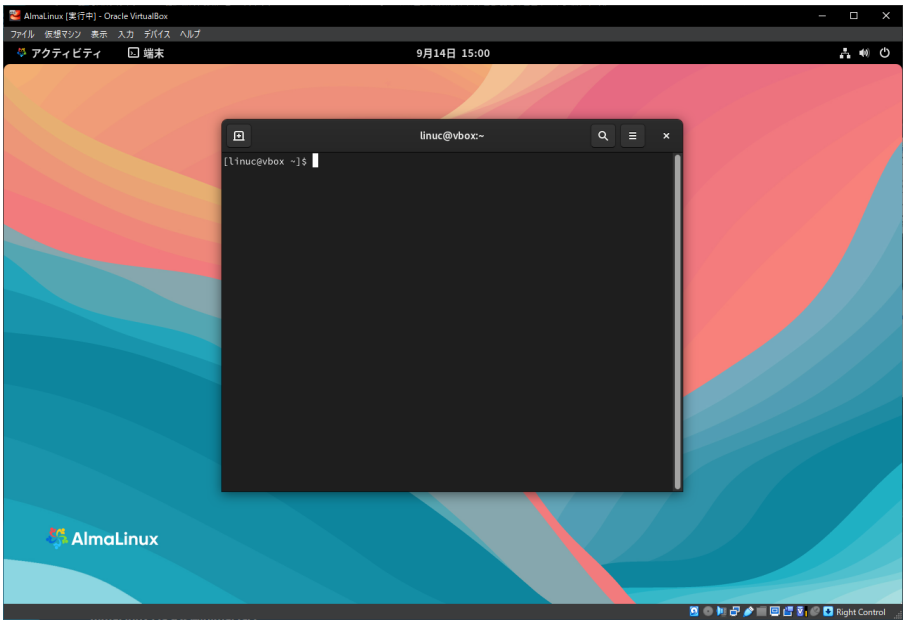

図 32: 端末起動の画面

4.3　Web サーバーを動かしてみる

Linux では、コマンドを実行して各種サービスの起動や停止を行います。ここでは、OS インストール時に追加で導入した Web サーバーを起動して、Web ブラウザからアクセスしてみます。

4.3.1　Web サーバーを起動する

Web サーバーを起動します。Web サーバーは、AlmaLinux では httpd サービスと呼ばれています。サービスの起動や停止は systemctl コマンドを実行して行います。

Web サーバーのようなサービスは、実行するためには Linux のシステム管理者のユーザー権限が必要になります。システム管理者やユーザー権限については第 7 章で解説します。

「端末」アプリのコマンドプロンプトで、systemctl コマンドを実行します。

```
$ systemctl start httpd
```

systemctl コマンドを実行すると、システム管理者のユーザー権限があることを確認するためにユーザー linuc のパスワードを要求するダイアログが表示されるので、パスワードを入力します。OS インストール時の「ユーザーの作成」でユーザー linuc を作成する際に「このユーザーを管理者にする」をチェックしているので、ユーザー linuc のパスワードを入力することでシステム管理者のユーザー権限があることを確認できます。

4.3.2　Web サーバー起動の確認とコマンド履歴

認証に成功すると、httpd サービスが起動します。実行結果が何も表示されていないので、起動されたことを確認するため、systemctl status コマンドを実行します。

コマンドプロンプトには、コマンド履歴という機能があります。以前に実行したコマンドを呼び出してそのまま実行したり、修正して実行したりできます。カーソルキーの上下でコマンド履歴を呼び出し、修正する場合にはカーソルキーの左右でカーソルを動かし、Delete キーや BackSpace キーで文字を消すことができます。

カーソルキーの上を押して「systemctl start httpd」コマンドを呼び出し、「start」を消して「status」に書き換えて実行します。

```
$ systemctl start httpd
```

↓ カーソルを左に移動して rt を削除

```
$ systemctl sta httpd
```

↓ tus を追加して実行

```
$ systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: d>
     Active: active (running) since Mon 2024-10-21 16:46:51 JST; 1s ago
       Docs: man:httpd.service(8)
   Main PID: 8248 (httpd)
     Status: "Started, listening on: port 443, port 80"
      Tasks: 178 (limit: 10104)
     Memory: 32.5M
        CPU: 34ms
     CGroup: /system.slice/httpd.service
             ├─8248 /usr/sbin/httpd -DFOREGROUND
             ├─8249 /usr/sbin/httpd -DFOREGROUND
             ├─8250 /usr/sbin/httpd -DFOREGROUND
             ├─8251 /usr/sbin/httpd -DFOREGROUND
             ├─8252 /usr/sbin/httpd -DFOREGROUND
             └─8253 /usr/sbin/httpd -DFOREGROUND

10月 21 16:46:51 localhost.localdomain systemd[1]: Starting The Apache HTTP Ser>
10月 21 16:46:51 localhost.localdomain httpd[8248]: AH00558: httpd: Could not r>
10月 21 16:46:51 localhost.localdomain systemd[1]: Started The Apache HTTP Serv>
10月 21 16:46:51 localhost.localdomain httpd[8248]: Server configured, listenin>
```

実行結果の「Active:」の項目が「active (running)」になっていれば、httpd サービスが実行中であることがわかります。サービスはいくつかのプロセスとしてバックグラウンドで実行されています。プロセスについては第 10 章で解説します。確認を終えたら「q」（Quit）キーを入力して表示を終了します。

4.3.3 Web ブラウザで Web サーバーにアクセスする

Web ブラウザを起動して、Web サーバーにアクセスしてみます。

画面左上の「アクティビティ」をクリックし、画面下に表示されるアイコンから一番左にある「Firefox」アプリのアイコンをクリックします。

Web ブラウザのウインドウが表示されたら、アドレスに「localhost」と入力して、Enter キーを押します。「AlmaLinux Test Page」が表示されたら、Web サーバーにアクセスできたことが確認できます。

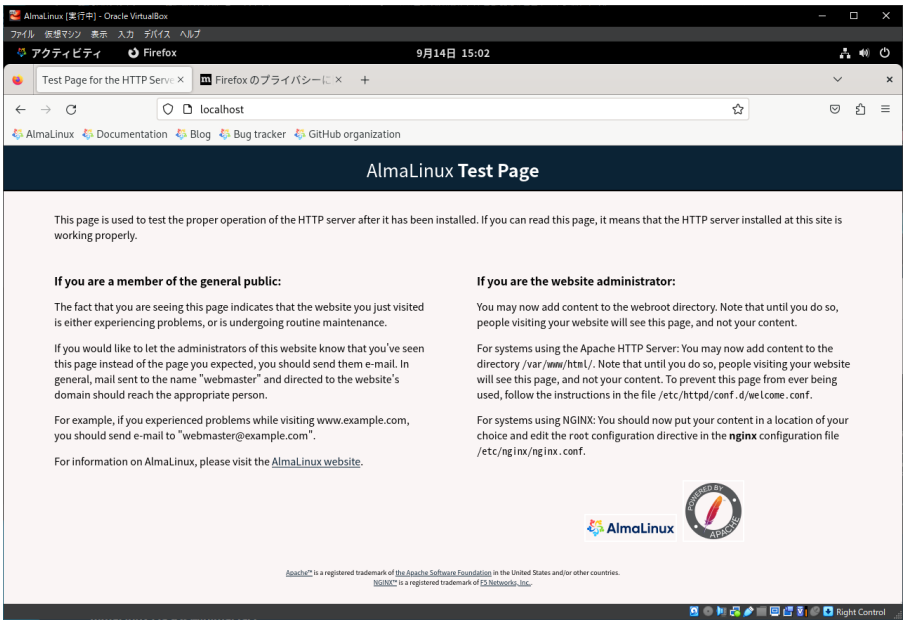

図 33: テストページ画面

4.3.4 Web サーバーを停止してみる

Web サーバーを停止してみます。

コマンドプロンプトで、systemctl stop コマンドを実行します。コマンド履歴とコマンドライン編集を使って実行してみてください。

```
$ systemctl stop httpd
```

再度、ユーザー linuc のパスワード認証が要求されるので、パスワードを入力します。再度 systemctl status コマンドで確認すると、「Active:」の項目が「inactive (dead)」になっています。Web ブラウザで再読み込みを行うと「正常に接続できませんでした」と表示されます。

再度、systemctl start コマンドで起動し直して、正常な状態に戻るのも確認してみてください。

4.4　SSH でリモートログインする

Linux が動作しているマシンが手元にある場合には GUI で直接ログインできますが、多くの場合 Linux が動作するマシンはデータセンターやマシンルームの中に設置されていたり、クラウド上で動作しています。このようにリモートにある Linux を操作するには、SSH でリモートログインする必要があります。

本教科書の実習は基本的に GUI で直接ログインし、「端末」アプリのコマンドプロンプトで進めることができますが、リモート操作に慣れるためにホスト OS からゲスト OS に SSH でリモートログインしてみることをお勧めします。

4.4.1　Linux の IP アドレスを調べる

事前に Linux で ip コマンドを実行して、ログイン先の IP アドレスを調べてみます。

「端末」アプリのウインドウ内にコマンドプロンプトが表示されているので、コマンドを入力し、Enter キーを押して実行します。「ip a」コマンドは、実行したマシンに割り当てられている IP アドレスなどの情報を表示します。

```
$ ip a
```

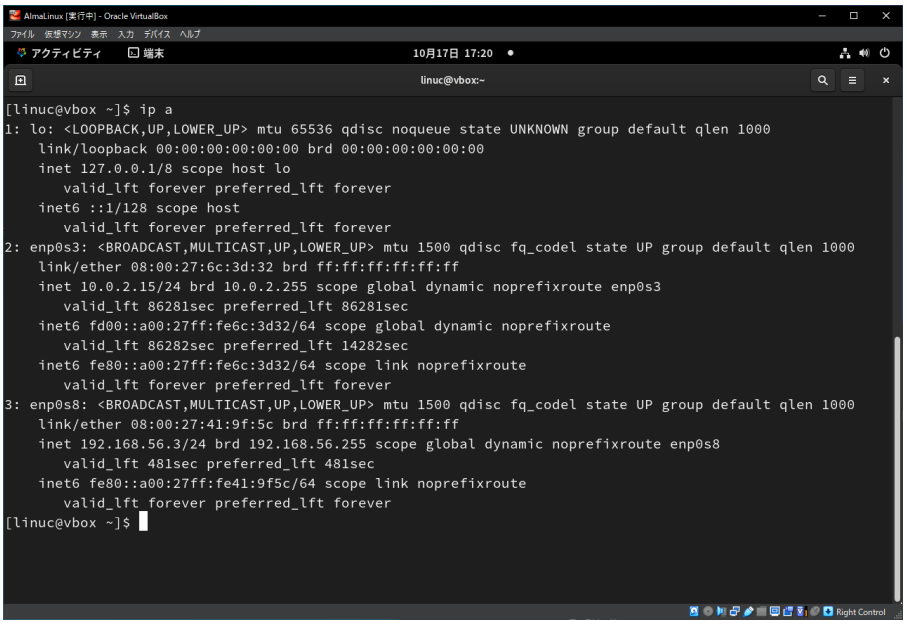

図 34: ip コマンド実行画面

実行した仮想マシンには、ネットワークアダプターを 2 つ設定しています。自分自身に接続する「ローカルループバック」を含めて、3 つのネットワークインターフェースの情報が表示されます。IP アドレスは「inet」で始まる行に記述されています。

```
$ ip a
 （略）
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    group default qlen 1000
    link/ether 08:00:27:a2:d5:45 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.3/24 brd 192.168.56.255 scope global dynamic noprefixroute
        enp0s8
 （略）
```

「192.168.56.x/24」はホストオンリーネットワークに設定したアダプターの IP アドレスです。DHCP で動的に割り当てられるため、第 4 オクテットは変わる場合があります。

ホスト OS とゲスト OS を接続するにはホストオンリーネットワークを経由する必要があります。SSH のログイン先の IP アドレスはこの「192.168.56.x」になります。

4.4.2 Windows から SSH を行う方法

Windows から仮想マシン上で動作する Linux に SSH でリモートログインするには、Windows 側で SSH クライアントを実行する必要があります。方法としては、以下の 2 つの方法があります。

- Windows でコマンドプロンプトや PowerShell を実行し、ssh コマンドを実行する
- Tera Term などの SSH プロトコルをサポートしたソフトウェアを導入する

今回は Windows の標準機能である ssh コマンドを使ってみます。

4.4.3 Windows のコマンドプロンプトを実行する

Windows のコマンドプロンプトを実行します。タスクバーの検索ウインドウに「cmd」と入力し、候補として出てきた「コマンドプロンプト」を実行します。

タスクバーに検索ウインドウが無い場合には、タスクバーを右クリックして「タスクバーの設定」から、検索ウインドウを表示するように設定してください。

4.4.4 ssh コマンドでゲスト OS に接続する

コマンドプロンプトで ssh コマンドを実行して、ゲスト OS の Linux に接続します。

```
書式
ssh ユーザー名@接続先IPアドレス
```

以下の例では、ユーザー linuc で 192.168.56.3 に接続しようとしています。

```
>ssh linuc@192.168.56.3
The authenticity of host '192.168.56.3 (192.168.56.3)' can't be established.
ECDSA key fingerprint is SHA256:/yjso78Rqa2Sv+UWJ/k8ofOrrT0dFWdX2+Efyuef8qY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
    ※yesと入力
Warning: Permanently added '192.168.56.3' (ECDSA) to the list of known hosts.
linuc@192.168.56.3's password: ※ユーザーlinucのパスワードを入力（非表示）
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Thu Oct 17 17:22:16 2024 from 192.168.56.1
[linuc@vbox ~]$
```

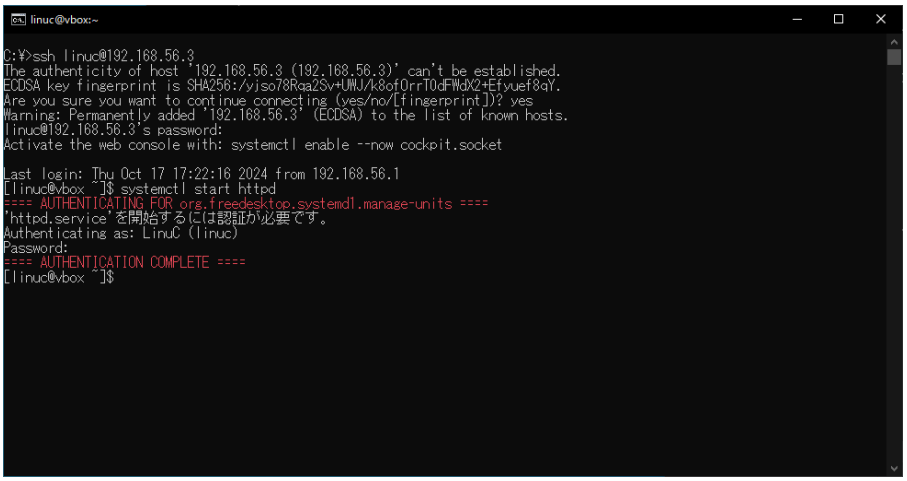

図 35: リモートログイン画面

初めて SSH で接続する場合、接続先のホスト証明書が送られてきます。接続を続ける場合には「yes」と入力します。その後、ユーザー linuc のパスワードが要求されるので、パスワードを入力すると接続が完了します。

コマンドプロンプトが表示されれば、GUI の「端末」アプリで実行するのと同じようにコマンドを実行できます。

5　基本的なコマンド

Linux をコマンドで操作するには、ファイルやディレクトリの操作を行えることが基本となります。

本章では、ファイルやディレクトリを操作するコマンドや Linux を利用する上で必要な基本的なコマンドを学習します。

本章の内容

- ファイルとディレクトリの参照
- ファイルの内容を表示
- 空のファイルを作成する
- ディレクトリの操作
- 特別なディレクトリ
- ファイルのコピー (cp コマンド)
- ファイルの移動 (mv コマンド)
- ファイルの削除 (rm コマンド)
- ファイルの検索（find コマンド）
- コマンドのパスを表示（which コマンド）
- コマンドのヘルプを表示する
- マニュアルの使い方

5.1　ファイルとディレクトリの参照

Linux は、コマンドやプログラム、設定ファイルなど様々なファイルでシステムを構成しています。また、それらのファイルを目的別に分類するため、様々なディレクトリに分けて格納しています。これらのファイル、ディレクトリを参照するのは、Linux 操作の基本です。

ファイルやディレクトリの一覧を取得するには ls(LiSt) コマンドを使用します。

5.1.1　ファイルとディレクトリ

コンピュータの基本的な作業は、情報（＝データ）を処理することにあります。データをコンピュータ上に保存しておく方法として、ファイルという形式が用いられます。文書を作成したら文書ファイルを保存しますし、デジカメで写真を撮ったら画像ファイルを保存する、といった具合です。

ファイルの数が増えると、内容や用途といった目的別に分ける必要が出てきます。このときに必要なものが、ディレクトリという考え方です。ディレクトリは、複数のファイルをまとめて保存できます。また、ディレクトリの中にディレクトリを作成することも可能です。

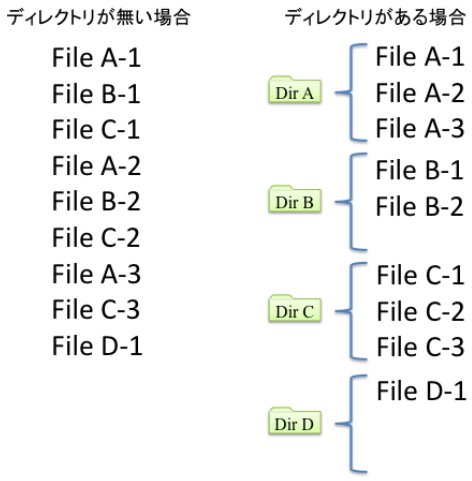

図 36: ファイルとディレクトリ

5.1.2 ls コマンドによるファイルやディレクトリの参照

ls コマンドでは、引数（ひきすう）として指定したファイルやディレクトリの情報を取得できます。ディレクトリを指定した場合、そのディレクトリの中にあるファイルやディレクトリの一覧を取得できます。引数を省略した場合、カレントディレクトリの情報を取得します。

```
書式
ls [オプション] [ファイルまたはディレクトリ]

オプション
-a
.で始まる隠しファイル等もすべて(All)出力します。

-l
詳細(Long)形式で出力します。

-R
指定したディレクトリ以下すべてを出力します。
このオプションのように「指定したディレクトリ以下すべて」という動作を
「再帰」（recursive）と呼びます。同じような再帰の動作をするオプションは
-R、または-rになることが多くあります。
```

現在作業しているディレクトリ（カレントディレクトリ）内にあるファイルやディレクトリの一覧を表示してみましょう。いくつかのディレクトリがあることがわかります。

```
$ ls
ダウンロード デスクトップ ビデオ 画像
テンプレート ドキュメント 音楽 公開
```

ls コマンドの後に引数としてファイルやディレクトリを指定した場合、指定したディレクトリ内にあるファイルやディレクトリを一覧表示できます。以下の実行例では/usr ディレクトリ内を参照しています。

```
$ ls /usr
bin  games  include  lib  lib64  libexec  local  sbin  share  src  tmp
```

5.1.3 書式

書式とは、そのコマンドが用いられる形式を示しています。コマンド実行時に指定できるオプションや引数などの要素を、どのように指定するかを示しています。

本教科書では、初めて出てくるコマンドには書式を記述しています。[] でくくられている場合、そのオプションや引数は省略可能です。省略された場合、デフォルトの設定が暗黙に指定されて動作します。

本教科書に記述している書式やオプションは、実習を行う上での最低限のものになっています。より詳しい書式やオプションについてはマニュアル等を参照してください。

5.1.4 オプション

コマンドはオプションを指定することで、その動作を変更できます。

オプションには、-（ハイフン）に 1 文字だけ指定する形式、ハイフンが不要な形式、ハイフンが 2 つに単語で指定する形式があります。使用可能なオプション形式はコマンドによって異なります。

本教科書では、ある動作をさせるためのコマンドとしてオプションまでまとめて表記する場合があります。

5.1.5 サブコマンド

比較的新しく作られたコマンドでは、サブコマンドを指定して動作を変更するものもあります。サブコマンドという名称自体、定義された正式な名称ではありませんが、本教科書では便宜上このように呼んでいます。

たとえば、サービスの起動や停止を行う systemctl コマンドは、start や stop などのサブコマンドを指定して実行します。

```
書式
systemctl [オプション] サブコマンド ユニット

サブコマンド
start
指定したサービスユニットを開始する

stop
指定したサービスユニットを停止する

status
指定したサービスユニットの状態を表示する
```

systemctl はサブコマンドを省略できません。このような場合、本教科書では「systemctl start コマンド」のようにサブコマンドまでまとめて表記する場合があります。

ユニットとは systemctl コマンドが操作の対象とするもので、たとえば httpd サービスユニットなどがあります。次に説明する引数に相当します。

5.1.6 引数（ひきすう）

コマンドの実行対象となるファイルやディレクトリなどを引数として指定します。引数を省略した場合、デフォルトの値が適用されます。引数を省略できないコマンドや、引数を必要としないコマンドもあります。

5.1.7 ls -l コマンドの実行

-l オプションを付けて ls コマンドを実行した場合、詳細な形式で出力されます。

```
$ ls -l
合計 0
drwxr-xr-x. 2 linuc linuc 6  8月 10 15:09 ダウンロード
drwxr-xr-x. 2 linuc linuc 6  8月 10 15:09 テンプレート
（略）
```

5.1.8 ドットファイルの表示

Linux では、ファイル名が「.」（ドット）で始まるドットファイルという形式があります。ドットファイルは通常の ls コマンドでは表示されない隠しファイルとなっています。ドットファイルは様々なアプリケーションの設定などを補完するために利用されています。

ドットファイルを表示するには、-a オプションを付けて ls コマンドを実行します。以下の例では詳細形式の-l オプションと一緒に実行しています。

```
$ ls -la
合計 24
drwx------. 14 linuc linuc 4096  8月 10 15:09 .
drwxr-xr-x.  3 root  root    19  8月 10 15:05 ..
-rw-r--r--.  1 linuc linuc   18  4月 30 23:14 .bash_logout
-rw-r--r--.  1 linuc linuc  141  4月 30 23:14 .bash_profile
（略）
```

「.」や「..」、「.bash_logout」などのドットファイルが表示されているのが分かります。

5.1.9 オプションの同時指定と順序

オプションは、まとめてもまとめなくても同じ結果になります。さらに、順不同なので入れ替えても同じ結果になります。

```
$ ls -la
$ ls -al
$ ls -l -a
$ ls -a -l
```

ただし、ごく一部のコマンド（tar コマンドなど）では指定方法が決まっている場合があります。マニュアルやヘルプで確認してください。

5.1.10 ワイルドカードを使った絞り込み

ワイルドカードを使うと、引数で指定するファイルやディレクトリの絞り込みが行えます。ワイルドカードには、「*」（アスタリスク）と「?」の 2 種類があります。

ワイルドカード	動作
*	0 個以上の文字列
?	1 文字

まずは/etc ディレクトリ上にあるファイルやディレクトリの一覧を表示してみましょう。

```
$ ls /etc
DIR_COLORS                gnupg                     pm
DIR_COLORS.lightbgcolor   groff                     pnm2ppa.conf
（略）
```

/etc ディレクトリには、多くのファイルやディレクトリが存在していることがわかります。ワイルドカードを使って絞り込んでリスト表示してみましょう。

/etc ディレクトリ内の.conf ファイルを絞り込んで表示するには、以下のように指定します。

```
$ ls /etc/*.conf
anthy-unicode.conf  kdump.conf      nsswitch.conf   sysctl.conf
appstream.conf      krb5.conf       pbm2ppa.conf    updatedb.conf
（略）
```

「*.conf」と指定することで、.conf で終わるファイルがリスト表示されました。

ファイル名の文字数がわかっている場合は「?」を使って絞り込めます。?は一個につき何らかの 1 文字を示します。

以下の例では、ファイル名が「3 文字.conf」のファイルが検索結果として出力されます。

```
$ ls /etc/???.conf
yum.conf
```

yum.conf が条件に合うファイルとして表示されました。

5.2 ファイルの内容を表示

ファイルの内容を表示するコマンドについて説明します。単純にファイルの内容を表示する cat コマンドと、ページ単位で表示する less コマンドがあります。

5.2.1 ファイルの内容を表示 (cat コマンド)

cat（conCATenate）コマンドは、本来は引数に指定された複数のファイルを連結（concatenate）して出力するコマンドですが、ファイルを 1 つだけ指定して内容を表示するコマンドとして使用されることが多いです。

```
書式
cat ファイル名

オプション
-n
行番号を付加して表示します。
```

ホームディレクトリ内のファイル.bashrc を確認してみましょう。

```
$ cat ~/.bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
（略）
```

-n オプションを付けることで、行番号を付けて表示できます。

```
$ cat -n ~/.bashrc
     1  # .bashrc
     2
     3  # Source global definitions
     4  if [ -f /etc/bashrc ]; then
     5      . /etc/bashrc
     6  fi
（略）
```

5.2.2 ページャを使った表示

cat コマンドを使ってファイルの内容を表示するとき、行数が多いと表示が流れてしまいます。コマンドラインを実行する端末は多くの場合表示行数は 25 行に設定されているため、25 行以上の行は流れてしまってスクロールさせないと確認できません。

たくさんの行があっても、画面制御を行なってスクロールを途中で止めてくれる機能のことをページングといい、それを実現するコマンドをページャといいます。

代表的なページャとしては less コマンドがあげられます。less コマンドは、引数にページング表示したいファイルを指定します。引数を省略した場合、標準入力から入力されたものをページング表示します。標準入力は 6 章で解説します。

```
書式
less [ファイル]
```

less コマンドの操作

キー	動作
スペース	次のページに進む
b	前の一画面に戻る
f	次の一画面に進む
↑	前の行に進む
↓	次の行に進む
/単語	単語を検索します。n キーで検索結果をジャンプします
q	ページャコマンドを終了 (Quit) します。

ページャを使って、たくさんの行があるファイルの表示を行います。さまざまなキーを入力して動作を確認した後、「q」キーを入力して終了させてください。

```
$ less /etc/services
```

5.3 空のファイルを作成する

touch コマンドを実行すると、ファイルサイズが 0 バイトの空のファイルを作成できます。

touch コマンドは本来、ファイルの更新日を変更するコマンドですが、存在しないファイルを引数に指定して実行すると空のファイルが作成されます。今後の実習では、ファイル操作のテスト用ファイルを作成するのに使用します。

```
書式
touch ファイル名

$ touch test
$ ls -l test
-rw-r--r--. 1 linuc linuc 0  8月 18 10:41 test
```

test ファイルが作成されました。ファイルの中身は空なので、サイズが 0 バイトになっています。

5.4 ディレクトリの操作

ディレクトリの操作に使われるコマンドとして、pwd、cd、mkdir、rmdir について説明します。

5.4.1 カレントディレクトリの表示 (pwd コマンド)

pwd(Print Working Directory) コマンドは、現在作業を行っているカレントディレクトリを表示します。

以下の例では、ログイン直後、端末を起動した直後のカレントディレクトリを表示しています。

```
$ pwd
/home/linuc
```

カレントディレクトリが/home/linuc であることがわかります。「/」はディレクトリの区切りを示す記号です。

5.4.2 ディレクトリの変更 (cd コマンド)

cd(Change Directory) コマンドを実行すると、現在いるディレクトリを変更することができます。

引数として移動先のディレクトリを指定します。ディレクトリを指定しない場合はホームディレクトリに移動します。引数に「-」（ハイフン）を指定すると、1 つ前のカレントディレクトリに戻ります。

```
書式
cd [ディレクトリ]
```

以下の例では、pwd で現在いる場所を確認しながら cd コマンドでディレクトリの移動を行ないます。

現在のディレクトリを確認します。

```
$ pwd
/home/linuc
```

/usr ディレクトリに移動します。

```
$ cd /usr
$ pwd
/usr
```

引数無しで cd コマンドを実行すると、ホームディレクトリに移動します。

```
$ cd
$ pwd
/home/linuc
```

cd -（ハイフン）コマンドで 1 つ前のディレクトリだった/usr ディレクトリに戻します。

```
$ cd -
$ pwd
/usr
```

cd コマンドだけを実行すると、ホームディレクトリに戻ります。「-」を指定すると、その前にカレントディレクトリだった/usr に戻ります。間違えてカレントディレクトリを変更してしまった場合などに使用するとよいでしょう。

5.4.3 ディレクトリの作成 (mkdir コマンド)

mkdir コマンドを実行すると、ディレクトリを作成できます。引数として作成するディレクトリ名を指定します。

```
書式
mkdir ディレクトリ名

オプション
-p
指定されたディレクトリの上位ディレクトリも作成する
```

以下の例では、ホームディレクトリにディレクトリ dir1 を作成し、dir1 の下にディレクトリ dir2 を作成しますが、mkdir コマンドを 2 つに分けて実行しました。これは、ディレクトリを作るときはその上位層ができていないと作成できない、という制約があるためです。

ls コマンドに-R オプションをつけて実行すると、指定したディレクトリ以下のすべてを再帰的に表示することができます。

```
$ cd
$ pwd
/home/linuc
$ mkdir dir1
$ mkdir dir1/dir2
$ ls -R dir1
dir1:
dir2

dir1/dir2:
```

mkdir コマンドに-p オプションを付けて実行すると次のように一度に実行できます。

```
$ mkdir dir3/dir4
mkdir: ディレクトリ `dir3/dir4' を作成できません:
    そのようなファイルやディレクトリはありません
$ mkdir -p dir3/dir4
$ ls -R dir3
dir3:
dir4

dir3/dir4:
```

5.4.4 ディレクトリの削除 (rmdir コマンド)

ディレクトリを削除するには、rmdir コマンドを使用します。

```
書式
rmdir ディレクトリ名

オプション
-p
指定した階層までのディレクトリを一括で削除します。
```

rmdir は、ディレクトリの中にファイルやディレクトリが存在していると、ディレクトリは削除できません。

```
$ ls
dir1  ダウンロード  デスクトップ  ビデオ  画像
dir3  テンプレート  ドキュメント  音楽    公開
$ ls -R dir1
dir1:
dir2

dir1/dir2:
```

dir1 を削除してみます。

```
$ rmdir dir1
rmdir: 'dir1' を削除できません: ディレクトリは空ではありません
```

dir1/dir2 が存在しているので削除できません。

rmdir -p コマンドは、各ディレクトリにファイルが含まれない場合には一括でディレクトリを削除できます。

```
$ rmdir -p dir1/dir2
$ ls
dir3          テンプレート  ドキュメント  音楽  公開
ダウンロード  デスクトップ  ビデオ        画像
```

dir1 が dir2 と一緒に削除されました。

5.4.5 rm -r コマンドによるディレクトリとファイルの一括削除

ディレクトリ内にファイルが存在する場合、rmdir -p コマンドでもディレクトリは削除できません。ファイルも含めて一括でディレクトリを削除するには、rm -r コマンドを実行します。

touch コマンドを使うと、引数に指定したファイル名でサイズ 0 のファイルを作成できます。

```
$ touch dir3/dir4/test
$ ls -R dir3
dir3:
dir4

dir3/dir4:
test
```

rmdir -p コマンドを実行しますが、dir4 内に touch コマンドで作成したファイルがあるので削除できません。

```
$ rmdir -p dir3/dir4
rmdir: 'dir3/dir4' を削除できません: ディレクトリは空ではありません
```

rm -r コマンドで、dir3 ディレクトリ以下を一括で削除できます。

```
$ rm -r dir3
$ ls
テンプレート  ドキュメント  音楽  公開
ダウンロード  デスクトップ  ビデオ        画像
```

rm -r コマンドは指定したディレクトリ以下を一括削除してくれるので便利ですが、間違えてすべてを消してしまわないように注意が必要です。

5.5 特別なディレクトリ

Linux では、ディレクトリの中で特別なディレクトリや、それを表す記号があります。以下がその代表的なものです。

5.5.1 カレントディレクトリ (.)

現在いるディレクトリのことです。pwd コマンドで確認できます。

コマンド操作では「.」（ドット）を使って表わします。

5.5.2 親ディレクトリ (..)

1 階層上のディレクトリのことです。カレントディレクトリが/home/linuc の場合、/home が親ディレクトリにあたります。

コマンド操作では「..」（ドット 2 つ）を使って表わします。

5.5.3 ホームディレクトリ (~)

ユーザーの作業用となるディレクトリです。通常、/home ディレクトリ以下にユーザーごとに異なるホームディレクトリを用意します。

ログインしたユーザーは最初にホームディレクトリがカレントディレクトリになります。ユーザーは自分のファイルをホームディレクトリ以下に保存します。メールサーバーの場合、各ユーザー宛のメールが保存される場合もあります。

コマンド操作では「~」（チルダ）を使って表わします。

5.5.4 ルートディレクトリ (/)

ディレクトリ階層の、最上位階層を示します。

コマンド操作では「/」（ルート）を使って表わします。

5.5.5 パスの絶対指定と相対指定

ディレクトリの表記のことを「パス」（PATH) と呼ぶ場合があります。コマンド操作では対象となるファイルやディレクトリの指定にパスを記述します。

カレントディレクトリが/home/linuc であるとき、/usr/bin ディレクトリを指定するには次の 2 通りが存在します。

- /usr/bin
- ../../usr/bin

「/usr/bin」は、最上位のルートディレクトリ (/) からディレクトリ名を指定しています。これを絶対指定と呼びます。

「../../usr/bin」は、カレントディレクトリから階層をたどったディレクトリ名を指定しています。これを相対指定といいます。

絶対指定は、どのディレクトリがカレントディレクトリなのかに関係なくパスを指定できます。常に「/」（ルート）から指定するので、パスの記述が長くなることがあります。

相対指定は、カレントディレクトリが変わると表記が異なります。カレントディレクトリから見て近いディレクトリのときは指定の記述が短くて済みます。

使い分け方ですが、誰が見ても間違えないように記述するドキュメントなどでは絶対指定を使用し、ちょっとしたコマンドでの操作では絶対指定と相対指定を併用することが多いようです。

以下の例で、それぞれのディレクトリ指定についての動作を確認してみます。

ホームディレクトリをカレントディレクトリにします。

```
$ cd ~
$ pwd
/home/linuc
```

相対指定で親ディレクトリをカレントディレクトリにします。

```
$ cd ..
$ pwd
/home
```

絶対指定で/usr/bin をカレントディレクトリにします。

```
$ cd /usr/bin
$ pwd
/usr/bin
```

相対指定で/（ルート）をカレントディレクトリにします。

```
$ cd ../../
$ pwd
/
```

5.6 ファイルのコピー (cp コマンド)

cp(CoPy) コマンドを実行すると、ファイルをコピーすることができます。引数に指定するコピー元やコピー先、オプションによって様々な形でコピーが行えます。

```
書式
cp [オプション] コピー元 コピー先

オプション
-r
ディレクトリをコピーします。
ディレクトリの中にある全てのファイルおよびディレクトリが一括でコピーされます。
```

5.6.1 ファイルをディレクトリにコピー

コピー先にディレクトリを指定すると、ディレクトリ内にファイルのコピーが作成されます。ファイル名は、コピー元のファイル名のままコピーされます。

```
$ cd ~
$ mkdir work
$ cp /etc/hosts /home/linuc/work
$ ls -l /etc/hosts
-rw-r--r--. 1 root root 158  6月 23  2020 /etc/hosts
$ ls -l work
合計 4
-rw-r--r--. 1 linuc linuc 158  8月 11 12:47 hosts
```

同じサイズなので、/etc/hosts が hosts としてコピーされたことがわかります。

カレントディレクトリにコピーする場合には、コピー先に「.」を指定します。

```
$ cd work
$ pwd
/home/linuc/work
$ cp /etc/services .
$ ls -l /etc/services
-rw-r--r--. 1 root root 692252  6月 23  2020 /etc/services
$ ls -l
合計 684
-rw-r--r--. 1 linuc linuc    158  8月 11 12:47 hosts
-rw-r--r--. 1 linuc linuc 692252  8月 11 12:48 services
```

5.6.2 ファイルを別のファイル名としてコピー

コピー先に存在しないファイル名を指定すると、指定したファイル名でファイルのコピーが作成されます。

```
$ cp /etc/services cptest
$ ls -l
合計 1364
-rw-r--r--. 1 linuc linuc 692252  8月 11 12:49 cptest
-rw-r--r--. 1 linuc linuc    158  8月 11 12:47 hosts
-rw-r--r--. 1 linuc linuc 692252  8月 11 12:48 services
```

ファイルサイズから/etc/services が cptest としてコピーされたことがわかります。

5.6.3 ファイルを上書きでコピー

コピー先に存在するファイル名を指定すると、ファイルは上書きコピーされます。

```
$ cp /etc/hosts cptest
$ ls -l
合計 688
-rw-r--r--. 1 linuc linuc    158  8月 11 12:50 cptest
-rw-r--r--. 1 linuc linuc    158  8月 11 12:47 hosts
-rw-r--r--. 1 linuc linuc 692252  8月 11 12:48 services
```

ファイルサイズから/etc/hosts が cptest として上書きコピーされたことがわかります。

5.6.4 ディレクトリをコピー

cp コマンドに-r オプションを付けてコピー元にディレクトリを指定すると、ディレクトリ内のファイルおよびディレクトリを一括で再帰的にコピーできます。コピー先にはディレクトリを指定します。指定したコピー先ディレクトリが無い場合、指定した名前でディレクトリが作成されます。

コピー元の moto ディレクトリを作成し、その中に touch コマンドで cptest ファイルを作成します。

```
$ mkdir moto
$ touch moto/cptest
$ ls moto
cptest
```

cp -r コマンドでコピー元の moto ディレクトリを、存在しない saki ディレクトリとしてコピーします。

```
$ cp -r moto saki
$ ls saki
cptest
```

saki ディレクトリが作成され、その中に cptest ファイルがコピーされているのが分かります。

コピー先ディレクトリが存在している場合、そのディレクトリの中にコピー元のディレクトリが作成されます。

```
$ mkdir aru
$ cp -r moto aru
$ ls -R aru
aru:
moto

aru/moto:
cptest
```

コピー先のディレクトリの有無によって階層構造が変わることがあるので、注意してください。

5.7 ファイルの移動 (mv コマンド)

mv(MoVe) コマンドを実行すると、ファイルを移動できます。また、mv コマンドを使ってファイル名の変更も行えます。

```
書式
mv 移動元 移動先
```

5.7.1 ファイルを別のディレクトリに移動

移動元がファイルで、移動先がディレクトリの場合、移動元のファイルを移動先のディレクトリ内に移動します。

```
$ touch moto/mvtest
$ ls moto
cptest  mvtest
$ mv moto/mvtest saki
$ ls moto
cptest
$ ls saki
cptest  mvtest
```

moto ディレクトリから saki ディレクトリにファイル mvtest が移動したことがわかります。

5.7.2 ディレクトリを別のディレクトリに移動

移動元がディレクトリで、移動先がディレクトリの場合、移動元のディレクトリは移動先のディレクトリ内に移動します。

```
$ mv moto saki
$ ls saki
cptest  moto  mvtest
```

moto ディレクトリが saki ディレクトリ内に移動したことがわかります。

5.7.3 ファイル名の変更

移動元がファイルで、移動先が存在しないファイルの場合、移動元のファイル名が移動先に指定したファイル名に変更されます。

```
$ ls
aru  cptest  hosts  saki  services
$ mv cptest renametest
$ ls
aru  hosts  renametest  saki  services
```

ファイル cptest の名前がファイル renametest に変更されたことがわかります。

5.7.4 ディレクトリ名の変更

移動元がディレクトリで、移動先が存在しないディレクトリの場合、移動元のディレクトリが移動先に指定したディレクトリ名に変更されます。

```
$ mv saki renamedir
$ ls
aru  hosts  renamedir  renametest  services
```

saki ディレクトリの名前が renamedir ディレクトリに変更されたことがわかります。

5.8 ファイルの削除 (rm コマンド)

rm(ReMove) コマンドを実行すると、ファイルを削除できます。

```
書式
rm [オプション] ファイル名

オプション
-f
強制的に削除を実行します。
ファイルによっては削除に確認が求められますが、確認を行わず強制的に削除を
行います。

-i
対象ファイルを本当に削除してよいか、確認を求めます。誤って削除することを
防ぐために利用されます。

-r
ディレクトリを削除します。ディレクトリの中のファイルやディレクトリも削除します。
```

```
$ pwd
/home/linuc/work
$ ls
aru  hosts  renamedir  renametest  services
$ ls renamedir
cptest  moto  mvtest
```

renametest ファイルを削除してみます。

```
$ rm renametest
$ ls
aru  hosts  renamedir  services
```

renamedir ディレクトリを削除してみます。

```
$ rm renamedir
rm: 'renamedir' を削除できません: ディレクトリです
```

ディレクトリ内にファイルやディレクトリがあるので削除できません。

rm -r コマンドでディレクトリを一括削除します。

```
$ rm -r renamedir/
$ ls
aru  hosts  services
```

renamedir ディレクトリが一括で削除されました。

5.8.1 削除したファイルは復元できない

Linux では、一度削除したファイルを復活させることはできません。-f オプションで確認無しで削除する場合や、-r オプションでディレクトリごと一括で削除する場合には注意が必要です。

5.9 ファイルの検索（find コマンド）

ファイルがどこのディレクトリに存在するか、find コマンドにて検索できます。

```
書式
find ディレクトリ -name ファイル名
```

/etc 配下にある hosts という名前のファイルを検索してみましょう。セキュリティなどの関係で参照が許されていないディレクトリ内は検索が行えませんため、エラーが出ます。

```
$ find /etc/ -name hosts
find: ‘/etc/lvm/devices’: 許可がありません
（略）
/etc/hosts
（略）
/etc/avahi/hosts
（略）
```

hosts という名前のファイルがどこにあるか表示されました。

5.10 コマンドのパスを表示（which コマンド）

コマンドの実体はプログラムです。プログラムもファイルの一種であり、/bin や/sbin といったプログラム用のディレクトリに配置されています。

基本的なコマンドを実行するとき、その実体であるプログラムがどのディレクトリに配置されているかを意識する必要はありません。これは、PATH という環境変数にプログラムが配置されているディレクトリが設定されているからです。

which コマンドを実行すると、引数に指定したコマンドが PATH 環境変数に含まれるディレクトリのうち、どのディレクトリにあるものかを表示できます。

```
書式
which コマンド名
```

以下の実行例では、cat コマンドがどのディレクトリにあるのかを調べています。

```
$ which cat
/usr/bin/cat
```

cat コマンドを実行すると、/usr/bin/cat が呼び出されているのがわかります。

5.10.1 環境変数 PATH の設定確認

環境変数 PATH に設定された値を表示するには、echo コマンドを使用します。環境変数名の前に$を付けることで、その環境変数に設定された値を参照できます。

```
$ echo $PATH
/home/linuc/.local/bin:/home/linuc/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/u
sr/sbin
```

5.11 コマンドのヘルプを表示する

コマンドにはヘルプが含まれていることがあります。実行したいコマンドに対して- help オプションをつけて実行することで、コマンドのヘルプを表示できます。

```
書式
コマンド --help
```

以下の実行例では、ls コマンドのヘルプを表示しています。

```
$ ls --help
使用法: ls [オプション]... [ファイル]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
（略）
```

5.12 マニュアルの使い方

Linux には、便利なオンラインマニュアルが含まれています。ここではその使い方を紹介します。

```
書式
man [セクション] コマンド名
```

以下の実行例では、man コマンドで ls コマンドのマニュアルを参照しています。

```
$ man ls
LS(1)                            User Commands                            LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort  is  speci‐
       fied.
（略）
```

主な見出しの意味は、以下の通りです。

- SYNOPSIS（書式）コマンドの書式です。
- DESCRIPTION（説明）コマンドなどの説明です。オプションの解説が含まれる場合があります。
- OPTIONS コマンドのオプションです。
- SEE ALSO（関連事項）このプログラムに関連しているコマンドなどが記載されています。ここに列挙している内容を man コマンドで調べることができます。

5.12.1 マニュアルのセクション

ls のマニュアルの先頭に「LS(1) 」という記述があります。これは ls のマニュアルがセクション 1 に分類されている、という意味です。マニュアルのセクションとは、マニュアルを種類毎に分類したものです。セクションには番号がついており、以下の表のようになっています。

項目	内容
1	ユーザコマンド
2	システムコール
3	システムライブラリや関数
4	デバイスやデバイスドライバ
5	ファイルの形式
6	ゲームやデモなど
7	その他
8	システム管理系のコマンド
9	カーネルなどの情報

5.12.2 セクションを指定したマニュアルの表示

同じ名称でも、複数のマニュアルに分かれていることがあります。passwd コマンドはセクション 1 のユーザコマンドである他、ユーザーのパスワードを保存した設定ファイルでもあるのでセクション 5 にもマニュアルがあります。

```
$ man passwd
PASSWD(1)                   General Commands Manual                  PASSWD(1)

名前
       passwd - ユーザパスワードを変更する

書式
       passwd  [-k]  [-l]  [-u  [-f]]  [-d] [-e] [-n mindays] [-x maxdays] [-w
       warndays] [-i inactivedays] [-S] [--stdin] [username]
（略）
```

セクション 5 の passwd のエントリを見るには、セクション番号をオプションとして指定します。

```
$ man 5 passwd
passwd(5)                     File Formats Manual                     passwd(5)

NAME
       passwd - password file

DESCRIPTION
       The  /etc/passwd file is a text file that describes user login accounts
       for the system.  It should have read permission allowed for  all  users
       (many  utilities,  like ls(1) use it to map user IDs to usernames), but
       write access only for the superuser.
（略）
```

passwd ファイルの書式について説明したマニュアルが表示されました。

6 標準入出力とフィルタコマンド

Linux のコマンドの多くは、「処理を行うデータの入力」と「処理を行った結果の出力」を持っています。これを標準入出力と呼んでいます。いくつかのコマンドの標準入出力を組み合わせてデータを処理することで、必要な結果を得ることができます。このような処理が行えるコマンドはフィルタコマンドと呼ばれています。

この章では、標準入出力とフィルタコマンドの使い方について学びます。

本章の内容

- 標準入出力
- リダイレクト
- 標準エラー出力
- パイプ
- データの先頭や末尾の表示 (head コマンド・tail コマンド)
- テキストファイルのソート (sort コマンド)
- 行の重複の消去 (uniq コマンド)
- 文字をカウントする（wc コマンド）
- 文字列を検索する（grep コマンド）

6.1 標準入出力

Linux の多くのコマンドは、「1 つの入力」と「2 つの出力」があります。それぞれ標準入力・標準出力・標準エラー出力と呼びます。

- 標準入力
  標準入力はコマンドに入ってくるデータのことです。標準入力のデフォルトはキーボードになっています。
- 標準出力
  標準出力はコマンドの実行結果を書き出す先のことです。標準出力のデフォルトはコンソールになっています。
- 標準エラー出力
  標準エラー出力はエラーメッセージを書き出す先のことです。標準エラー出力のデフォルトはコンソールになっています。

たとえば ls コマンドを実行した場合、カレントディレクトリのファイルとディクトリの一覧がコンソールに表示されます。

図 37: 標準入出力

6.2 リダイレクト

標準出力は通常コンソールに表示されますが、リダイレクトを使うと標準出力をファイルに書き込めます。

6.2.1 標準出力のリダイレクト

コマンドの実行結果として標準出力に出力される内容をファイルにリダイレクトします。リダイレクトは「>」を使って表します。

```
書式
コマンド > 出力先ファイル
```

ls コマンドの結果をリダイレクトしてファイルに書き込んでみます。作業用に work ディレクトリを作成して、作業を行います。

```
$ cd
$ mkdir work
$ cd work
$ touch test
```

リダイレクトを行わずに ls コマンドを実行してみます。

```
$ ls
test
```

結果はコンソールに表示されます。

標準出力を ls-output ファイルにリダイレクトしてみます。

```
$ ls > ls-output
$ ls
ls-output  test
$ cat ls-output
ls-output
test
```

標準出力がファイルにリダイレクトされたので、コンソールには何も表示されず、作成されたファイル内に ls コマンドの実行結果が書き込まれていました。ls-output が先に作成されてから ls コマンドが実行されるため、ls-output 自身も ls コマンドの結果に含まれています。

6.2.2 標準出力の追加リダイレクト

すでにリダイレクト先のファイルが存在している場合、リダイレクト結果は上書きされます。上書きせず、以前の結果を残したまま追記したい場合は、アペンド (») の記号を用います。

```
書式
コマンド >> 出力先ファイル
```

まず、通常のリダイレクトで ls コマンドを実行してみます。

```
$ ls > ls-output
$ cat ls-output
ls-output
test
```

前回書き込んだ結果は上書きされ、コマンド実行 1 回分の結果のみが書き込まれています。

次に「»」でリダイレクトを行ってみます。

```
$ ls >> ls-output
$ cat ls-output
ls-output
test
ls-output
test
```

2 回目のリダイレクトは追加リダイレクトのため、1 回目の結果を残したまま 2 回目の実行結果を追記しています。

6.2.3 cat コマンドによるファイル作成

cat コマンドは引数を指定しないと、標準入力で受け取ったデータをそのまま標準出力に出力します。デフォルトの標準入力はキーボードからの入力です。cat コマンドの標準出力をファイルにリダイレクトすることで、キーボードから入力した文字列をファイルに書き込めます。

```
$ cat > cat-output
Hello!
This is cat redirect.
（行頭でCtrl+dを押す）
$ cat cat-output
Hello!
This is cat redirect.
```

「Ctrl+d」は EOF(End Of File) を入力するキーで、データ入力の終わりを意味します。cat コマンドの標準入力は EOF を受け取ったことで入力が終わったと判断し、cat コマンドの実行を終了します。

6.3 標準エラー出力

コマンドを実行してエラーが発生すると、エラーメッセージはコンソールに出力されます。エラーの出力は、標準出力ではなく、標準エラー出力へと出力されています。どちらの出力もデフォルトの出力先がコンソールになっているため区別が付きませんが、リダイレクトすることで別々に扱えます。

標準出力には番号がついており、標準出力が 1、標準エラー出力が 2 になっています。標準エラー出力をリダイレクトするには「2>」と指定します。

```
書式
コマンド 2> 出力先ファイル
```

まず、エラーを発生させてみます。

```
$ ls nodir
ls: 'nodir' にアクセスできません: そのようなファイルやディレクトリはありません
```

存在しないディレクトリに対して ls コマンドを実行するとエラーが発生して、エラーメッセージが表示されます。このエラーメッセージをファイルにリダイレクトします。

```
$ ls nodir 2> ls-error
$ cat ls-error
ls: 'nodir' にアクセスできません: そのようなファイルやディレクトリはありません
```

コマンド実行時のメッセージはリダイレクトされたのでコンソールには表示されず、ファイルに書き込まれているのがわかります。

6.3.1 標準出力と標準エラー出力を別々にリダイレクトする

リダイレクトすることで、標準出力と標準エラー出力は別々に扱われます。それぞれを別々のファイルにリダイレクトすることもできます。

```
書式
コマンド > 標準出力先ファイル 2> 標準エラー出力先ファイル
```

以下の例では、/etc ディレクトリ内を ls -R コマンドで表示しています。いくつかの下位ディレクトリは参照できずエラーになります。正常な実行結果は ls-etc ファイル、エラーメッセージは ls-etc-error ファイルにリダイレクトされます。

```
$ ls -R /etc > ls-etc 2> ls-etc-error
$ cat ls-etc
/etc:
DIR_COLORS
（略）
almalinux-saphana.repo
$ cat ls-etc-error
ls: ディレクトリ '/etc/audit' を開くことが出来ません: 許可がありません
（略）
ls: ディレクトリ '/etc/sudoers.d' を開くことが出来ません: 許可がありません
```

ls-etc には正常に実行された結果、ls-etc-error にはエラーメッセージがリダイレクトされています。

6.3.2　標準出力と標準エラー出力をまとめてリダイレクトする

標準出力と標準エラー出力をまとめてリダイレクトして 1 つのファイルに書き込みたい場合には「2>&1」と指定します。標準エラー出力（2>）を標準出力 (1) に混ぜる（&）というような意味合いになります。

```
書式
コマンド > 標準出力先ファイル 2>&1
```

前に実行したコマンドの結果を 1 つの ls-etc-mix ファイルにリダイレクトします。

```
$ ls -R /etc > ls-etc-mix 2>&1
$ less ls-etc-mix
```

ls-etc-mix には、正常に実行された結果とエラーメッセージがまとめてリダイレクトされています。エラーが発生する都度リダイレクトされているので、エラーメッセージはいろいろなところに書き込まれていることを確認してください。確認が終了したら「q」を入力して less コマンドを終了します。

6.3.3　標準入力にリダイレクトする

リダイレクトは標準入力に対しても行えます。

```
書式
コマンド < ファイル
```

たとえば、以下の例では ls-etc-mix ファイルを less コマンドの標準入力にリダイレクトしています。

```
$ less < ls-etc-mix
```

ただし、ほとんどのコマンドが引数にファイルを指定して読み込めるので、標準入力のリダイレクトを使うことは希です。

6.4　パイプ

パイプを使うと、標準出力をその他のコマンドの標準入力に渡すことができます。パイプを使うことで、複数のコマンドを組み合わせて処理が行えます。パイプからのデータを受け取って処理するコマンドを「フィルタ」と呼びます。

```
書式
コマンド1 | コマンド2
```

パイプは「|」で指定します。一般的な日本語キーボードでは Shift キーを押しながら円マークキー（BackSpace キーの左）を入力します。

書式では、コマンド 1 の標準出力をコマンド 2 の標準入力に渡しています。このようなパイプによるデータの流れをテキストストリームとも呼びます。

図 38: パイプ

6.4.1 less コマンドによるページング

less コマンドは標準入力のデータをページ表示できます。コマンド実行時の結果表示が長すぎる場合などに使います。

```
$ cat /etc/services | less
# /etc/services:
# $Id: services,v 1.49 2017/08/18 12:43:23 ovasik Exp $
#
# Network services, Internet style
# IANA services version: last updated 2016-07-08
（略）
```

コマンドを実行している端末ソフトウェアがバックスクロールをサポートしているような場合には遡って見ることもできますが、遡れる行数が限られている場合には有効な方法です。

標準出力と標準エラー出力を 1 つのファイルに出力する場合の方法も利用できます。「2>&1」を指定しないで実行すると、エラーメッセージがまとめて表示されますが、スクロールすると消えてしまいます。

以下の 2 つのコマンドを実行して、結果を比較してみてください。

```
$ ls -R /etc | less
$ ls -R /etc 2>&1 | less
```

6.5 データの先頭や末尾の表示 (head コマンド・tail コマンド)

head コマンドや tail コマンドを実行すると、データの先頭や末尾など一部分のみを表示できます。

6.5.1 head コマンドによる先頭の表示

head コマンドはデータの先頭を表示します。オプションを指定しない場合は、先頭から 10 行を標準出力します。

```
書式
head [オプション] [ファイル]

オプション
-n 行
先頭から指定した行を標準出力します。-行数と指定することもできます。
```

以下の例では、cat -n コマンドの標準出力をパイプで受け取って先頭 5 行を表示しています。

```
$ cat -n /etc/services | head -n 5
     1  # /etc/services:
     2  # $Id: services,v 1.49 2017/08/18 12:43:23 ovasik Exp $
     3  #
     4  # Network services, Internet style
     5  # IANA services version: last updated 2016-07-08
```

6.5.2 tail コマンドによる末尾の表示

tail コマンドはデータの終わり部分を標準出力します。オプションを指定しない場合は、末尾から 10 行を標準出力します。

```
書式
tail [オプション] [ファイル]

オプション
-n 行
末尾から指定した行を標準出力します。-行数と指定することもできます。
```

以下の例では、cat -n コマンドの標準出力をパイプで受け取って末尾 5 行を表示しています。-行数というオプションの指定方法を使っています。

```
$ cat -n /etc/services | tail -5
 11469  axio-disc       35100/udp               # Axiomatic discovery protocol
 11470  pmwebapi        44323/tcp               # Performance Co-Pilot client
     HTTP API
 11471  cloudcheck-ping 45514/udp               # ASSIA CloudCheck WiFi
     Management keepalive
 11472  cloudcheck      45514/tcp               # ASSIA CloudCheck WiFi
     Management System
 11473  spremotetablet  46998/tcp               # Capture handwritten signatures
```

6.6 テキストファイルのソート (sort コマンド)

sort コマンドはデータをソートします。オプションでどのような順序でソートするか指定することができます。

```
書式
sort [オプション] [ファイル]

オプション
-k n
n列目のデータをソートします。

-t 文字列
列の区切りとして文字列を使用します。デフォルトは空白文字が使用されます。

-n
数値としてソートします。

-r
逆順でソートします。
```

各オプションの動作を確認してみましょう。

6.6.1 ソート用データの確認

Linux のユーザー情報が記述されているファイルである/etc/passwd をデータとして使用して、sort コマンドの動作を確認してみます。

tail コマンドで末尾 5 行だけ表示してみます。

```
$ tail -5 /etc/passwd
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/usr/sbin/nologin
chrony:x:982:981:chrony system user:/var/lib/chrony:/sbin/nologin
dnsmasq:x:981:980:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
linuc:x:1000:1000:LinuC:/home/linuc:/bin/bash
```

先頭がユーザー名、3 番目がユーザー ID 番号です。これらのデータを使ってどのようにソートが変わるのか確認します。

6.6.2 単純なソート

ユーザー名でソートしてみます。

```
$ tail -5 /etc/passwd | sort
chrony:x:982:981:chrony system user:/var/lib/chrony:/sbin/nologin
dnsmasq:x:981:980:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
linuc:x:1000:1000:LinuC:/home/linuc:/bin/bash
```

```
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/usr/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
```

ユーザー名がアルファベット順でソートされました。

6.6.3 逆順のソート

逆順でソートしてみます。

```
$ tail -5 /etc/passwd | sort -r
tcpdump:x:72:72::/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/usr/sbin/nologin
linuc:x:1000:1000:LinuC:/home/linuc:/bin/bash
dnsmasq:x:981:980:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
chrony:x:982:981:chrony system user:/var/lib/chrony:/sbin/nologin
```

ユーザー名がアルファベット逆順でソートされました。

6.6.4 列を指定したソート

3 番目のユーザー ID 番号でソートしてみます。区切り文字を「:」に指定します。

```
$ tail -5 /etc/passwd | sort -k 3 -t :
linuc:x:1000:1000:LinuC:/home/linuc:/bin/bash
tcpdump:x:72:72::/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/usr/sbin/nologin
dnsmasq:x:981:980:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
chrony:x:982:981:chrony system user:/var/lib/chrony:/sbin/nologin
```

ユーザー ID 番号でのソートになりましたが、数値ではなく文字列としてソートされてしまったので、1000 が一番最初に来てしまいました。

6.6.5 数値としてのソート

ユーザー ID 番号を数値としてソートするよう-n オプションを追加します。

```
$ tail -5 /etc/passwd | sort -k 3 -t : -n
tcpdump:x:72:72::/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/usr/sbin/nologin
dnsmasq:x:981:980:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
chrony:x:982:981:chrony system user:/var/lib/chrony:/sbin/nologin
linuc:x:1000:1000:LinuC:/home/linuc:/bin/bash
```

ユーザー ID 番号を数値の小さい順番にソートしました。

6.7 行の重複の消去 (uniq コマンド)

uniq コマンドを使うことで直前の行と同じ内容があった場合、対象行を出力しません。連続している同じ内容の行を、1 行にまとめることができます。

```
書式
uniq [ファイル]
```

6.7.1 重複の消去用データの作成

以下のような内容のファイルを作成します。

```
$ cat > uniq-test
A
B
A
C
C
D
（行頭でCtrl+dを入力）
```

```
$ cat uniq-test
A
B
A
C
C
D
```

6.7.2 重複を消去する

uniq コマンドを実行します。

```
$ cat uniq-test | uniq
A
B
A
C
D
```

前後の重複していた文字列 C が 1 行にまとめられましたが、A は間に B が入っているので重複とは見なされず残ったままです。

uniq コマンドに渡す前に、sort コマンドでソートしてみます。

```
$ cat uniq-test | sort | uniq
A
B
C
D
```

今度は A も重複が消去されました。

6.8 文字をカウントする（wc コマンド）

wc コマンドは、データの文字をカウントします。

```
書式
wc [オプション] [ファイル]

オプション
-c
文字数をカウントする

-l
行をカウントする

-w
単語をカウントする
```

各オプションの動作を確認してみましょう。

6.8.1 文字をカウントする

wc コマンドで、文字をカウントします。オプションを指定しないと、行数、単語数、文字数がまとめて表示されます。

```
$ cat /etc/services | wc
  11473   63129  692252
```

オプションを指定することで、指定した種類でのカウントのみを行えます。

```
$ cat /etc/services | wc -l
11473
```

```
$ cat /etc/services | wc -w
63129
```

```
$ cat /etc/services | wc -c
692252
```

6.9 文字列を検索する（grep コマンド）

grep コマンドは、指定された条件でデータ内の文字列を検索します。

```
書式
grep [オプション] 検索条件 [ファイル]
```

検索条件はただの文字列のほか、正規表現で指定します。

6.9.1 正規表現

grep コマンドでは、検索条件として正規表現が用いられます。正規表現は多くのプログラミング言語でも利用されているパターンの表現方法です。

正規表現で使われる記号と意味

記号	意味
^	行頭を表す
$	行末を表す
.	任意の一字を意味する
*	直前文字の 0 回以上の繰り返しを意味する
[…]	…の中の任意の一字を意味する
[^...]	…の文字が含まれないことを意味する
\	正規表現の記号をエスケープする

正規表現の利用例

記号	意味
^a	a で始まっている行
b$	b で終わっている行
a.b	a と b の間に 1 文字入っている
[ab]ab	a もしくは b に続く ab(aab,bab)
[^ab]ab	a もしくは b で始まらない (not) で、ab が続くもの (例: xab, zab 等)

6.9.2　単純な文字列検索

grep コマンドの検索条件指定に正規表現を使わなければ、単純な文字列検索が行えます。

```
$ cat /etc/services | grep http
#       http://www.iana.org/assignments/port-numbers
http            80/tcp          www www-http    # WorldWideWeb HTTP
http            80/udp          www www-http    # HyperText Transfer Protocol
http            80/sctp                         # HyperText Transfer Protocol
https           443/tcp                         # http protocol over TLS/SSL
https           443/udp                         # http protocol over TLS/SSL
https           443/sctp                        # http protocol over TLS/SSL
gss-http        488/tcp
gss-http        488/udp
 （略）
```

http という文字列が含まれている行を検索できました。

6.9.3　先頭文字列を指定した検索

先頭の文字列を表す「^」を指定して検索を行います。

```
cat /etc/services | grep ^http
http            80/tcp          www www-http    # WorldWideWeb HTTP
http            80/udp          www www-http    # HyperText Transfer Protocol
http            80/sctp                         # HyperText Transfer Protocol
https           443/tcp                         # http protocol over TLS/SSL
https           443/udp                         # http protocol over TLS/SSL
https           443/sctp                        # http protocol over TLS/SSL
http-mgmt       280/tcp                 # http-mgmt
http-mgmt       280/udp                 # http-mgmt
 （略）
```

http から始まる行だけに絞り込まれました。

6.9.4　行末文字列を指定した検索

行末の文字列を表す「$」を指定して検索を行います。

```
$ cat /etc/services | grep http$
md-cg-http      2688/tcp                # md-cf-http
md-cg-http      2688/udp                # md-cf-http
webemshttp      2851/tcp                # webemshttp
webemshttp      2851/udp                # webemshttp
plysrv-http     6770/tcp                # PolyServe http
plysrv-http     6770/udp                # PolyServe http
manyone-http    8910/tcp                # manyone-http
manyone-http    8910/udp                # manyone-http
```

http で終わる行だけに絞り込まれました。

7 vi エディタ

Linux の設定をするためにしばしば利用される vi(vim) を操作しながら、エディタの基本的な機能を学んでみましょう。

本章では、vi の機能のうち、ファイルの開閉やテキストの検索や置換、ページ移動や編集方法など基礎的な操作を解説します。

本章の内容

- vi はモーダル型エディタ
- vi の基本的な使用方法
- 入力モードへの切り替え
- 編集モードでのカーソルの移動
- 入力モードでのカーソル移動
- 様々な編集操作
- 検索

7.1 vi はモーダル型エディタ

vi はモーダル型のエディタです。モーダルとは「モードがある」という意味です。Windows などで使われている、モードがないモードレス型のエディタと操作方法が異なります。

vi を起動すると、最初は編集モードになっています。行や文字の削除、カット、コピー、ペーストなどの編集操作が行えるモードになっています。

入力モードに切り替えると、モードレス型のエディタのように文字の入力や削除が行えます。

7.1.1 vi と vim の関係

vi は、様々な UNIX にインストールされているため、どのような環境においても設定ファイルなどの編集が行えるように使い方を覚えておくべき基本的なエディタです。

現在ではオリジナルの vi ではなく、新たにクローンとして開発された vim（Vi IMproved・改善された vi）が使用されています。vim では、より使いやすくなるよう様々な機能が強化改善されています。

7.2 vi の基本的な使用方法

vi はモーダル型エディタのため、入力モードや編集モードへの切り替え、ファイルの保存や終了方法がモードレス型エディタと異なります。

vi の起動からファイルの保存、vi の終了までの基本的な流れを確認しましょう。

7.2.1 vi を起動する

vi でファイルを開くには vi コマンドの後に編集したいファイルを指定します。

```
書式
vi [ファイル]
```

指定したファイルが存在しない場合、保存時に新しいファイルが作成されます。ファイルを指定しなかった場合、保存時にファイル名を指定する必要があります。

新しく test.txt を作成するように指定して、vi を起動します。

```
$ vi test.txt
```

図 39: vi 実行画面

7.2.2 入力モードに切り替える

vi を起動した直後は編集モードになっています。入力モードに切り替えて、文字を入力してみます。

一番簡単な入力モードへの切り替えは、「i」（Insert）を入力します。画面左下に「-挿入 -」と表示されて、入力モードに切り替わります。

何か適当な文字を入力してみます。BackSpace キーや Delete キーで文字を消すこともできます。

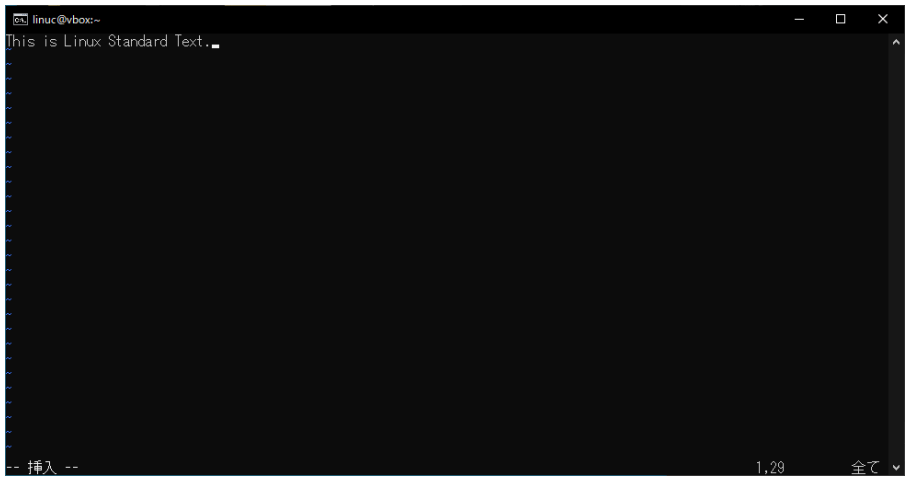

図 40: vi 入力モード画面

vim では、入力モードの状態でもカーソルキーでカーソルを移動できます。

7.2.3 編集モードに切り替える

入力モードから編集モードに切り替えるには、ESC キーを入力します。

画面左下の「- 挿入 -」の表示が消えて、編集モードに切り替わります。

vi を使っていると、現在の状態がわからなくなってしまうことがあります。そのような時には、ひとまず ESC キーを入力して編集モードに切り替えるとよいでしょう。

7.2.4 ファイルを保存する

ファイルを保存するには、編集モードで保存のコマンドを実行します。

コマンドを実行するには「:」を入力します。画面左下に「:」が表示されて、コマンドを受け付ける状態になります。ファイルを保存するには「w」（Write）を入力します。

新しく作成したファイルの場合、画面左下に「“test.txt” [新] xL, xB 書込み」と表示されて、新しくファイルが作成されたことが分かります。

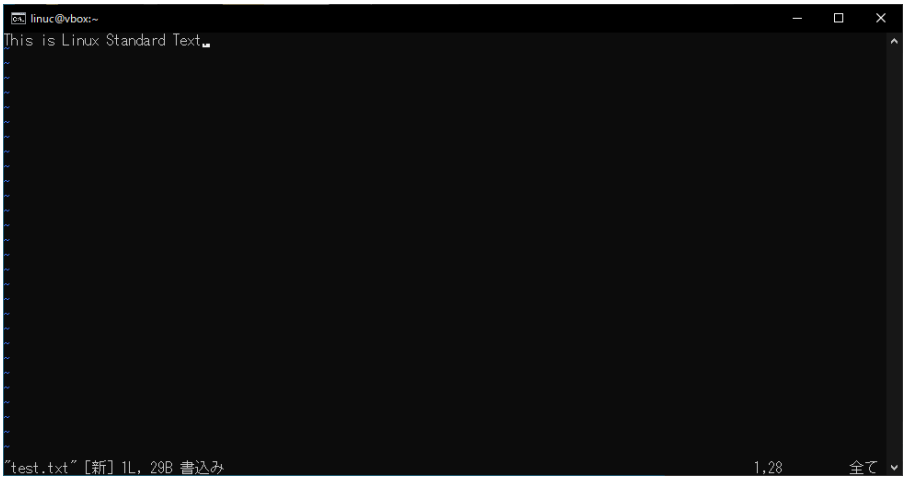

図 41: vi ファイル書き込み画面

7.2.5 vi を終了する

vi を終了するには、編集モードの状態で終了のコマンドを入力します。

終了は「:q」（Quit）とコマンドを入力します。正常に終了すると、コマンドラインに戻ります。

7.2.6 ファイルが未保存の状態から終了する

編集中のファイルが未保存の場合、警告が表示されて終了できません。未保存の状態から終了するには、次の 2 つの方法のいずれかを行う必要があります。

- ファイルの保存と終了を同時に行うファイルの保存と vi の終了を同時に行うには「:wq」と入力します。
- ファイルを保存せずに終了する間違えて修正してしまった場合など、保存せずに終了したい場合には「:q!」と入力します。編集内容は破棄されて、コマンドラインに戻ります。

7.2.7 ZZ で保存終了する

:wq と似た動作をするものとして ZZ があります。

:wq は変更が無い場合でもファイルを保存して終了するので、タイムスタンプが:wq 実行時点のものに変更されます。

一方、ZZ は変更がある場合には保存し、変更が無い場合には保存せず vi を終了するのでタイムスタンプが変更されません。

プログラムのソースコードファイルの場合、タイムスタンプで変更の有無を判断するので、:wq だと不都合が起きる場合がありますので、:wq と ZZ を使い分けるようにするとよいでしょう。

7.3 入力モードへの切り替え

編集モードから入力モードへの切り替えは、「i」以外にもいくつかあります。それぞれ入力モードに入った際のカーソル位置などが異なるので、状況に応じて使い分けるとよいでしょう。

以下の解説は、それぞれ自身で試してみてください。

7.3.1 行の先頭に入力

「i」はカーソルの位置から入力になりますが、「I」（Shift+i）を入力すると、カーソルがどの位置にあってもカーソルが行頭まで移動して入力モードになります。

7.3.2 カーソルの後ろ、行末から入力

「i」はカーソルの位置から入力になりますが、行末にカーソルがある時に末尾以降への文字入力が行えません。正確には、入力モードに切り替えた後、右カーソルで 1 文字分カーソルを移動させる必要があります。

「a」（Append）を入力すると、カーソルが 1 文字分右に移動した後に入力モードになります。また「A」（Shift+a）を入力すると、カーソルがどの位置にあってもカーソルが行末まで移動して入力モードになります。

7.3.3 カーソルの前後の行から入力

カーソルの前の行に入力したり、後ろの行に入力できます。

「o」を入力すると、カーソルの後ろの行に空行が挿入されて入力モードになります。また「O」（Shift+o）を入力すると、カーソル位置に空行が挿入されて入力モードになります。

7.4 編集モードでのカーソルの移動

入力モードでカーソルキーを使ってカーソルを移動させることもできますが、本来の vi は編集モードでカーソルを移動させます。また、各種編集作業を行う際にもカーソル移動が必要となるので、編集モードでのカーソル移動について解説します。

7.4.1 1 文字単位の移動

編集モードでのカーソル移動は、vim ではカーソルキーでも行えますが、hjkl キーでも行えます。

カーソルの移動

コマンド	カーソルの動き
h	←左
j	↓下
k	↑上
l	→右

キーボードの並びで h が左端、l が右端なので、そのまま左右になります。

7.4.2 行頭、行末への移動

「0」（ゼロ）を入力すると、現在の行の行頭に移動します。

「$」を入力すると、現在のカレント行の行末に移動します。

7.4.3 先頭行、最終行への移動

「gg」を入力すると文書の先頭、1 行目に移動します。

「G」（Shift+g）を入力すると文書の最後尾、最終行に移動します。

7.5 入力モードでのカーソル移動

入力モードでも、カーソルキー以外でカーソルを移動させることができます。

7.5.1 ページ単位でのカーソル移動

「Ctrl+f」（Forward）で次のページに移動します。

「Ctrl+b」（Backward）で前のページに移動します。

7.5.2 行番号を指定した移動

「:」の後に数値を入力すると、指定された番号の行に移動します。

7.6 様々な編集操作

編集モードでは文字の削除の他、「カット＆ペースト」や「コピー＆ペースト」、アンドゥなどが行えます。

編集モードで使える基本的なコマンドです。

カット・アンド・ペーストで使うコマンド

コマンド	内容
x	1 文字削除（バッファに入る）
dd	1 行削除（バッファに入る）
yy	1 行コピー（ヤンク）
p	カーソルの文字の次、または後ろの行にペースト
P	カーソルの文字の前、または前の行にペースト
r	カーソル位置の文字を書き換える
cw	カーソル位置からの単語を書き換える
u	編集作業を取り消し（アンドゥ）

7.6.1 文字のカット＆ペースト

「x」で文字を削除すると、その文字はバッファに格納されます。

「p」を入力すると、バッファに格納されている文字がカーソルの後ろにペーストされます。「P」（Shift+p）を入力すると、カーソルの位置にペーストされます。

7.6.2 行のカット＆ペースト

「dd」で行を削除すると、その行はバッファに格納されます。

「p」を入力すると、バッファに格納されている行がカーソルの後ろの行にペーストされます。「P」（Shift+p）を入力すると、カーソルの前の行にペーストされます。

7.6.3 行のコピー＆ペースト

「yy」で行をバッファに格納できます。これを「ヤンク」（Yank・引っ張るの意）と呼びます。

「p」を入力すると、バッファに格納されている行がカーソルの後ろの行にペーストされます。「P」（Shift+p）を入力すると、カーソルの前の行にペーストされます。

7.6.4 1 文字の書き換え

「r」を入力すると、カーソル位置の 1 文字を書き換えれます。表示は変わりませんが、書き換えたい文字を入力すると文字が書き換わります。

7.6.5 単語の書き換え

「cw」を入力すると、単語の書き換えができます。カーソル位置から単語の終わりまでが消去され、文字入力モードに切り替わります。書き換えたい文字を入力して、必要に応じて ESC キーを押して編集モードに戻ります。

単語の終わりがどこまでなのかは、英語等の表記の場合は空白で判定されます。日本語の場合には単語終わりの判定が難しく 1 文字だけ、あるいは漢字だけなどの書き換えになってしまいます。

7.6.6 アンドゥによる編集作業の取り消し

カット＆ペーストなどの編集作業は、vim ではアンドゥで取り消すことができます。

「u」（Undo）を入力すると、直前の編集作業が取り消されます。入力し続けると、一番最初の無編集の状態まで戻すことができます。

7.7 検索

編集モードで/コマンドを使うと、目的の文字列を検索してカーソルを移動します。

検索に関するコマンド

コマンド	内容
/検索文字列	文字列の検索
n	下方向へ再検索
N	上方向へ再検索

7.7.1 文字列の検索

編集モードで「/」を入力すると、画面左下に/が表示されます。検索したい文字列を続けて入力して Enter キーを入力すると、文字列が検索されてカーソルが移動します。

同じ文字列が複数ある場合、「n」を入力すると次の文字列が検索されてカーソルが移動します。「N」（Shift+n）を入力すると前の文字列が検索されてカーソルが移動します。

最終行まで到達すると、「下まで検索したので上に戻ります」と表示されて、最初に検索された文字列にカーソルが移動します。

文字列を検索すると検索した文字列がハイライト表示されます。このハイライト表示は次の文字列を検索するまでハイライト表示が残ります。ハイライトを消すには:nohlsearch コマンドを実行してください。

8 ユーザーとグループの管理

ユーザーとグループは、Linux のシステムを制御するための基本的な単位として機能します。

本章では、ユーザーとグループの管理について解説します。ユーザーの作成や削除、ユーザーパスワードの設定、グループを作成してユーザーを割り当てたりグループ設定の変更やグループの削除について理解を深めましょう。

本章の内容

- ユーザーとグループ
- システムを管理するユーザー root
- ユーザーの管理
- グループの管理
- パスワードファイル/etc/passwd とシャドウファイル/etc/shadow

8.1 ユーザーとグループ

ユーザーは、Linux を利用するための基本的な単位です。たとえば Linux を操作するには、ユーザーでログインします。メールサーバーであれば、各ユーザー宛のメールを受け取ることができます。

グループは、複数のユーザーをまとめて管理する仕組みです。ある特定のグループに所属しているユーザーだけがシステムを管理できる、というような使い方をします。ファイルサーバーであれば、特定のグループだけで読み書きできるファイル共有を作成できます。

ユーザーとグループを適切に設定することで、ファイルやディレクトリ、任意のプログラムなどを必要なユーザーにのみ参照・編集する権限や実行する権限を与えることができます。

8.2 システムを管理するユーザー root

Linux のシステム全体を管理するユーザーは root です。root はすべてのファイルに対するアクセスが許可されていたり、システム管理用のコマンドが実行できるなど、一般ユーザーに比べて強力な権限を持っているので、「特権ユーザー」や「スーパーユーザー」などと呼ばれます。

root として操作が行えると、システムを乗っ取ったり破壊したりできるので、セキュリティの観点から root は厳格に管理する必要があります。

root で作業をするには、su コマンドでユーザーを切り替えるか、sudo コマンドで root の権限でコマンドを実行します。実務でシステム管理を行う際には、sudo コマンドを使う方法が一般的です。

8.2.1 su コマンド

su（Substitute User）コマンドは、一時的に他のユーザーに切り替えるためのコマンドです。

su コマンドの引数として、切り替え先のユーザーを指定します。省略すると root を指定したことになります。

切り替え先ユーザーのパスワードを入力する必要があります。パスワードが設定されていないユーザーには切り替えられません。

オプションを省略すると、シェルなどの環境をそのまま引き継いでユーザーだけが切り替わります。「-」オプションを指定して実行すると、指定したユーザーでログインし直したのと同じ環境が設定されてユーザーが切り替わります。

```
書式
su [-] [ユーザー]

オプション
-
ログインシェルを起動してユーザーを切り替える
```

8.2.2 su コマンドで root に切り替える

su コマンドを実行して、一般ユーザーから root に切り替えます。切り替える際には、root のパスワードを入力する必要があります。インストール時に root のパスワードを設定していない場合、su コマンドで root に切り替

えることはできません。現在ではセキュリティ保護の観点から、root のパスワードを設定せず、su コマンドを実行できないように設定することが多くなっています。

id コマンドを実行すると、ユーザー ID や所属しているグループが確認できます。

```
書式
id [ユーザー]
```

以下の実習を行いたい場合には、次の sudo コマンドの使い方を先に学習して、sudo コマンドを使って root のパスワードを設定した後、実習を行ってください。手順については後述します。

```
$ su -
パスワード: ※rootのパスワードを入力（非表示）
最終ログイン: 2024/08/11 (日) 17:19:01 JST 日時 pts/1
# id
uid=0(root) gid=0(root) groups=0(root)
    context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
# exit
ログアウト
```

8.2.3 sudo コマンドを使って root 権限でコマンドを実行する

sudo コマンドを実行すると、引数として与えられたコマンドを root 権限で実行できます。

普段の作業は一般ユーザーで行ない、必要に応じて sudo コマンドを使ってシステム管理用のコマンドを実行することで、su コマンドなどでユーザーを root に切り替えることなく、root 権限が必要なシステム管理を行えます。

sudo を実行するには、実行する権限が与えられていることと、実行したユーザーのパスワードが必要です。パスワードは 1 回入力すると、デフォルトで 5 分間は省略できます。パスワード入力は設定変更で省略させることもできます。権限付与と設定変更は後述します。

```
書式
sudo コマンド
```

一般ユーザーではアクセスできないファイルも、sudo コマンドを使うと root 権限でコマンドを実行しアクセス権限を得ることができます。

```
$ id linuc
uid=1000(linuc) gid=1000(linuc) groups=1000(linuc),10(wheel)
$ tail /var/log/messages
tail: '/var/log/messages' を 読み込み用に開くことが出来ません: 許可がありません
$ sudo tail /var/log/messages

あなたはシステム管理者から通常の講習を受けたはずです。
これは通常、以下の3点に要約されます:

    #1) 他人のプライバシーを尊重すること。
    #2) タイプする前に考えること。
    #3) 大いなる力には大いなる責任が伴うこと。

[sudo] linuc のパスワード: ※ユーザーlinucのパスワードを入力（非表示）
Aug 15 12:00:13 localhost systemd[15213]: Created slice User Background Tasks
    Slice.
Aug 15 12:00:13 localhost systemd[15213]: Starting Cleanup of User's Temporary
    Files and Directories...
（略）
```

/var/log/messages は一般ユーザーではアクセスできないログファイルでしたが、sudo コマンドで tail コマンドを root 権限で実行したことでアクセスすることができました。

以下の説明で sudo コマンドをつけて実行しているコマンドは、すべて root 権限が必要なシステム管理用のコマンドの実行、あるいは root 権限が必要なファイルに対するアクセスを行っています。

8.2.4 wheel グループ所属を確認する

sudo コマンドは、デフォルトの設定では wheel グループに所属しているユーザーに実行を許可しています。

id コマンドを実行して、自分の所属しているグループを確認します。

```
$ id
uid=1000(linuc) gid=1000(linuc) groups=1000(linuc),10(wheel)
    context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

wheel グループに所属していることがわかります。インストール時に「このユーザーを管理者にする」をチェックして作成したユーザーは、sudo コマンドを実行できるよう wheel グループに最初から所属しています。

8.2.5 sudo コマンドをパスワード無しで実行できるようにする

sudo コマンドの設定を変更して、パスワード無しで sudo コマンドを実行できるようにします。

sudo コマンドの設定は/etc/sudoers に記述されています。この設定ファイルは vi で直接修正するのではなく、visudo コマンドを実行して修正します。

```
$ sudo visudo
[sudo] linuc のパスワード: ※ユーザーlinucのパスワードを入力（非表示）
```

vi で/etc/sudoers が開けたら、以下の設定行を見つけて変更し、保存、終了します（vi の編集モードでコマンド:wq）。設定は即時有効になります。

```
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL
```

↓ 行頭に「#」を入れてコメントアウトする

```
# %wheel  ALL=(ALL)       ALL
```

```
## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL
```

↓ 行頭のコメントアウト「#」を削除して有効にする

```
%wheel        ALL=(ALL)       NOPASSWD: ALL
```

最初の設定は、wheel グループに所属している（%wheel）ユーザーは sudo コマンドを実行できる、パスワードは必要、という設定です。この設定をコメントアウトして無効にしました。コメントアウトするには行頭に「#」と記述します。

次の設定は、wheel グループに所属しているすべてのユーザーは sudo コマンドを実行できる、パスワードは不要（NOPASSWD: ALL）、という設定です。この設定の行頭のコメントアウトを外して有効にしました。

前の sudo コマンド実行から 5 分以上時間をあけて、再度 sudo コマンドを実行してみてください。パスワード入力無しで実行できるようになっています。

8.3 ユーザーの管理

システム管理者は必要に応じて、ユーザーの作成や削除を行います。

たとえば、ログインして何か作業をさせたい場合のほか、メールサーバーであればメールアカウントを発行するためにユーザーを作成します。

8.3.1 ユーザーの作成

新しくユーザーを作成するには useradd コマンドを使います。ユーザー作成には root 権限が必要となります。

ユーザーは必ずグループに所属します。これをプライマリグループ（主グループ）と呼びます。ユーザーはさらに複数のサブグループに所属することもできます。

作成したユーザーの情報は/etc/passwd に記述されます。詳細は後述します。

useradd コマンドでユーザーを作成すると、自動的に/home ディレクトリの下にユーザー名でホームディレクトリを作成して、必要なファイルをコピーしてくれます。

作成したユーザーでログインするためにパスワード認証を行う場合には、後述する passwd コマンドを使ってパスワードを登録する必要があります。

```
書式
useradd ユーザー名

オプション
-g グループ名
プライマリグループ名を指定します。グループ名は/etc/groupで定義したグループ名です。

-G グループ名
サブグループを指定します。複数グループを指定するには「,」（カンマ）で区切ります。
```

以下の実行例では、ユーザー sato を作成します。

```
$ sudo useradd sato
$ grep sato /etc/passwd
sato:x:1001:1001::/home/sato:/bin/bash
$ ls -ld /home/sato
drwx------. 3 sato sato 78  8月 15 14:03 /home/sato
```

ユーザー sato は、sato グループに所属し、ホームディレクトリとして/home/sato が作成されて割り当てられているのがわかります。

8.3.2 パスワードの設定

ユーザーのパスワードを設定するには、passwd コマンドを使います。

一般ユーザーは自分のパスワードのみ設定できます。root はすべてのユーザーのパスワードを設定できます。

```
書式
passwd [ユーザー]
```

以下の実行例では、ユーザー sato のパスワードを変更します。

```
$ sudo passwd sato
ユーザー sato のパスワードを変更。
新しい パスワード: ※新しいパスワードを入力（非表示）
新しい パスワードを再入力してください: ※新しいパスワードを入力（非表示）
passwd: すべての認証トークンが正しく更新できました。
```

root 権限で passwd コマンドを実行すると強制的にパスワードを設定できるので、現在のパスワードを入力する必要がありません。一般ユーザーが自分のパスワードを変更するときには現在のパスワードの入力が要求されます。

8.3.3 root のパスワードを設定して su コマンドを実行する

su コマンドの説明で、OS インストール時に root のパスワードを設定していないため、su コマンドで root に切り替えることができませんでした。

以下の手順で root のパスワードを設定することで、su コマンドで root に切り替えられるようになります。

```
$ sudo passwd root
ユーザー root のパスワードを変更。
新しい パスワード: ※rootの新しいパスワードを入力（非表示）
新しい パスワードを再入力してください:
    ※rootの新しいパスワードを再入力（非表示）
passwd: すべての認証トークンが正しく更新できました。
$ su -
パスワード: ※rootのパスワードを入力（非表示）
# id
uid=0(root) gid=0(root) groups=0(root)
    context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

su コマンドで root に切り替えられました。

8.3.4 ユーザーアカウントの変更

ユーザーのアカウント情報を変更するには usermod コマンドを使います。

```
書式
usermod  ユーザー名

-g グループ
プライマリグループを変更します。グループ名は/etc/groupで定義したグループ名です。

-G グループ
サブグループを変更します。複数グループを指定するには「,」（カンマ）で区切ります。

-a
サブグループの変更（-G）を追加として処理します。
```

以下の実行例では、ユーザー sato のサブグループとして wheel グループを指定します。

```
$ id sato
uid=1001(sato) gid=1001(sato) groups=1001(sato)
$ sudo usermod -G wheel sato
uid=1001(sato) gid=1001(sato) groups=1001(sato),10(wheel)
```

ユーザー sato のサブグループに wheel グループが追加されました。

サブグループを指定する-G オプションは、所属させたいサブグループをすべてカンマ区切りで指定する必要があるため、指定忘れがあるとそのサブグループの所属から外れてしまいます。-a オプションを併用すると、既に所属しているサブグループはそのままに指定したサブグループを追加できます。

```
$ sudo usermod -G kvm -a sato
$ id sato
uid=1001(sato) gid=1001(sato) groups=1001(sato),10(wheel),36(kvm)
```

kvm グループがサブグループとして追加されました。

8.3.5 ユーザーの削除

ユーザーを削除するには userdel コマンドを使います。

```
書式
userdel ユーザー名

オプション
-r
ホームディレクトリを削除します。
```

以下の実行例では、ユーザー test を作成し、削除します。

```
$ sudo useradd test
$ id test
uid=1002(test) gid=1002(test) groups=1002(test)
$ ls -ld /home/test
drwx------. 3 test test 78  8月 15 14:29 /home/test
$ sudo userdel -r test
$ id test
id: `test': no such user
$ ls -ld /home/test
ls: '/home/test' にアクセスできません:
    そのようなファイルやディレクトリはありません
```

-r オプションを付けて実行することで、ホームディレクトリも削除されているのがわかります。

8.4 グループの管理

複数のユーザーをまとめて扱うためにグループを使います。

最初から用意されているグループに加え、システム管理者が必要に応じてグループを作成できます。

グループの定義は/etc/group に記述されます。詳細は後述します。

8.4.1 グループの作成

新しくグループを作成するには groupadd コマンドを使います。

```
書式
groupadd グループ
```

以下の実行例では、test グループを追加し、ユーザー sato のサブグループに設定します。

```
$ sudo groupadd test
$ sudo usermod -G test -a sato
$ grep test /etc/group
test:x:1002:sato
```

ユーザーのサブグループは/etc/group に所属しているユーザーとして記述されます。上記の例では、test グループにユーザー sato が所属しているのがわかります。

8.4.2 グループを削除

グループを削除するには groupdel コマンドを使います。

ユーザーのプライマリグループは削除できませんが、サブグループは削除できてしまうので注意してください。

```
書式
groupdel グループ名
```

以下の実行例では、test グループを削除しています。また linuc グループを削除しようとしています。

```
$ sudo groupdel test
$ grep test /etc/group
※グループが削除されたので何も表示されない
$ sudo groupdel linuc
groupdel: ユーザー 'linuc' のプライマリグループは削除できません。
```

test グループにはユーザー sato が所属していましたが、サブグループだったので削除できました。linuc グループはユーザー linuc のプライマリーグループなので削除できません。

8.5 パスワードファイル/etc/passwd とシャドウファイル/etc/shadow

ユーザーの定義は/etc/passwd(パスワードファイル) に記述されています。

ユーザーに設定したパスワードは、古い UNIX では/etc/passwd に記述されていましたが、現在ではセキュリティのためパスワードは/etc/shadow ファイル (シャドウファイル) に記述されています。

8.5.1 パスワードファイル (/etc/passwd)

ユーザーの情報は/etc/passwd ファイル (パスワードファイル) に保存され、1 行に 1 ユーザーの情報を: で区切って記述します。

パスワードファイルに登録された 1 ユーザーの内容 (1 行) は次の様になります。

```
ユーザー名: パスワード:UID:GID:コメント:ホームディレクトリ:ログインシェル
```

パスワードファイルの内容

項目	内容
ユーザー名	そのシステムでのユーザー名。大文字を含まないようにする
パスワード	以前はユーザーの暗号化されたパスワード、現在は x です
UID	ユーザー ID 番号
GID	ユーザーが属するプライマリグループ ID 番号
コメント	ユーザーの名前またはコメントのフィールド
ホームディレクトリ	ユーザーのホームディレクトリ
ログインシェル	ログイン時に起動されるユーザーのコマンドインタプリタ

パスワードファイルはエディタで直接編集するべきではありません。useradd コマンドなどを使って操作することが推奨されます。

8.5.2 シャドウファイル (/etc/shadow)

ユーザーのパスワードはパスワードファイルではなく、シャドウファイル (/etc/shadow) に保存されます。

シャドウファイルに登録された 1 つのユーザー (1 行) の内容は次の様になります。

```
ユーザー名:パスワード:最終変更日:変更可能日:変更要求日:警告日数:無効日数:失効日数:予約
```

シャドウファイルの内容

項目	内容
ユーザー名	ユーザー名
パスワード	パスワード
最終変更日	1970 年 1 月 1 日から、最後にパスワードが変更された日までの日数
変更可能日数	パスワードが変更可能となるまでの日数
変更要求日数	パスワードを変更しなくてはならなくなる日までの日数
警告日数	パスワード有効期限が来る前に、ユーザーが警告を受ける日数
無効日数	パスワード有効期限が過ぎ、アカウントが使用不能になるまでの日数
失効日数	1970 年 1 月 1 日からアカウントが使用不能になる日までの日数
予約	予約フィールド

シャドウファイルはエディタで直接編集するべきではありません。日付に使われている 1970 年 1 月 1 日は Linux がシステムの基準としている年月日です。

8.5.3 グループファイル (/etc/group)

グループの情報は/etc/group ファイル (グループファイル) に保存され、1 行に 1 グループの情報を: で区切って記述します。

グループファイルに登録された 1 つのグループの内容 (1 行) は次の様になります。

```
グループ名:パスワード:GID:ユーザー
```

グループファイルの内容

項目	内容
グループ名	グループの名前。
パスワード	以前は暗号化されたグループのパスワード、またはパスワードが不要なら空欄
GID	グループ ID 番号
ユーザー	グループに所属するユーザー名のリスト。それぞれのユーザー名はコンマで区切られる。

グループファイルはエディタで直接編集するべきではありません。groupadd コマンドなどを使って操作することが推奨されます。

9 ファイルやディレクトリのアクセス制御

アクセス制御とは、ファイルやディレクトリに対してアクセスを許可したり許可しないようにする仕組みです。Linux はマルチユーザーの OS なので、セキュリティ保護などの観点からファイルに対するアクセス制御が必要です。アクセス制御は、ファイルに対する所有者や所有グループを設定し、それぞれに対するアクセス権を設定することで実現しています。

本章では、ファイルのアクセス制御を行うためのユーザー権限とアクセス権について解説します。

本章の内容

- ファイルの所有者と所有グループ
- ファイルとアクセス権

9.1 ファイルの所有者と所有グループ

ファイルやディレクトリには、所有者（ユーザー）と所有グループが設定されます。この所有権に基づいてアクセス制御が行われます。

ファイルを作成すると、ファイルの所有者は作成したユーザーに、所有グループは作成したユーザーのプライマリーグループに設定されます。

9.1.1 所有者と所有グループの確認

所有者、所有グループは、ls コマンドに-l オプションをつけて実行すれば確認できます。

touch コマンドで test ファイルを作成し、所有者と所有グループを確認します。

```
$ touch test
$ ls -l test
-rw-r--r--. 1 linuc linuc 0  8月 16 10:44 test
```

左側の r や w が並んで表記されている部分がアクセス権です。3 番目の項目が所有者（ユーザー linuc）、4 番目の項目が所有グループ（linuc グループ）です。

9.1.2 所有者の変更

ファイルの所有者を変更するには chown コマンドを使います。指定方法によって所有グループも変更できます。

```
書式
chown [オプション] [ユーザー][:[グループ]] ファイル

オプション
-R
引数で指定したディレクトリ内のファイルやディレクトリの所有者を再帰的に変更。
```

ユーザーとグループを変更するには、root 権限が必要です。

新しい所有者は、以下のパターンで指定できます。

所有者を変更する

```
chown ユーザー ファイル
```

所有者と所有グループを変更する

```
chown ユーザー:グループ ファイル
```

所有グループを変更する

```
chown :グループ ファイル
```

以下の実行例では、作成した test ファイルの所有者を変更します。一般ユーザーでは所有者は変更できないので、sudo コマンドを使って root 権限で chown コマンドを実行します。

```
$ chown sato test
chown: 'test' の所有者を変更中: 許可されていない操作です
$ sudo chown sato test
$ ls -l test
-rw-r--r--. 1 sato linuc 0  8月 16 10:44 test
$ rm test
rm: 書き込み保護されたファイル 通常の空ファイル 'test' を削除しますか?y
```

test ファイルの所有者をユーザー sato に変更できました。最後にファイルを削除しようとすると、所有者が異なるファイルを削除しようとしているため、確認が必要となります。

9.1.3 所有グループの変更

ファイルの所有グループを変更するには chgrp コマンドを使います。

```
書式
chgrp グループ ファイル

オプション
-R
引数で指定したディレクトリ内のファイルやディレクトリの所有グループを再帰的に変更。
```

一般ユーザーは、自分が所属しているグループを所有グループとして指定できます。sudo コマンドを使って root 権限で chgrp コマンドを実行することで、すべてのグループを所有グループとして指定できます。

以下の実行例では、ディレクトリを作成し、一般ユーザーとして所有グループを変更しています。

```
$ id
uid=1000(linuc) gid=1000(linuc) groups=1000(linuc),10(wheel)
    context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
$ mkdir testdir
$ chgrp wheel testdir
$ ls -ld testdir
drwxr-xr-x. 2 linuc wheel 6  8月 16 10:57 testdir
$ chgrp kvm testdir
chgrp: 'testdir' のグループを変更中: 許可されていない操作です
```

サブグループとして所属している wheel グループを所有グループに変更できましたが、所属していない kvm グループには変更できませんでした。

9.2　ファイルとアクセス権

ファイルのアクセス権には「読み込み（Read）」「書き込み（Write）」「実行（eXecute）」の 3 つの種類があります。このアクセス権を、所有者、所有グループ、その他のユーザーの 3 つに対してそれぞれ設定できます。

ファイルへのアクセス権のことを「パーミッション」や「モード」とも呼びます。

ファイルのアクセス権を変更するには chmod コマンドを使います。

9.2.1　アクセス権を確認する

ファイルのモードは、ls コマンドに-l オプションをつけて実行すれば確認できます。一番左がファイルのモードを示しています。

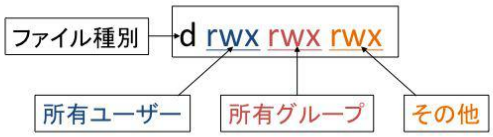

図 42: ファイルのモード

r,w,x の意味

項目	8 進数	内容
r	4	読み込み
w	2	書き込み
x	1	実行、またはディレクトリ内に入れる

rwx は、所有者と所有グループ、その他のユーザーの 3 つに対して指定できます。r が設定されているとファイルやディレクトリの読み込みが可能、w が設定されているとファイルやディレクトリへの書き込みが可能です。x が設定されているとファイルをプログラムとして実行できるか、ディレクトリであればディレクトリに入れます。

9.2.2　アクセス権の変更

ファイルのアクセス権を変更するには chmod コマンドを使います。

```
書式
chmod モード[,モード]... ファイル
chmod 8進数表記のモード ファイル

オプション
-R
引数で指定したディレクトリ内のファイルやディレクトリのアクセス権を再帰的に変更。
```

モードの指定は 2 通りの記述方法があります。

- ユーザー種別毎のモード書式をカンマで区切って指定
- 8 進数 3 桁でまとめて指定

9.2.3　モード書式によるアクセス権の設定

モードは「u」(所有ユーザー)、「g」(所有グループ)、「o」(その他のユーザー) に対して、「r」(読み)、「w」(書き)、「x」(実行またはディレクトリの変更) を「設定 (=)」、「追加 (+)」、「取り消し (-)」します。u、g、o の全てに同じ権限を指定するときは a を指定します。

以下の実行例では、test ファイルにグループの書き込み権限を追加しています。

```
$ touch test
$ ls -l test
-rw-r--r--. 1 linuc linuc 0  8月 16 14:24 test
$ chmod g+w test
$ ls -l test
-rw-rw-r--. 1 linuc linuc 0  8月 16 14:24 test
```

以下の実行例では、testdir ディレクトリにグループとその他のユーザーの書き込み権限を追加しています。

```
$ ls -ld testdir
drwxr-xr-x. 2 linuc wheel 6  8月 16 14:25 testdir
$ chmod g+w,o+w testdir
$ ls -ld testdir
drwxrwxrwx. 2 linuc wheel 6  8月 16 14:25 testdir
```

9.2.4　8 進数によるアクセス権の設定

8 進数によるアクセス権の設定は、設定したいアクセス権を 8 進数の合計値に置き換え、3 桁で指定します。

	ユーザー			グループ			その他		
パーミッション	r	w	x	r	w	x	r	w	x
8進数	4	2	1	4	2	1	4	2	1
設定値	合計値			合計値			合計値		

図 43: ファイルのモード 8 進数表記

たとえば、誰でも読み書きが行えるよう「rw-rw-rw-」と指定する場合、r は 4、w は 2 なので「420 420 420」の合計で「666」となります。「rwxr-xr-x」と指定する場合、x は 1 なので「421 401 401」の合計で「755」となります。

以下の実行例では、ファイルのアクセス権を指定しています。

```
$ ls -l test
-rw-rw-r--. 1 linuc linuc 0  8月 16 14:24 test
$ chmod 666 test
$ ls -l test
-rw-rw-rw-. 1 linuc linuc 0  8月 16 14:24 test
```

「666」と指定したので、アクセス権は「rw-rw-rw-」となりました。

以下の実行例では、ディレクトリのアクセス権を指定しています。

```
$ ls -ld testdir
drwxrwxrwx. 2 linuc wheel 6  8月 16 14:25 testdir
$ chmod 755 testdir/
$ ls -ld testdir
drwxr-xr-x. 2 linuc wheel 6  8月 16 14:25 testdir
```

「755」と指定したので、「rwxr-xr-x」となりました。

9.2.5　アクセス権変更によるアクセス制御の確認

アクセス権を変更して、アクセス制御が行われるところを確認します。

以下の実行例では、まずファイルに書き込み可能な状態で cat コマンドを使って文字列を書き込みます。

```
$ ls -l test
-rw-rw-r--. 1 linuc linuc 0  8月 16 14:24 test
$ cat > test
Write Test.
（行頭でCtrl+dを押す）
```

```
$ cat test
Write Test.
```

書き込みできていることが確認できます。

次に、書き込めていたファイルの所有者に対する書き込みのアクセス権を無くして、書き込みができなくなることを確認します。

```
$ chmod u-w test
$ ls -l test
-r--rw-r--. 1 linuc linuc 12  8月 16 18:10 test
$ cat > test
-bash: test: 許可がありません
```

書き込み権限がなくなったので、書き込みができなくなりました。

以下の実行例では、ディレクトリのアクセス権を確認してみます。書き込み可能なディレクトリ内に touch コマンドで test ファイルを作成します。

```
$ ls -ld testdir
drwxr-xr-x. 2 linuc wheel 6  8月 16 14:25 testdir
$ touch testdir/test
$ ls -l testdir
合計 0
-rw-r--r--. 1 linuc linuc 0  8月 16 18:21 test
```

次に、ディレクトリに対する書き込みのアクセス権を無くすと、ファイルが作成できなくなることを確認します。

```
$ chmod 555 testdir
$ ls -ld testdir
dr-xr-xr-x. 2 linuc wheel 18  8月 16 18:21 testdir
$ touch testdir/test2
touch: 'testdir/test2' に touch できません: 許可がありません
```

書き込み権限がなくなったので、ファイルが作成できなくなりました。

10 ネットワークの設定と管理

Linux はサーバー用途で利用することが多いため、ネットワークに接続することが重要になります。

本章では、Linux をネットワークに接続するために必要とされる基礎知識と確認コマンドと設定を見ていきます。

本章の内容

- IP アドレスの確認
- ping コマンドによる IP 通信の確認
- 名前解決の確認
- ルーティングの確認

10.1 IP アドレスの確認

最初に、使用している Linux に設定されている IP アドレスの確認を行います。IP アドレスの確認は、ip コマンドを使います。

ip コマンドはネットワークに関わる様々な情報の表示が行えるコマンドで、サブコマンドの指定によって動作が変わります。サブコマンドとして「address」を指定することで、ネットワークインターフェースの情報を表示します。インターフェースを省略すると、すべてのインターフェースについて表示します。

```
書式
ip a[ddress] [インターフェース]
```

以下の実行例では、ip a コマンドで IP アドレスその他の情報を表示しています。

```
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
    default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    group default qlen 1000
    link/ether 08:00:27:fb:82:26 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s8
       valid_lft 74465sec preferred_lft 74465sec
    inet6 fd00::a00:27ff:fefb:8226/64 scope global dynamic noprefixroute
       valid_lft 85946sec preferred_lft 13946sec
    inet6 fe80::a00:27ff:fefb:8226/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    group default qlen 1000
    link/ether 08:00:27:a2:d5:45 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.3/24 brd 192.168.56.255 scope global dynamic noprefixroute
        enp0s9
       valid_lft 74464sec preferred_lft 74464sec
    inet6 fe80::a00:27ff:fea2:d545/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

ここでは、3 つのインターフェースがあるのがわかります。

10.1.1 ローカルループバックアドレス

ネットワークインターフェース lo は、ローカルループバックのためのインターフェースです。ローカルループバックは自分自身と通信するための仕組みです。IP アドレスは通常 127.0.0.1 が割り当てられます。

10.1.2　外部通信用の IP アドレス

ネットワークインターフェース enp0s3 が、NAT による外部との通信のために用意されているインターフェースです。IP アドレスとして 10.0.2.15 が割り当てられています。

10.1.3　内部通信用の IP アドレス

ネットワークインターフェース enp0s8 が、ホストオンリーネットワークによるホスト OS との通信のために用意されているインターフェースです。IP アドレスとして 192.168.56.3 が割り当てられています。

10.2　ping コマンドによる IP 通信の確認

コンピューター間の通信は、IP（Internet Protocol）というプロトコルによる通信で成り立っています。

IP による通信が行えることを確認するには、ping コマンドを使います。

```
書式
ping IPアドレスまたはホスト名
```

ping コマンドの引数に IP アドレス、またはホスト名を指定することで、指定先との間で IP 通信が行えるかを確認します。

インターネットに接続できる環境であれば、インターネット上のコンピューターとの接続も確認できます。

以下の実行例では、インターネット上にある linuc.org という名前のサーバーとの間での IP 通信を確認しています。

```
$ ping linuc.org
PING linuc.org (219.94.236.161) 56(84) bytes of data.
64 バイト応答 送信元 219.94.236.161: icmp_seq=1 ttl=255 時間=16.3ミリ秒
64 バイト応答 送信元 219.94.236.161: icmp_seq=2 ttl=255 時間=15.7ミリ秒
^C
--- linuc.org ping 統計 ---
送信パケット数 2, 受信パケット数 2, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 15.668/15.991/16.315/0.323 ms
```

ping コマンドは、停止しない限り繰り返し通信ができるかの確認を行い続けます。Ctrl+c を入力して、停止してください。

ping コマンドの結果には、通信が往復するのにどれぐらい時間がかかったのかも表示されます。海外などネットワーク的に距離があるコンピューターとの通信には時間がかかりますが、距離が近いコンピューターとの通信に時間がかかる場合には、ネットワークが混雑しているか、あるいはネットワーク障害などが起きている可能性があります。

10.2.1　ping コマンドに反応しない場合

ping コマンドの通信は TCP や UDP ではなく、ICMP の Echo という仕組みを使っていますが、相手のコンピューターが Echo に反応しない場合があります。これはセキュリティの関係上、ICMP Echo に反応して存在を知られないようにするためです。ping コマンドに反応しないからといって、IP による通信が行えないわけではないことに注意が必要です。

10.3　名前解決の確認

名前解決とは、ホスト名を IP アドレスに変換する仕組みのことです。Web サーバーやメールサーバーでやり取りをするには必須となる技術です。

すべてのインターネットの通信を IP アドレスを直に指定して行うのは不便です。そこで、外部からのアクセスを受け付けたい組織はドメイン名を取得し、インターネットに接続するコンピューターにドメイン名（正確にはホスト名）を割り当てます。割り当てたホスト名と IP アドレスの組み合わせは、名前解決で調べられるようにしておきます。利用者は名前解決を行って、ホスト名から IP アドレスを取得して通信を行います。

名前解決の仕組みは様々なものがありますが、最もよく使われているのが DNS（Domain Name System）です。DNS を使って、名前解決ができるかどうかを確認するには、dig コマンドを使います。

```
書式
dig ホスト名
```

以下の実行例では、dig コマンドを使って linuc.org を名前解決しています。

```
$ dig www.linuc.org

; <<>> DiG 9.16.23-RH <<>> www.linuc.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37099
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.linuc.org.          IN   A

;; ANSWER SECTION:
www.linuc.org.       3588    IN   A    219.94.236.161

;; Query time: 20 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Mon Sep 16 14:14:49 JST 2024
;; MSG SIZE  rcvd: 58
```

名前解決が行われて、ANSWER SECTION に IP アドレスが表示されているのがわかります。

10.3.1　参照している DNS の確認

DNS は、名前解決を行ってくれる DNS サーバーが動作しており、その DNS サーバーに対して名前解決のリクエストを送ることで IP アドレスがわかります。参照している DNS を間違えていると、名前解決が行えなくなります。

参照している DNS は、/etc/resolv.conf に記述されています。

```
$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 10.0.2.3
nameserver fd00::3
```

この設定では、IP アドレス 10.0.2.3 を DNS サーバーとして参照しているのがわかります。この設定ファイルを直接修正して動作を変更することもできますが、1 行目に書かれている通り、この設定ファイルは NetworkManager という仕組みでシステム起動時に自動的に生成されています。設定変更は後述する NetworkManager の設定変更方法に従って行ってください。

10.4　ルーティングの確認

コンピューターがネットワークで通信するためには LAN（Local Area Network）に接続します。同じ LAN に接続されているコンピューター同士は直接通信を行えます。一方、その他の LAN に接続されているコンピューターやインターネット上に接続されているコンピューターと通信を行うには、ルーターを通じて通信を行う必要があります。ルーターを介して外部ネットワークと通信を行うことを「ルーティング」と呼びます。

ルーティングを行うためのデフォルトのルーターのことを「デフォルトゲートウェイ」と呼びます。

コンピューターに設定されているルーティングを確認するには、ip route コマンドを実行します。

```
$ ip route
default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
192.168.56.0/24 dev enp0s8 proto kernel scope link src 192.168.56.3 metric 101
```

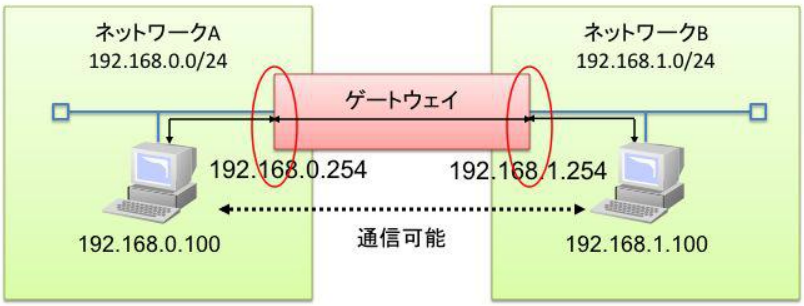

図 44: ルーティングの例

1 行目は、デフォルトゲートウェイが 10.0.2.2 に設定されており、通信にはネットワークインターフェース enp0s3 を使用することを表しています。2 行目は、このコンピューターが NAT で接続されているネットワーク（10.0.2.0/24）と通信するには、ネットワークインターフェース enp0s3 を使用すると設定されていることを表しています。3 行目は、このコンピューターがホストオンリーネットワークで接続されているネットワーク（192.168.56.0/24）と通信するには、ネットワークインターフェース enp0s8 を使用すると設定されていることを表しています。

外部との通信が行えない場合には、まずコンピューター自身のルーティングを確認するようにしてください。

11 プロセス管理

Linux のシステムは、様々なプログラムが動作して構成されています。動作しているプログラムを Linux は「プロセス」として扱います。

本章では、プロセスの確認や管理について解説します。

本章の内容

- プロセスとは
- top コマンド
- シグナルによるプロセスの制御

11.1 プロセスとは

Linux では実行中のプログラム（アプリケーション）を「プロセス」として扱います。

コマンドを実行するために使っているシェル自身もプロセスです。シェルからコマンドを実行すると、プログラムが起動されてプロセスとなり、Linux カーネルが管理します。プログラムが終了すると、プロセスは無くなります。

プロセスがプロセスを生成する場合、生成した側を親プロセス、生成された側を子プロセスと呼びます。シェルからコマンドを実行する場合、シェルが親プロセス、コマンドが子プロセスとなります。

11.1.1 ps コマンドでプロセスを確認する

動作しているプロセスを確認するには、ps コマンドを使います。ps コマンドはオプションをつけることで様々なプロセスの情報を表示できます。

```
書式
ps [オプション]

オプション
a
すべてのプロセスを表示する

f
プロセスの親子関係を表示する

u
実行ユーザーを表示する

x
制御端末のないバックグラウンドプロセスも表示する
```

ps コマンドを、オプション無しで実行します。

```
$ ps
    PID TTY          TIME CMD
   3306 pts/1    00:00:00 bash
   3356 pts/1    00:00:00 ps
```

現在使用しているシェルと ps コマンドのみ表示されます。

11.1.2 プロセスの親子関係を確認する

f オプションをつけて実行します。プロセスの親子関係が表示されます。

```
$ ps f
    PID TTY      STAT   TIME COMMAND
   2618 pts/0    Ss+    0:00 bash
   3362 pts/0    R+     0:00  \_ ps f
   1816 tty2     Ssl+   0:00 /usr/libexec/gdm-wayland-session --register-session
   1823 tty2     Sl+    0:00  \_ /usr/libexec/gnome-session-binary
```

ps コマンドのプロセスがシェルから起動された子プロセスであることがわかります。また、GUI 環境を利用するために必要なプロセスが別に動作しているのがわかります。

11.1.3 プロセスを実行したユーザーを確認する

u オプションをつけて実行します。プロセスを実行したユーザーの情報が表示されます。

```
$ ps u
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
linuc       1816  0.0  0.4 375108  7644 tty2     Ssl+ 10:30   0:00 /usr/libexec/
linuc       1823  0.0  1.0 518668 17776 tty2     Sl+  10:30   0:00 /usr/libexec/
linuc       2618  0.0  0.3 223940  5120 pts/0    Ss+  10:32   0:00 bash
linuc       3368  0.0  0.1 225488  3072 pts/0    R+   15:03   0:00 ps u
```

ユーザー linuc がプロセスを実行していることがわかります。

11.1.4 バックグラウンド動作しているプロセスを確認する

シェルから起動したプロセス以外に、様々なプロセスが動作して Linux のシステムを構成しています。それらのプロセスも ps コマンドで確認できます。

ps コマンドを x オプションをつけて実行します。制御端末のないバックグラウンドで動作しているプロセスの情報も表示されます。

```
$ ps x
    PID TTY      STAT   TIME COMMAND
   1786 ?        Ss     0:00 /usr/lib/systemd/systemd --user
   1788 ?        S      0:00 (sd-pam)
   1807 ?        Sl     0:00 /usr/bin/gnome-keyring-daemon --daemonize --login
   1816 tty2     Ssl+   0:00 /usr/libexec/gdm-wayland-session --register-session
 （略）
   2618 pts/0    Ss+    0:00 bash
   3369 pts/0    R+     0:00 ps x
```

2 番目の TTY の列が制御端末を表しています。この項目が?になっているプロセスは制御端末が無いので、バックグラウンドで動作しています。

11.1.5 制御端末のあるすべてのプロセスを確認する

Linux のシステム全体のプロセスを確認するには、a オプションをつけて実行します。制御端末のあるすべてのプロセスの情報が表示されます。

```
$ ps a
    PID TTY      STAT   TIME COMMAND
   1816 tty2     Ssl+   0:00 /usr/libexec/gdm-wayland-session --register-session
   1823 tty2     Sl+    0:00 /usr/libexec/gnome-session-binary
   2618 pts/0    Ss+    0:00 bash
   3376 pts/0    R+     0:00 ps a
```

GUI のためのプロセスが動作しているのがわかります。

11.1.6 すべてのプロセスを確認する

ax オプションをつけて実行します。制御端末のないバックグラウンドで動作しているプロセスもすべて表示されます。

```
$ ps ax
    PID TTY      STAT   TIME COMMAND
      1 ?        Ss     0:00 /usr/lib/systemd/systemd rhgb --switched-root --sys
      2 ?        S      0:00 [kthreadd]
 （略）
```

カーネル起動後に最初に起動されるプロセスがプロセス ID1 番となりますが、systemd が最初に起動しているのがわかります。

11.1.7 pstree コマンド

プロセスの親子関係をツリー表示するには、pstree コマンドを使います。ps f コマンドでも親子関係は表示できますが、より見やすい表示が行えます。

```
$ pstree
systemd─┬─ModemManager───3*[{ModemManager}]
        ├─NetworkManager───2*[{NetworkManager}]
 （略）
```

11.2 top コマンド

プロセスの状態を確認するツールとして、top コマンドがあります。

top コマンドは実行中のプロセスの状態をリアルタイムで表示します。プロセスを CPU やメモリの使用率でソートしたり、システム全体のリソース負荷を確認できます。

top コマンドのキー操作

キー	動作
?	ヘルプを表示する
スペース	表示を更新する
1	CPU 別に CPU の使用率を表示する
P	CPU 使用率でプロセスをソート
M	メモリ使用率でプロセスをソート
< >	ソートのための項目を左右に選択する
x	ソートのために選択している項目をハイライト
b	ソートのために選択している項目を分かりやすくする
q	終了する

top コマンドを実行して、各キー操作の動作を確認してみてください。

```
$ top
```

「b」で選択している項目を分かりやすくする動作だけは、先に「x」で選択をハイライトにしてからでないと動作が分かりにくくなっています。「xb」と入力してみてください。

11.3 シグナルによるプロセスの制御

プロセスは処理が正常に終わって終了したり、エラーを起こして異常終了する他、外部からシグナルを送ることで処理を停止させたり、終了させたりすることができます。

コマンドラインでプロセスにシグナルを送信する方法は 2 つあります。

- 端末でキー入力（例：Ctrl+c や Ctrl+z など）
- kill コマンド

11.3.1 シグナル番号とシグナル名

シグナルには、シグナル番号およびシグナル名が割り当てられており、代表的なものに以下のシグナルがあります。

代表的なシグナル

シグナル番号	シグナル名
1	HUP
2	INT

シグナル番号	シグナル名
9	KILL
15	TERM
18	TSTP

上記以外にもシグナルがあり、kill コマンドに-l オプションをつけて実行することでシグナルの種類を表示することができます。

```
$ kill -l
 1) SIGHUP   2) SIGINT   3) SIGQUIT  4) SIGILL
 5) SIGTRAP  6) SIGABRT  7) SIGEMT   8) SIGFPE
 9) SIGKILL 10) SIGBUS  11) SIGSEGV 12) SIGSYS
13) SIGPIPE 14) SIGALRM 15) SIGTERM 16) SIGURG
17) SIGSTOP 18) SIGTSTP 19) SIGCONT 20) SIGCHLD
21) SIGTTIN 22) SIGTTOU 23) SIGIO   24) SIGXCPU
25) SIGXFSZ 26) SIGVTALRM   27) SIGPROF 28) SIGWINCH
29) SIGINFO 30) SIGUSR1 31) SIGUSR2
```

11.3.2 kill コマンドによるシグナル送信

kill コマンドでプロセスにシグナルを送信してみます。

```
書式
kill [オプション] プロセスID

オプション
-シグナル番号
指定したシグナル番号のシグナルをプロセスに送信する
```

以下の実行例では、実行中のプロセスに kill コマンドでシグナルを送信して、プロセスを終了しています。

```
$ sudo tail -f /var/log/messages &
[1] 22385
（略）
$ kill 22385
$ ※メッセージが表示されるようにEnterキーを入力
[1]+  Terminated              sudo tail -f /var/log/messages
```

&をつけてコマンドを実行すると、そのコマンドはバックグラウンドジョブとしてプロセスが実行されたままになり、制御はコマンドプロンプトに戻ります。tail -f コマンドは終了の指示を受けない限り引数で指定されたファイルに書き込まれた情報を標準出力に表示し続けます。

シェルプロンプトに戻った際に表示されているのが、ジョブ番号とプロセス ID です。kill コマンドの引数としてこのプロセス ID を指定することでプロセスが終了しました。

11.3.3 シグナル指定の使い方

オプションでシグナルを指定しない場合、デフォルトで 15 番の TERM（SIGTERM）が送信されます。プログラムが暴走してしまい制御できなくなったプロセスを強制的に終了するためにシグナル番号 9 番（KILL）を送信するなどの使い方をします。

12　腕試しの課題とまとめ

仮想マシンの作成から OS のインストール、各種コマンドの使い方の解説と進めてきました。

一通り終わったら、仮想マシンを削除（除去）して、再度 1 から手順を復習してみてください。復習が終わったら、今度は目次だけ見ながら同じ手順を繰り返してみてください。目次を見ただけでどんな内容か分からなかったところがあれば、あらためて教科書の解説や実習を試してみてください。

3 回ほど本教科書をこなしたら、腕試しの課題に挑戦です。

12.1　腕試しの課題

ここまで学んできたことの 1 つ 1 つは Linux を操作するというパズルのピースのようなものですが、全体像としての Linux 操作はまだ見えにくいかも知れません。

次のステップとして『Linux サーバー構築標準教科書』に挑戦してみて欲しいのですが、その前にこの教科書の範囲内でも可能な腕試しの問題を出してみようと思います。この教科書の内容＋ α で解けるので、挑戦してみてください。

12.1.1　問題：ユーザーディレクトリ機能の有効化

Apache Web サーバーには、ユーザーディレクトリという機能があります。この機能は、ユーザーのホームディレクトリに「public_html」というディレクトリを作成し、その中に HTML ファイルを設置すると「http://サーバーアドレス/~ユーザー名/HTML ファイル名」として参照できるというものです。

この機能を有効にするには、以下の作業が必要です。

- Apache Web サーバーの設定ファイルである/etc/httpd/conf.d ディレクトリの userdir.conf ファイルの設定を変更して、Apache Web サーバーを再起動する
- ユーザーを作成し、そのユーザーのホームディレクトリに public_html ディレクトリを作成して、必要なアクセス権を設定する
- ~/public_html ディレクトリに index.html などのファイルを置いて、Web ブラウザからアクセスしてみる

userdir.conf ファイルのコメントにも英語ですが設定内容についての記載があるのでヒントとして参照して、ユーザーディレクトリ機能を有効にしてみてください。

12.1.2　セキュリティの対応

1 点だけ、userdir.conf ファイルにも書かれていない、セキュリティに関わる対応が必要となります。すべての設定が終わったら、ユーザーディレクトリ機能で Web ページを公開したいユーザーで実行してください。

```
$ chcon -R -t httpd_sys_content_t ~/public_html/
```

これは Linux のセキュリティ機能の 1 つである SELinux に関する設定です。作成した~/public_html ディレクトリに対して SELinux が参照する値を設定するコマンドです。

SELinux については『Linux システム管理標準教科書』で解説しているので、参考にしてください。

12.1.3 解答

どうしてもやり方が分からない、解答が知りたい、という人は以下の記事を参照してみてください。

```
https://linuc.org/study/column/4513/
```

図 45: https://linuc.org/study/column/4513/

12.2 まとめ

ここまで実践できるようになれば、基本的なサーバー構築の手順をこなすことはできるようになったはずです。次のステップとして、個々の作業の意味を理解する、全体的な作業の流れを理解することができるように、とにかくいろいろなサーバーを構築してみたり、今回使用した AlmaLinux 以外のディストリビューションもインストールして試してみてください。

Linux標準教科書

2026年2月18日　Ver.4.0.1

発行元　LPI-Japan