

13 【付録B】初めてのLinux環境

本章では、筆者が担当した実習科目にて、大学に入学したばかりの1年生前期の学生に、プログラミングそのものの知識と並行して教える必要があったLinuxの知識や操作方法について、必要最低限のものを列挙したものです。本書の主旨から外れる内容なので削除してもよかったです、とりあえず付録として掲載しています。あくまでも筆者の非常勤先大学の計算機環境を説明する章ですので、必ずしも読者の環境にて同一の操作ができるとは限らないと考えて下さい。

13.1 Linuxとは

皆さんは大学の計算機の講義の中で、きっとオペレーティングシステム という単語を習うことでしょう。ここではごく簡単な説明にとどめますが、オペレーティングシステムとは、計算機を構成する各種装置（ハードウェア）に直接命令を送るプログラムのことであり、計算機の電源を入れたときにハードウェアが最初に読み込むプログラムと考えることもできます。

オペレーティングシステムの中でも有名で、皆さんの日常生活に浸透しているものといえば、Microsoft社によるWindowsと、Apple社によるMac OSです。家庭用パソコンのための商用オペレーティングシステムといえば、この2つが10年以上の長い間にわたって双壁として活躍しています。

それに対して、主に業務用や研究用の目的で、WindowsやMac OSよりも長い歴史を誇るオペレーティングシステムにUNIX（ユニックス）というものがあります。UNIXは1970年代から開発が進められているオペレーティングシステムで、主にワークステーションという（一般的にパソコンよりも高価な）計算機で活躍していました。

Linuxは、UNIXを専用ワークステーションではなくパソコン用に開発し、しかも無料で配布する（現在では商用製品も多数存在しますが）という、画期的な試みの中で生まれたオペレーティングシステムの代表です。Linuxの中核部分は、1991年に当時フィンランドのヘルシンキ大学在学中であったリーナス・トーバルズが個人で開発を開始したものであり、彼のファーストネームをとってLinuxと呼ばれています。

Linuxの発音は、リナックス、リーヌクス、リヌックス、リヌクス、ライナックス、リーナクス、などさまざまなものが聞かれます。筆者はリナックスと発音しています。

13.2 Linuxシステムにログインする

現在の大半のオペレーティングシステムでは、マルチユーザ という考え方を導入しています。マルチユーザとは簡単にいうと、複数の利用者が同一の計算機を共有する仕組みのことです。

マルチユーザなオペレーティングシステムでは、各利用者の設定や情報は、明確に区別されて保存されています。そのため利用者が計算機を使い始めるときには、自分が誰であるかを計算機に対して名乗り、それが計算機に認められて初めて、自分の設定や情報が計算機上で有効になり、計算機を利用できる状態になる、というような仕組みが搭載されています。この「自分が誰であるかを計算機に対して名乗る」という行為がログイン というものです。

このログインという仕組みは、大学の端末室のような環境では非常に便利です。なぜなら皆さんは、大学の端末室では、いつも同じ席に座るとは限らないからです。マルチユーザなオペレーティングシステムでは、ログインの操作さえ正しくできれば、毎回違う席に座っていても、毎回同じ設定や情報を有効に活用できるようになります。ですので皆さんは、端末室でのプログラミング実習では、自分がどこに座っているかを意識する必要はありません。

ログインをするときに、利用者がキーボードを叩いて示す情報は、大抵の場合は以下の2つです。

ユーザ名：利用者の名前のようなもの。他者にも公開されている。

パスワード：利用者だけが知っている文字列。

ユーザ名とパスワードは、例えるなら銀行の口座番号と暗証番号のような関係にあります。口座番号は他者に知らせる情報であり、暗証番号は自分以外の誰にも知らせない情報です。

ですので今後、皆さんは端末室を利用するにあたり、

- ユーザ名とパスワードを忘れないようにする
- パスワードを他者に知らせない

という点を徹底するよう、くれぐれもお願いします。

なお今後、利用者の皆さんが計算機に向かって、キーボードやマウスを使って情報を与えることを、入力する と称します。逆に計算機が、利用者の皆さんに向かって、画面やスピーカやプリンタを使って情報を与えることを、出力する と称します。

以下に、ユーザ名とパスワードの入力手順を示します。

1. キーボードを叩くと、画面上のユーザ名を記入する欄に、皆さんの叩いた文字が表示されることを確認してください。¹⁰
2. 皆さんのユーザ名をキーボードで入力し、最後に「Enter」を押してください。この「Enter」を押す、という行為がないと、ユーザ名の入力は有効になりませんので、注意して下さい。
3. ここまで正常に操作できれば、画面上のパスワードの記入欄にキーボード入力できる状態になっているはずですので、確認してください。¹¹
4. 皆さんのパスワードをキーボードで入力し、最後に「Enter」を押してください。この「Enter」を押す、という行為がないと、ユーザー名の入力は有効になりませんので、注意して下さい。また一般的に、パスワードは画面表示されません。¹²慣れないうちは、ついキーボード入力を間違えそうになるかと思いますが、注意して入力してください。
5. ユーザ名またはパスワードが間違っている場合には、それを警告する表示が出てくるはずですが、正しく入力できた場合には、ユーザ名とパスワードの記入欄が消えて、新しい画面表示内容が見えてくるはずですが。

どうしてもユーザ名やパスワードを入力できない場合、あるいはユーザ名やパスワードをまだ取得していない場合には、教員やTAをつかまえて、なんとかしてください。

13.3 ログイン後の操作とログアウト

ログインに成功すると、画面上の表示が大きく変わります。おそらく皆さんの画面には、アイコン と呼ばれるイラスト風の小さいボタンや、メニュー と呼ばれる選択肢が表示されていることでしょう。

このアイコンやメニューは、皆さんのマウス操作によって、計算機上の特定のプログラムを動かすために提供されているものです。このように、アイコンやメニューなどの視覚的な部品を使って、キーボードだけでなくマウスも使って計算機を操作する仕組みを、GUI(グラフィカル・ユーザ・インタフェース)といいます。また、このような操作によって起動される特定のプログラムを、アプリケーション といいます。また、画面上でアプリケーションが占有する、窓のような長方形領域を、ウィンドウ といいます。また、多数のウィンドウを操作する仕組みの整ったGUIを、ウィンドウシステムといいます。

作業を終了して端末室を出るとき、皆さんは必ず、ログアウト という操作をしなければなりません。この操作は、計算機を「ログイン操作ができる状態」に戻すことを指します。

¹⁰もしユーザ名の記入結果が表示されない場合には、画面上の矢印記号(カーソル)がユーザ名の記入欄を指すように、マウスで移動させてください。そこでマウスの左ボタンを押す(クリックする)ことによって、ユーザ名の記入欄を使えるようになります。

¹¹もしパスワードをキーボード入力できない場合には、画面上の矢印記号(カーソル)がパスワードの記入欄を指すように、マウスで移動させてください。そこでマウスの左ボタンを押す(クリックする)ことによって、パスワードの記入欄を使えるようになります。

¹²多くのシステムでは、パスワード入力時に、「****」という伏せ字が表示されます。

皆さんがログアウト操作をしないで端末室を退出してしまうと、この計算機を他の人が使うことはできません。それどころか、悪意ある人が、皆さんになりすまして計算機を使ってしまい、皆さんが保存している大切な情報を失ってしまう可能性があります。これらのことから、

- 退出時にはログアウトを忘れない

という点を、くれぐれも徹底して下さい。

ログアウトの方法は以下の通りです。

1. 画面左上の「アクション」をマウスでクリックする。
2. 選択肢が下に伸びるようにして一覧表示されるので、その中から「ログアウト」をマウスでクリックする。
3. 「本当にログアウトしてもよろしいですか？」という表示が出たら、アクション欄で「ログアウト」をマウスで選択して、「OK」をマウスでクリックする。

13.4 ターミナルとブラウザの起動

非常勤先大学にて筆者の担当科目を受講するにあたり、真っ先に覚えたいといけないうアプリケーションが3つありました。それはターミナル、ブラウザ、テキストエディタです。このうちテキストエディタについては 13.6 節で後述するとして、ここではターミナルとブラウザについて説明します。

13.4.1 ターミナル

ターミナルとは、キーボード入力によって計算機を操作するためのアプリケーションです。ターミナルは以下のような操作によって起動されます。

1. メニューの「アプリケーション」をマウスで選ぶ。
2. メニューの「システム・ツール」をマウスで選ぶ。
3. メニューの「GNOME 端末」をメニューで選ぶ。

以下の資料では、GNOME 端末を「ターミナル」と称します。

13.4.2 ブラウザ

ブラウザとは、計算機ネットワークを通して発信されている情報を閲覧するためのアプリケーションです。最近では、俗に「ネットを使う」というと、ブラウザを通して情報を閲覧することを指す場合が多いようです。

ブラウザで表示するための情報発信の基盤となるネットワーク技術を、一般的に World Wide Web または単純に Web(ウェブ:クモの巣) と呼びます。ブラウザは正確には Web ブラウザとも呼ばれます。

13.5 Linux 上での主要なコマンド

続いて、ターミナル上で入力できる Linux 主要なコマンドについて説明します。なお本書では、筆者が非常勤先大学の担当科目で実際に紹介した最小限のコマンドだけを説明します。これ以上の詳しい内容については、検索エンジンを用いて Web ブラウザ上で調べるか、専門書を購入するか、のいずれかにしてください。

なお以下に紹介するコマンドはすべて、単にその文字を入力するだけでなく最後に Enter を押して初めて有効になるという点に、くれぐれも注意して下さい。

13.5.1 ファイル

計算機ではデータや文書を **ファイル** という単位で保存します。おそらく皆さんは、何枚かの紙をひとまとめにする容器をファイルと呼んだことがあるかと思いますが、あれと同様に、ひとまとめのデータや文書を計算機ではファイルという単位で扱います。

これから作成される皆さんのプログラムも、各々が 1 個のファイルとして保存されます。

13.5.2 ディレクトリ (またはフォルダ)

いくつかのファイルをまとめて保管する入れ物を、**ディレクトリ** または **フォルダ** といいます。歴史的に Linux をはじめとする UNIX 系のオペレーティングシステムではディレクトリという単語が使われていますが、ウィンドウシステムではフォルダと呼ぶことが多いため、現在では両者は同義語として混在しています。原則としてディレクトリと呼ぶことにします。

ディレクトリは二重三重の入れ子を構成することができます。言い換えれば、ディレクトリの中にディレクトリを作ることが可能です。これは例えるなら、大学の中に学部があり、学部の中に学科があり、というような上下関係を作ることと類似しています。このような入れ子の構造、または上下関係をもつ構造を、**階層構造** と呼びます。

ターミナルを操作する際には常に、いま自分がどこのディレクトリの中にいるか、という状態を意識する必要があります。自分がいま所在するディレクトリを **カレントディレクトリ** といいます。ターミナル上でカレントディレクトリを表示するためには、

```
pwd
```

と入力してください。これでカレントディレクトリが表示されます。例えば

```
/home/itot
```

と表示されたとします。これは「home というディレクトリの中に、itot というディレクトリがあって、自分はいま itot というディレクトリの中にいます」ということを意味しています。

13.5.3 ディレクトリの作成

前節にて「ディレクトリの中にディレクトリを作ることが可能です」と書きました。そこで続いて、自分のカレントディレクトリの中に新しいディレクトリを作成する方法を説明します。いま、「itot というカレントディレクトリの中に、java というディレクトリを作成したい」とします。このようなときには、

```
mkdir java
```

と入力してみてください。ここで `mkdir` というコマンドは、カレントディレクトリの直下に、新しいディレクトリを作成する、という意味を持ちます。

13.5.4 ディレクトリの移動

続いて、カレントディレクトリを移動する方法について説明します。いま、「itot というカレントディレクトリの中に java というディレクトリがあって、自分はいまそこに移動したい」とします。このようなときには、

```
cd java
```

と入力してみてください。ここで cd というコマンドは、カレントディレクトリから、その直下にあるディレクトリに移動する、という意味を持ちます。上記のコマンドのあとに pwd コマンドを行うと、きっと

```
/home/itot/java
```

と表示されるのではないかと思います。

続いて、「カレントディレクトリの1個上のディレクトリに移動したい」とします。このようなときには、「1個上のディレクトリ」を意味する..(ピリオド2個)という記号を使って

```
cd ..
```

と入力してください。上記のコマンドのあとに pwd コマンドを行うと、きっと

```
/home/itot
```

と表示されるのではないかと思います。

さて皆さんは、ログインをしてターミナルを起動すると、原則として毎回同じディレクトリから作業を開始することになるはずですが、一般的にオペレーティングシステムでは、各々の利用者に対して、作業の開始場所となるディレクトリを指定することができます。この「作業の開始場所となるディレクトリ」を、ホームディレクトリと呼びます。ターミナル上でホームディレクトリに戻る際には、

```
cd
```

と入力してください。cd の後に何も打たないで Enter を押すと、ホームディレクトリに戻る、とっていただければと思います。

13.5.5 ディレクトリの中身を見る

演習が進むと、おそらくプログラムのファイルがたくさんつくられることと思います。自分のディレクトリの中に、どんなファイルが生成されているかを確認するには、

```
ls
```

というコマンドを入力してみてください。自分が作成した、末尾（拡張子）が `java` であるプログラムファイルの名前がそれに似た名前で拡張子が `class` であるファイルの名前が表示されるでしょう。本書に沿ってプログラミングを進めている途中であれば、例えば

```
Bmi.class Bmi.java Calculate.class Calculate.java
```

というように表示されるのではないかと思います。

なお、ファイルの大きさや更新日時などの付随情報も同時に表示したい場合には、

```
ls -l
```

というように、「`-l`」をつける必要があります。

13.5.6 ファイルのコピー

プログラミングの課題では、似たような複数のプログラムを作成することがあります。このとき各々のファイルを、1行目から作成しなおすのは非効率ですよね。そこでプログラミングの多くの場面では、まず似たような既存のプログラムファイルをコピーして、そのコピーしたファイルを編集して新しいプログラムを作ります。

ファイルをコピーするとき、例えば `Hello.java` というファイルを `Calculate.java` というファイルにコピーするときには、

```
cp Hello.java Calculate.java
```

というように、`cp` というコマンドを使い、続いてコピー元ファイル名、コピー先ファイル名、という順に入力してください。ここで、コピー元ファイル名と、コピー先ファイル名の順番を逆にしないよう、くれぐれも注意して下さい。

また、上記のコマンドは `Hello.java` が置かれているディレクトリにて実行すれば有効になります。他のディレクトリをカレントディレクトリにしているときに上記のコマンドを実行する場合には、「パス」という考え方で `Hello.java` および `Calculate.java` の置き場所を指示する必要があります。本章では説明を割愛します。

13.5.7 ファイルの移動

ファイルのコピーに似たコマンドとして、移動というコマンドがあります。移動というよりは「ファイル名の変更」と考えたほうが、わかりやすいかもしれません。ファイルを移動するとき、例えば `Hello.java`

というファイルを `Calculate.java` というファイル名に変えるときには、

```
mv Hello.java Calculate.java
```

というように、`mv` というコマンドを使い、続いて移動元ファイル名、移動先ファイル名、という順に入力してください。ここでも、移動元ファイル名と、移動先ファイル名の順番を逆にしないよう、くれぐれも注意して下さい。

13.5.8 ファイルの削除

不要になったファイルを削除する方法について説明します。例えば `Unnecessary.java` というファイルを削除するには、

```
rm Unnecessary.java
```

というように、`rm` というコマンドを使い、続いて削除するファイル名を書いてください。なお、`rm` コマンドによって削除されたファイルは復活不可能 ですので、`rm` コマンドを使うときは、くれぐれも慎重をお願いします。

13.5.9 Java のコンパイルと実行

Java 言語のプログラムを開発したら、コンパイルの必要があります。例えば `Hello.java` というファイルをコンパイルするには、

```
javac Hello.java
```

というように、`javac` コマンドを使ってください。`javac` のあとには、コンパイル対象となるファイル名を記載します。

カレントディレクトリにあるプログラムファイルを全てコンパイルするときには、

```
javac *.java
```

というように実行してください。ここで「*」とは「全ての」という意味をもつ記号です。このような記号を使ってファイル名などを表現することを、正規表現 といいます。正規表現に用いる記号には「*」以外にもいくつかあります。詳しくは検索エンジンで調べるか、専門書で独習して下さい。

Java 言語で開発されたプログラムを実行するときには、

```
java Hello
```

というように、`java` コマンドを使ってください。ここで `java` コマンドの後の文字列には、`main` を持つクラスのクラス名、いいかえればプログラムファイル名から「.java」を除いたものを使います。

13.6 Emacs で文字情報を編集する

Java 言語のプログラムファイルをはじめとする文字情報を編集する際には、テキストエディタ というアプリケーションを用います。Windows では「メモ帳 (Notepad)」が、Mac OS では「シンプルテキスト」が、パソコン購入時に付随されている簡易なテキストエディタの代表格といえるでしょう。

本章では「Emacs」というテキストエディタを紹介します。Emacs は、Windows でも Mac OS でも Linux でも共通に使えるテキストエディタの代表的なものであり、大学に限らず非常に多くの現場に普及しています。

Emacs を使って特定のファイルを編集するときには、ターミナル上で

```
emacs Hello.java
```

というように、emacs コマンドのあとにファイル名を書いて実行します。既に存在するファイルであれば、そのファイルの中身が表示されます。初めて編集するファイルであれば、白紙の状態 Emacs を起動します。

ただし、このコマンドで Emacs を実行すると、それを実行したターミナルに別のコマンドを入力できない状態になってしまいます。それを避ける方法として本講義では、

```
emacs Hello.java &
```

というように、最後に&をつけて起動することを推奨します。

ここまで出来たら、Emacs のウィンドウ上でプログラムを編集し、保存して下さい。保存の操作は以下の2通りのいずれかです。

- Ctrl キーと x キーを同時に押し、続いて Ctrl キーと s キーを同時に押す
- メニューで「ファイル」の「保存」を選ぶ

正しく保存できれば、Emacs のウィンドウの左下端に「Wrote ...」というメッセージが表示されます。これを確認できましたら、正常に保存できたと思ってください。

Emacs には、他にも非常に多くの機能がありますが、ここでは最小限の解説にとどめます。これ以上の詳しい内容については、検索エンジンを用いて Web ブラウザ上で調べるか、専門書を購入するか、のいずれかにしてください。

13.7 困ったらここを読もう

筆者が非常勤講師としてプログラミングを実習するにあたり、勤務先大学の環境にて多くの学生から出てきた質問に対する回答を、以下にまとめてみました。

[質問] ターミナルに何を入力しても反応しない・Emacs を起動するとき行末に&を入れるのを忘れた

Emacs を起動するとき、例えば

```
emacs Hello.java &
```

というように、文末に&を入れることを強く推奨しています。これを入れ忘れると、Emacs を終了するまでの間は、ターミナル (GNOME 端末) に何をキーボード入力しても受け付けてくれません。

もし Emacs の起動時に&を入れ忘れたら、ターミナルにて Ctrl キーと z キーを同時に押し、続いて bg と入力して Enter キーを押すという操作を試みてください。この操作によって、すでに起動していた Emacs は、文末に&をつけて起動したのと同じ状態になります。そして、ターミナルはキーボード入力を受け付けるようになります。

[質問] プログラムを書き換えても javac の結果が変わらない・プログラムを書き換えて保存したつもりが保存されていなかった

プログラムを書き換えても、その後の javac コマンドや java コマンドの結果が変わらない場合、たいていは Emacs でファイル名を誤っているか、保存に失敗しています。

Emacs 上で書いたプログラムのファイル名が正しいかどうかを確認するには、まず Emacs のウィンドウの左下部分、太字でファイル名が書かれている部位を見てください。ここで書かれたファイル名が誤っている場合には、ファイル名を正してください。本章で説明した mv コマンドが有効でしょう。

Emacs 上で書いたプログラムが保存されているかどうかを確認するためには Emacs のウィンドウの左下部分、ファイル名の左を見てください。ここに** という印が見える場合には、プログラムは最新の状態で保存されていませんので、もう一度保存してください。

[質問] Emacs が書き込みを受け付けなくなった

Emacs の操作中に、何らかのキー操作をしてしまったために、プログラム本文の書き込みを受け付けなくなることがあります。このような場合には、Emacs のウィンドウの一番下の行を見てください。ここに何らかのメッセージがある場合には、これに応じた入力を要求されていて、それを入力するまでプログラム本文を書き込めない状態になっています。これを元の状態に戻す一番簡単な方法は、Emacs のウィンドウの一番下の行に、マウス操作でカーソルを表示させて、そこで Ctrl キーと g キーを同時に押すという操作を行うことです。Ctrl キーと g キーを同時に押すことで、直前の操作を中止することができるため、これによって Emacs はプログラムを書き込める状態に戻ります。

[質問] プログラムが反復処理し続けて終了しない

特に for 文や while 文を含むプログラムを書いたときに、プログラムに誤りがあると、実行時に反復処理を無限に繰り返して終了しないことがあります。このようなときは、ターミナルにて、Ctrl キーと c キーを同時に押す という操作を試みてください。この操作によって、実行中のプログラムが強制終了されます。

なお、この操作は自分で開発したプログラムに限らず、どのようなプログラムでも同様に使うことができます。

[質問] さっきターミナルに入力した命令を思い出したい

ターミナル上で、上向き、または下向きのカーソル（矢印）キーを押してください。1回押すごとに、前回の命令を表示します。何回か押すと、何回か前（または後）の命令を表示します。この状態でEnterキーを押すことで、表示された命令をもう一度実行できます。