

# プログラム作成の基礎

山本昌志\*

2006年4月19日

## 概要

UNIX を触ったこともなく、C 言語も全く知らない者が最初に学習する内容である。UNIX に触れてみることで、プログラム作成の最初の経験を行う。ここでは、UNIX のディレクトリーの操作と Hello World プログラムの作成方法が書かれている。

## 1 本日の授業内容とゴール

**授業内容** UNIX のディレクトリー構造を学習し、その操作の練習を行う。その後、実際にプログラムを作成して、実行させる。これらをとおして、プログラム作成の手順の概略を学ぶ。教科書 [1] の p.26～30 が範囲である。

**ゴール** 以下に示した内容が、本日の授業の目標である。

- UNIX のファイル構造が分かる。
- マウスを使って、ディレクトリーの作成ができる。
- ターミナルから、UNIX の以下のコマンドが使える。

pwd 自分がいるディレクトリーの表示  
ls ディレクトリー内のファイルとサブディレクトリーの表示  
cd ディレクトリーの移動

- emacs を使って、Hello World のプログラムがかける。
- Hello World のプログラムのコンパイルができる。
- Hello World のプログラムの実行ができる。

## 2 UNIX のディレクトリー構造と操作

### 2.1 UNIX を使う理由

これから、諸君は UNIX という OS<sup>1</sup> を使って、C 言語の学習を行う。もちろん、自宅や学校の Windows を使って、C 言語の学習を行っても良い。この授業で作成するプログラムは、Windows でも UNIX でも、あ

\*独立行政法人秋田工業高等専門学校電気工学科

<sup>1</sup>Operating System のこと。日本語では基本ソフトウェアと言われることもある。最もポピュラーな OS は、Windows である。

るいは Macintosh でも同じように動作する。講義で UNIX を使う理由は、将来、諸君が UNIX を使う可能性があるのである。とくに、情報系の仕事をするとき必ずそのスキルは必要で、学生の間に触れておく方がよい。

## 2.2 ディレクトリー構造

UNIX を使う場合、そのファイル構造を知っておく必要がある。プログラムやデータを整理して管理するためには、それを理解して、自在に操作しなくてはならない。これは、そんなに難しいものではない。Windows 使っている者であれば、Windows のファイル構造と操作は無意識に身につけている。UNIX も似たようなものである。

本授業で使う UNIX(Linux) のディレクトリー構造は、図 1 のようになっている。これは、木構造と呼ばれる階層構造で、図を反対にすれば、ちょうど木の幹と枝のようになっていることが分かる。木は根 (root) が必ずあり、そこから、枝が伸び、節で分かれ、末端には葉がある。「/」ルートディレクトリ (記号としては「スラッシュ」と発音) が根に相当し、その他のディレクトリが節に、ファイルが葉に相当する。ディレクトリ (Directory) は Windows のフォルダに相当するもので、他のファイルあるいはディレクトリを格納できる。ディレクトリを用いる事により、種類別にファイルを整理するとともに、階層別にそれらの管理ができるので便利である。

UNIX に限らず、Windows や Macintosh もファイルの管理はツリー構造となっている。ただし、UNIX ではファイルの管理は厳しいが、Windows や Macintosh はかなり甘い。前者は、複数のユーザーが使うことが前提としているため、ファイルの操作の権限を設けて、他人のファイルに触れなくていている。後者は、もともと個人の使用が前提なので、その辺はいい加減である。

これから、UNIX を使う場合、以下のことを理解しておく必要がある。

- UNIX のハードディスク<sup>2</sup>のファイル構造は、図 1 のようにツリー (木) 構造と呼ばれる階層構造になっている。それは、ファイルとディレクトリーから構成される。
- ハードディスクなどに記録されたデータのまとまりをファイルと言う。コンピュータが実行することができる命令の集合であるプログラムファイルと、コンピュータの利用者が作成した情報を記録しておくデータファイルがある。
- ファイルを分類・整理するための保管場所をディレクトリー<sup>3</sup>と言う。関連する複数のファイルをまとめて一つのディレクトリーに入れることにより、効率的に記憶装置を管理することができる。ディレクトリーの中にさらにディレクトリーを作成することもでき、階層構造によって細かい分類を表現することもできる。
- ディレクトリーには、以下のように表現されるものがある。
  - － 今、自分が居るディレクトリーを、カレントディレクトリーと言う。<sup>4</sup>カレントディレクトリーを明示したい場合は、1つのピリオド「。」で表す。

<sup>2</sup>実際はハードディスクに限らず、CD-ROM やフロッピーディスクも、このツリー構造に含まれる。

<sup>3</sup>Mac や windows ではフォルダと呼ぶ

<sup>4</sup>ワーキングディレクトリーと言うこともある。

- カレントディレクトリーの1つ上のディレクトリーを親ディレクトリーと言う。2つのピリオド「..」で、親ディレクトリーを表す。
- カレントディレクトリーの1つ下のディレクトリーをサブディレクトリーと言う。複数存在することが可能なので、それぞれの名前で表す。
- ユーザー各個人が使用(読み、書き、実行)を許されている最上位のディレクトリーをホームディレクトリーと言う。
- ログインしたときに入るディレクトリーをログインディレクトリーと言う。通常はホームディレクトリーと同じである。

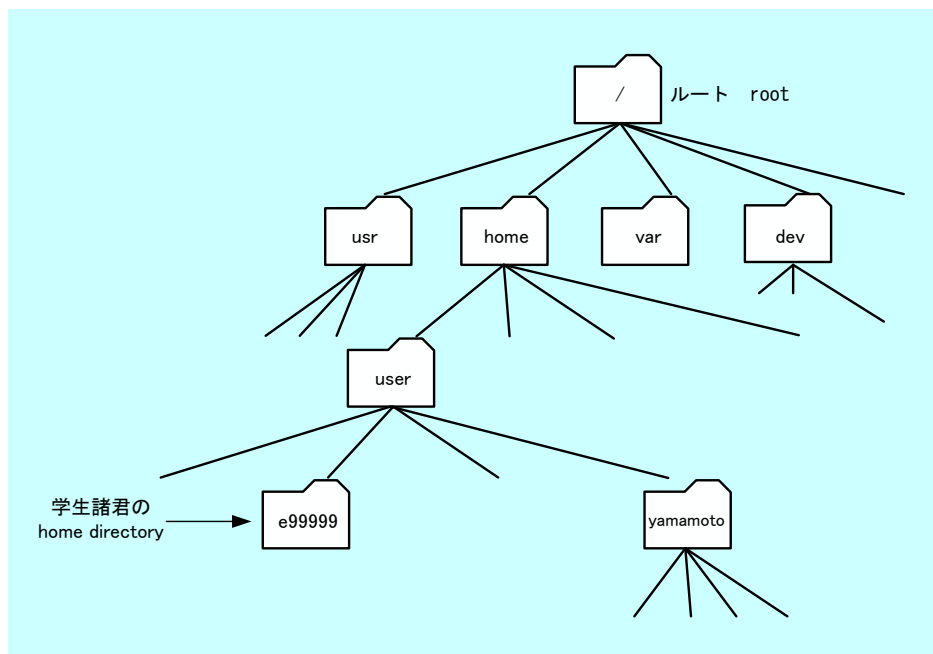


図 1: UNIX のファイル構造とそれを扱うコマンド

## 2.3 ディレクトリーの操作

デスクトップの上に、ユーザー名+ホームと書かれたアイコンがある。これが、諸君のホームディレクトリーを表している。ダブルクリックしてみると、その中身が見える。そして、次のマウス操作により、ディレクトリーの移動ができる。

- メニューバーにある上へと書かれた上三角マークをクリックすると、親ディレクトリーに移れる。
- ディレクトリーをダブルクリックするとサブディレクトリーへ移れる。あるいは、サブディレクトリーを右クリックして、[開く]を選択するとそこへ移れる。

- メニューバーにあるホームと書かれた家マークをクリックすると、ホームディレクトリーに移れる。

ホームディレクトリーよりも下の階層であれば、自由にディレクトリーの作成と削除ができる。また、ディレクトリーやファイルの移動もできる。マウスを使ってのファイルの操作は、以下のようにすれば良い。今後、プログラム作成を行うので、これらの操作には十分慣れておく必要がある。

- 右クリックして、新規作成のフォルダを選択する。そうすると、フォルダ名前を聞いてくるので、それをタイプする。これで、新たにサブディレクトリーができる。
- 削除したいディレクトリーあるいはファイルを右クリックする。そうして、ごみ箱へ移動を選ぶと、削除できる。
- ファイルやディレクトリーの移動は、ドラッグアンドドロップをつかう。

[練習 1] ホームディレクトリーに `hogehoge` と `fugafuga` というサブディレクトリーを作成せよ。

[練習 2] サブディレクトリー `hogehoge` に入り、その中に `hogehoge` というサブディレクトリーを作成せよ。

[練習 3] ここで作成したサブディレクトリーを全て削除せよ。

## 3 プログラムの作成と実行

### 3.1 一連の流れ

ディレクトリーの操作ができるようになったので、次はプログラムを作成と実行の練習を行う。まず、初心者が最初に作るものとして、最も有名な”Hello World”というプログラムをつくる。

#### 1. 作業ディレクトリーの作成

- デスクトップにある各人の `exxxxx` のホームをクリックして、ホームディレクトリーを開く。
- 右クリックして、新規作成のフォルダを選択する。そして、作業用ディレクトリー `work` を作る。

#### 2. ターミナルの起動と作業用ディレクトリーへの移動

- デスクトップのターミナルのアイコンをクリックして、ターミナルを立ち上げる。
- コマンド「`ls`」をタイプして、`[Enter]` キーを押す。すると、サブディレクトリーとファイルが表示される。
- コマンド「`cd work`」をタイプして、`[Enter]` キーを押す。すると、作業用ディレクトリーに移動できる。
- ここで、自分がいるディレクトリー (カレントディレクトリー) を見てみよう。コマンド「`pwd`」をタイプする。

UNIX では、ターミナルからコマンドを打ち込み操作する。これから、プログラムの作成と実行はこのターミナルをとおして行う。ターミナルのことをコンソール、あるいは端末ということがある。

### 3. エディターの起動

- 「emacs hello.c&」とタイプする。
- すると、emacs というソフトウェアが起動し、プログラムのソースを書くウィンドウが現れる。

emacs はテキストファイルを編集するソフトウェアである。テキストファイルが作成するエディターであれば、他のソフトウェアでもよい。emacs である必要はない。ただ、emacs は超有名なソフトウェアなので、慣れておいて、損はない。

### 4. プログラムの記述

- エディターのウィンドウに以下のプログラムを書く。

```
#include <stdio.h>

int main(){

    printf("Hello World !!\n");

    return 0;
}
```

### 5. プログラムの保管

- プログラムを書き終わったならば、[file] メニューの [Save(current buffer)] を選択する。あるいは、フロッピーディスクアイコンをクリックして、ソースファイルを保管する。
- ターミナル上で「ls」コマンドを打ち、ソースファイルが保管されていることを確認する。

保管された C 言語のテキストファイルを、ソースファイルと言う。カレントディレクトリーに hello.c というファイルがあり、それがソースファイルである。

### 6. コンパイル

- ターミナル上で「gcc -o aisatsu hello.c」と打ち込み、先ほど作成したソースファイルをコンパイルする。
- もし、コンパイルエラーが発生したら、ソースファイルを修正する。
- 実行ファイルができていないか「ls」コマンドで確認する。

C 言語の内容は、人間は理解できるが、コンピューターは理解不可能である。そこで、コンピューターが理解できる機械語に翻訳する作業が必要となってくる。その作業をコンパイルと言う。コンパイルによりできあがった機械語のファイルを実行ファイルという。コンパイルにより、ソースファイル (hello.c) から実行ファイル (aisatsu) を作成した。

## 7. 実行

- ターミナル上で「./aisatsu」と打ち込み、プログラムを実行させる。
- Hello World と表示されれば、プログラムは動作は完璧である。

これで、Hello World のプログラムは完成である。もし不幸にして、エラーが発生したならば、ソースファイルを直す。ちゃんと動作したならば、emacs を閉じてよい。

時間があれば、以下の練習を試してみよ。

[練習 1] 他のディレクトリーで同じプログラムを作成せよ。

[練習 2] Hello Akita !! と表示するプログラムを作成せよ。

## 参考文献

- [1] 内田智史監修, (株) システム計画研究所編. C 言語によるプログラミング 基礎編 第 2 版. (株) オーム社, 2006.