卒業論文

hikiutils を用いた
卒業論文作成

関西学院大学　理工学部　情報科学科

1234 西谷滋人

2017 年 3 月

指導教員　　西谷　滋人　教授

目次

目次

1 小嵐くんのプログラミングスキル向上プロジェクト

1.1 CUI

一般的に，コンピュータへの指図は，

- グラフィカルユーザインターフェース,Graphical User Interface, GUI

で行うことが普通です．しかし，その裏では，

- キャラクター (文字) ユーザインタフェース，character user interface, CUI

が実際の仕事をこなしています．

CUI は一般的なユーザには見えません．GUI に対してユーザが行った制御操作をうけとって CPU につなぐ役割を CUI は担っています．つまり，すごい機械の裏側ではたらくこびとの役なんです．

われわれプログラマーはこのこびとに直接指示を出します．

1.2 ディレクトリ

コンピュータには幾つものファイルがあります．このファイルを操作することによってプログラマが意図したようにコンピュータをふるまわせます．

ファイルの種類は，大雑把に，

- データファイル
  - ソース，設定，テキスト，図表など
- 実行ファイル
  - アプリケーション，フィルター，コマンドなど

に分けられます．

このたくさんあるファイルを管理しやすくするためにディレクトリと呼ばれる単位でくくるようになっています．

ディレクトリというのは電話帳とか，住所録に似ていることからつけられた名前です．

1.3 コマンド

CUI は別名 command line interface, CLI とも呼ばれます．コマンドというのは命令という意味です．英語の簡単な単語や文章からなります．

```
ls
```

と打ち込んでください．これは，リスト (list) の省略形を意図しており，ファイルのリストを出しなさいという命令です．

すべてのコマンドはその操作あるいは振る舞いを表現する言葉と結びついています．これは，プログラマが命令を思い出しやすいように，日常的な単語に結びつけるために意図されたためです．ディレクトリというのが電話帳とか，住所録に似ていることからつけられたのと同じ発想です．

1.4 カレントディレクトリー

ファイルのディレクトリはツリー構造になっています．木の幹から枝が分かれるようなイメージです．ツリーの一番根本のところをルートディレクトリと呼びます．木の根からつけられた名前です．

そこから順にディレクトリはツリーの階層にしたがって構成されています．プログラマが作業するには，その中のどこかに入ってファイルを操作したり，編集したりします．現在いる場所を表示するには，

```
pwd
```

と命令します．

```
print working directory
作業ディレクトリーをプリントしなさい
```

という意味です．

今いる場所はカレントディレクトリーと呼ばれます．もっかのあるいはいまのディレクトリという意味です．

ここにあるファイルやディレクトリを表示するのが，

```
ls
```

エルエスコマンドの目的です．

カレントディレクトリーにあたらしいディレクトリを作るには，

```
mkdir hogehoge
```

とします．hogehoge というのはなんでもいいけど適当な名前を打ち込む時によく使う表現です．

カレントディレクトリから hogehoge のディレクトリに移動するには，

```
cd hogehoge
```

とすれば動きます．ここで

```
pwd
```

と打てば，最後が hogehoge となったディレクトリ名が返されてくるはずです．

1.5 ファイル操作

なにか新しいファイルを作りましょう．

```
touch test.c
```

と打ち込んでください．これは test.c という名前のファイルに触りなさいという命令ですが，副作用として，その名前のファイルがない場合には，自動的にその名前がついた，中身が何もないファイルを作ります．

ファイルの名前は，

名前ドット C

というようにピリオドで区切られています．ピリオドより後ろにある部分を拡張子と呼びます．拡張子によってファイルの種類を区別しています．

C

というのは

C 言語で書かれたプログラムのソースコードだよ

ということを意図しています．

このファイルの名前を変更しましょう．

```
mv test.c first.c
```

と打ち込んでください．直訳すると

test.c を first.c に move しなさい，動かせ

ですが，その結果として名前が first.c に変更されます．

ファイルのコピーを作るのは，

```
cp first.c second.c
```

です．cp が copy の略だというのはわかっていただけますね．

ファイルを削除するのは

```
rm second.c
```

です．アールエムは remove の略です．

このとき

```
remove second.c?
```

と返ってくるときがあります．これは破壊的な操作をコンピュータに命令した時に本当にそれがプログラマの意図した操作かどうかをコンピュータが確認するために返してきた反応です．

```
y
```

yes の一文字目を打ち込んで return して，本気だぜというのをコンピュータに伝えてください．

2 編集

さて，では初めてのプログラムを作っていきましょう．

プログラムを書くには，それ専用のエディターを使います．プログラマは自分のお気に入りのエディターを持っています．私は emacs(イーマックス) です．これしかわからないので，これで説明していきます．

エディターを起動するには，

```
emacs first.c
```

と打ち込みます．

ここで次のテキストを打ち込んでください．

```
#include <stdio.h>

int main(void){
  printf("Hello world.\n");
}
```

これを c-x, c-s で保存します．そして c-z で emacs から抜けます．

この後，

```
gcc first.c
```

と打ち込んでください．これは，first.c を gcc でコンパイルしなさいという意味です．

```
ls
```

してみてください．新しいファイルとして a.out というのが出来上がっているはずです．

これが初めて作った実行ファイルです．これを実行させてみましょう．

```
./a.out
```

と打ち込んでください．

```
bob% ./a.out
Hello world.
```

と返ってくるはずです．

2.1 バグ取り

打ち込んだテキストに間違いがあるとうまく動きません．間違いのことをバグと言います．虫がコードを食い荒らしているイメージです．その時は emacs へ戻って編集していく必要があります．

例えば，

```
bob% cat first_bug.c
include 'stdio.h'

int main(void){
  printf("Hello world.")
```

というのを打ち込んだとします．幾つもバグ，虫がコードを食い荒らしていてうまくコンパイルが通りません．

```
 bob% gcc first_bug.c
first_bug.c:1:1: error: unknown type name 'include'
include 'stdio.h'
^
first_bug.c:1:9: error: expected identifier or '('
include 'stdio.h'
        ^
2 errors generated.
```

というのが典型的なエラーの報告です．通常はエラーが確認された一番上の行が表示されます．じっくりと聞いてどこからのエラーの報告かを理解してから emacs へ戻ってその行を編集してください．

プログラミング言語ではかっこやセミコロン，カンマやシャープ記号に意味をもたせています．バグ取りをしながら，C 言語で書かれたプログラムの典型的な構造を読み取ってください．

順に説明します．

1. printf という関数を使いますが，それを使うために必要なヘッダーファイルをあらかじめ読み込む必要があります．
2. 読み込みは include という英語があります．
3. そういう命令だよということで，#include とします．
4. そのファイルは鉤括弧¡¿でくくられるか，ダブルコーテーションでくくられる必要があります．
5. C 言語ではかならず main 関数というのを作る必要があります．
6. C 言語では関数が返す型を明示する必要があります．
7. int というのは main という関数が返す型が整数だよという意味です．
8. 関数は引数をとります．丸括弧のなかに引数を入れます．
9. main ですが今は何も引数を取らないということを明示するために void, 「空」を取らせています．
10. C 言語ではプログラム構造の単位を領域分けするために波カッコでくくります．
11. printf は format したテキストを打ち出せと命令する関数です．
12. printf の命令の内容は丸括弧 () の中に入れます．
13. printf ではダブルコーテーションで囲まれた中にフォーマットを入れています．
14. ここでは，”hello world.” と打出させています．
15. そのあと，改行をいれます．これは
    n エンエヌと発音します．
16. printf に関する命令が終わったことを明示するためにセミコロン; を入れます．
17. このセミコロンは文章のおわり，文末を意味するために，ターミネータと呼びます

これで先ほどの hello world プログラムのソースの解説は終わりです．

```
mv first.c hello_world.c
```

としてプログラムの振る舞いがわかる名前に付け替えておきましょう．

出来上がった a.out も hello_world とするといいでしょう．アンダーバーはファイル名で使えない空白の代わりによく使います．単語の区切りや幾つもの意味を組み合わせるための工夫です．

この後，c 言語の学習は変数，型，制御，loop, 関数と進めていきます．記述がくどいかもしれませんが，ゆるしてください．

来週書く内容を忘れないためのメモ，file mode, permission，省略，補間，ドット，ドットドット