

ISC

Theoretical Science Group
理論科学グループ

ここに何か出力するんですよ
こんな風にね

部報 273?号
— C 言語分科会 —

目次

前準備	2
-1. 情報棟でのコンパイルと実行の仕方.....	2

前準備

-1. 情報棟でのコンパイルと実行の仕方

ターミナルの起動

ターミナルは、Windows ではコマンドプロンプトなどに相当する、Mac のコンソール画面です。Mac は UNIX の CUI を引き継いでいるので、似たようなコマンドを入力できます。デフォルトのシェルは bash です。

まずターミナルを起動してください。ターミナルのアイコンは、「黒い画面の中に>_」と表示されているものです。ドック(アイコン)の一覧の中にはない場合は、Finder を立ち上げて、横のアプリケーションから探してください。

最重要コマンド

コンソール画面上でまず覚えるべきコマンドは以下の3つのコマンドです。

pwd	今いるディレクトリ (カレントディレクトリ) を表示します。
ls	今いるフォルダ (ディレクトリ) の下にある、ディレクトリ・フォルダ一覧を表示します。
cd	change directory の略。その名の通り、今いるディレクトリから、指定されたディレクトリへ移動します。

cd の使い方 :

- 現在いるディレクトリ (カレントディレクトリといいます) の下に「hoge」というディレクトリがあってそこに行きたいとき

```
cd hoge
```

- 現在いる上のディレクトリ (親ディレクトリといいます) に行くとき

```
cd ..
```

「..」は上のディレクトリを意味します。「../」と入力しても構いません。

- 一番上のディレクトリ (ルートといいます、Windows のドライブに相当) に移動するとき

```
cd /
```

「/」は一番上のディレクトリを意味します。

- 最初のディレクトリ (ホームディレクトリといいます) に移動するとき

```
cd
```

重要コマンド

exit	終了したりログアウトしたりします。
cat	ファイルの中身を表示します。
touch	空のファイルを作ります。
mkdir	ディレクトリを作ります。
cp	ファイルをコピーします。
mv	ファイルを移動したり名前を変えたりします。
rm	ファイルを削除します。
man	マニュアルを表示します。
nkf	文字コードを変換します。

cat の使い方

カレントディレクトリの下に hoge.txt と fuga.tex があるとします。

- hoge.txt の中身を表示するには

```
cat hoge.txt
```

- hoge.txt と fuga.tex の中身を連続して表示するには

```
cat hoge.txt fuga.tex
```

hoge.txt と fuga.tex の間には何も挿入されませんので注意してください。

先頭と最後だけを表示させるコマンドもあります。

- hoge.txt の先頭 10 行を表示するには

```
head -10 hoge.txt
```

- hoge.txt の最後 10 行を表示するには

```
tail -10 hoge.txt
```

-1. 情報棟でのコンパイルと実行の仕方

touch の使い方

hoge.txt が作りたいなら

```
touch hoge.txt
```

mkdir の使い方

ディレクトリ「hoge」が作りたいなら

```
mkdir hoge
```

複数のディレクトリ名を指定することで、同時に作ることが出来ます。

cp の使い方

カレントディレクトリの下に hoge.txt, fuga.tex とディレクトリ piyo があるとします。

- hoge.txt を同じディレクトリに hogehoge.txt という名前でコピー

```
cp hoge.txt hogehoge.txt
```

この場合は複数のファイルは指定できません。

- hoge.txt を piyo に同じ名前でコピー

```
cp hoge.txt piyo
```

ディレクトリの名前は既に存在しているものしか指定できません。

- hoge.txt と fuga.tex を piyo に同じ名前でコピー

```
cp hoge.txt fuga.tex piyo
```

ディレクトリにコピーする場合のみ複数指定できます。

- piyo を hogepiyo というディレクトリにコピー

```
cp -r piyo hogepiyo
```

hogepiyo というディレクトリが存在しない場合、hogepiyo というファイルが作られ、piyo の中身がそこにコピーされます。

hogepiyo というディレクトリが存在する場合、hogepiyo の下に piyo ごとコピーされます。このときは複数指定可。

mv の使い方

基本的には cp と同じです。カレントディレクトリの下に hoge.txt, fuga.tex とディレクトリ piyo があるとします。

- hoge.txt を同じディレクトリに hogehoge.txt という名前に変更

```
mv hoge.txt hogehoge.txt
```

- hoge.txt を piyo に同じ名前で移動

```
mv hoge.txt piyo
```

ディレクトリの名前は既に存在しているものしか指定できません。

- hoge.txt と fuga.tex を piyo に同じ名前で移動

```
mv hoge.txt fuga.tex piyo
```

ディレクトリに移動する場合のみ複数指定できます。

- ディレクトリ piyo を hogepiyo というディレクトリに名前の変更又は移動

```
mv piyo hogepiyo
```

-r オプションが要りません。

hogepiyo というディレクトリが存在しない場合、hogepiyo というディレクトリに名前が変更されます。

hogepiyo というディレクトリが存在する場合、hogepiyo の下に piyo ごと移動されます。複数指定可能。

rm の使い方

カレントディレクトリの下に hoge.txt, fuga.tex とディレクトリ piyo があるとします。

- hoge.txt を削除

```
rm hoge.txt
```

複数指定可です。

- 中身が空な piyo を削除

```
rmdir piyo
```

何気に別のコマンド。複数指定可です。

- piyo を丸ごと削除

```
rm -rf piyo
```

複数指定可です。

-1. 情報棟でのコンパイルと実行の仕方

nkf の使い方

日本語でプログラムをするとき、ちょっと便利なのがこのコマンドです。Windows のヘルプから見ると FUJITSU LTD. の Network Kanji Filter の略のように見えるが。カレントディレクトリの下に hoge.txt があるとします。

hoge.txt を EUC に変換するには

```
nkf -e hoge.txt
```

hoge.txt を UTF-8 に変換するには

```
nkf -w hoge.txt
```

しかし、これでは変換結果が表示されるだけで hoge.txt には変化はないので、下に解説する方法でこれを hoge2.txt に保存します。

```
nkf -w hoge.txt > hoge2.txt
```

なお、

```
nkf -w hoge.txt > hoge.txt
```

というように、自分自身を書き換えるような命令は激しくお勧めしません。情報棟の環境だと、内容が消去されたはず。

重要な文法

- > コマンドの出力をファイルに保存します。
- >> コマンドの出力をファイルに追加保存します。
- | コマンドの出力を別のコマンドに食わせるときに使います。

>の使い方

コマンドを hoge.txt に出力するには

```
CommandName > hoge.txt
```

もし hoge.txt がない場合、新しく作られ、ある場合は、その内容が消去されて新しく書き込まれます。

>>の使い方

コマンドを hoge.txt に出力するには

```
CommandName >> hoge.txt
```

もし hoge.txt がない場合、新しく作られ、ある場合は、その内容が保留されて追加で書き込まれます。

|の使い方

CommandName1 の出力を CommandName2 に食わせるには

```
CommandName1 | CommandName2
```

コマンドによっては、別のコマンドの出力をそのまま入力として受け取ることが出来るものもあります。例えば、

```
head -10 hoge.euc.txt | nkf -w
```

で hoge.euc.txt の先頭 10 行を UTF-8 に変換して表示することが出来ます。

Mac の重要なコマンド

Mac 特有のコマンドらしいです。

```
open
```

 Mac の設定にしたがって、ファイルを指定のアプリケーションで開きます。

open の使い方

hoge.txt を開くには

```
open hoge.txt
```

まあ、他のコマンドは興味があれば調べて覚えておくと言う程度で。

参考文献

UNIX のコマンドは「とほほのWWW入門」URL:<http://tohoho.wakusei.ne.jp/>を参照。

open に関しては、ハイパーワークブックを参照。

13 TSG 部報 No.273?

ソースファイルを作る

とりあえずターミナルを立ち上げてください。

まずこれから書くプログラムをおくフォルダを作ります。

例えばこんな感じ

```
mkdir program
cd program
```

そこに「helloworld.c」というファイルを作って下さい。

```
touch helloworld.c
```

今回はこれを Xcode で開きます。

```
open helloworld.c
```

好きな方は emacs などを使ってもらっても構いません。

```
emacs helloworld.c
```

Xcode では、Windows と同じような操作環境が使えます。また、保存も「コマンド +S」で出来ます。

なお、Mac での画面切り替えは「コマンド +Tab」です。

これでキーボードでほぼ全ての操作が出来ます。

ここで、helloworld.c に以下の C のコードを書き込んで保存してください。

```
#include <stdio.h>
int main(int argc, char *argv []) {
    printf("Hello _world!\n");
    return 0;
}
```

なお、この後も、コードの変更などを行ったら、保存は忘れないようにしてください。

コンパイルと実行

ここで書いたプログラムは、人間が読める形で書いてあるもので、機械は直接は読めません。機械がプログラムを使って、プログラムを逐一読んで実行するタイプの言語（インタプリタ型言語）もありますが、C はインタプリタ言語ではないので、機会が直接読める機械語に変換する必要があります。この、「機械語に変換する」作業を「コンパイル」と言い、コンパイルをするためのソフトウェアを「コンパイラ」と呼びます。

今回は gcc というコンパイラを使いたいと思います。

さっき書いたプログラムを保存したら、以下のようにコマンドを打ってください。

```
gcc helloworld.c
```

このようにコマンドを打つと、a.out というファイルが出力されます。a.out という名前が気に入らない人は、

```
gcc -o b.out helloworld.c
```

のように、gcc と helloworld.c の間に、「-o 出力ファイル名」と打ってください。

今回は画面に機械語が表示されるわけではないので、> は使えません。

このとき出力されたファイルが、さっきの C のコードが機械語になったプログラムですので、これを実行させます。Mac 環境上で実行させるには、「./出力ファイル名」と入力してください。例えば次のような感じです。

```
./a.out
```

これを入力すると、すぐに

```
helloworld
```

と表示されます。

編集後記

- ★ test
- ★ text
- ★ texture

理論科学グループ 部報 第 273?号

2008年5月1日 発行

発行者 ぶちょ

編集者 編集長の名前

発行所 理論科学グループ

〒153-0041 東京都目黒区駒場 3-8-1

東京大学教養学部内学生会館 305

Telephone: 03-5454-4343

©Theoretical Science Group, University of Tokyo, 2008.

All rights received.

Printed in Japan.

理論科学グループ部報 第 273?号

－ C 言語分科会 －

2008 年 5 月 1 日

THEORETICAL SCIENCE GROUP