



初心者のための C言語講座

#1: ウォーミング・アップ

#1-0) はじめに

新入生の皆さん、こんにちは。このC言語講座の担当をする、50代のZAKIと申します。これから宜しくお願いしますね。

本講座は、大学に入ってからCプログラミングを始めるような方を対象にしています。「プログラムならバリバリ書けるぜ!!」という人は、おそらく物足りない内容になるかと思しますので、その点ご了承くださいませ……m(_ _)m

また、基幹理工学部1年の必修には「Cプログラミング 入門」「Cプログラミング」という講義があります。内容はそれに準拠して作ってあるつもりですので、同学部の方々には、是非とも単位をとる手助けになればと思います。

それでは、これから一緒に頑張っていきましょう!!

#1-1) まずは『おまじない』

さて、早速プログラムを書いてみましょうーと言いたいところですが、おそらく何をすれば良いか分からないと思います。

ということで、まずは右に記したProgram1.1を写してみてください。何も考えずに写してしまいましょう。

写し終わったら、保存しましょう。名前は「start.c」とでもしておきましょうか。(実際は「～.c」なら何でもいいんですけど)

```
#include <stdio.h>

int main(void){
    return 0;
}
```

▲ Program1.1 start.c

#1-2) コンパイル

さて、せっかく書いてくれたプログラムですが、そのまま実行することはできません。C言語のままではコンピュータが読めないからです。

したがって、プログラムを実行する前には「**コンパイル**」という作業が必須になります。

「**コンパイル**」とは、皆さんが書いてくれたC言語によるテキストファイルを、コンピュータが読める機械語で書かれたバイナリファイルに直す作業のことです。機械語は0と1で構成されているので、人が直接書くのはあまり現実的ではありません。だから、人でも理解できるようなC言語などのプログラミング言語が必要になるのです。

プログラミング言語

- ・人が読める
- ・実行はできない

```
#include <stdio.h>
```

```
int main(void){
```

```
.....
```

コンパイル

```
01101110.....
```

機械語

- ・コンピュータが読める
- ・実行できる

▲ Figure1.1 コンパイルのイメージ

#1-2) コンパイル

使っている開発ソフトなどによっては、コンピュータが勝手にコンパイルもやってくれてしまう場合もありますが、一応コンパイルの方法も記しておきます。

コマンドプロンプトを開いて、以下のコマンドを打ち込んで、Enterを押してください。

gcc C言語のテキストファイル名

Enterを押すと、コンピュータがコンパイルを開始します。Figure1.2の例のように、しばらくしてエラーが何も表示されなければ、コンパイル成功です。

```
user@vm101:~$ gcc start.c
```



```
user@vm101:~$ gcc start.c  
user@vm101:~$
```

▲ Figure1.2 コンパイル作業

#1-2) コンパイル

逆に、Figure1.3の例のようにエラーが表示された場合は、コンパイルに失敗しています。

たいていの場合、書いたプログラムの方に問題があるので、自分の書いたプログラムが間違っていないか再度確認してみましょう。

また、コンパイルしようとするC言語のファイルが、現在自分のいるディレクトリと違う場所に存在する場合、そもそもコンパイルができません。(例えば、start.cを「Desktop」に保存したのに、自分が「Download」にいる場合など)自分が現在どこのディレクトリにいるのかは、コマンドプロンプト上で確認することができます。

```
user@vm101:~$ gcc start.c
start.c:1:20: fatal error: studio.h: No such file or
directory
#include <studio.h>
                ^
compilation terminated.
user@vm101:~$
```

▲ Figure1.3 コンパイル失敗例

```
.....
user@vm101:~/Desktop/Mis.w$
.....
```

▲ Figure1.4 ディレクトリ情報

(この場合は、「Desktop」中の「Mis.w」というディレクトリにいることになる)

#1-3) 実行

コンパイルが成功したら、いよいよ実行です。実行方法は開発環境によってかなり異なるのですが、今回は基幹理工学部の必修科目「Cプログラミング 入門」に合わせて、Unix環境での実行方法を教えます。

コマンドプロンプト上に、以下のコマンドを打ち込んで、Enterを押してください。

```
./a.out
```

これで晴れて、start.cが実行することができました。
やったね!!

```
user@vm101:~$ gcc start.c  
user@vm101:~$ ./a.out
```

▲ Figure1.5 実行作業

#1-4) 『おまじない』 を実行すると……

start.cを実行すると、おそらくFigure1.6のようになります。「……あれ？ 何も実行してなくね」と思ったアナタ。正解です。実は、このstart.cは何もしないプログラムなんです。

C言語では `int main(void){}` (「main関数」と言います)の中に書かれてあることを上から順に実行していきます。return 0はプログラム終了のことなので、start.cは実行直後に終了するという何の役にも立たないプログラムだった訳です。

```
user@vm101:~$ gcc start.c
user@vm101:~$ ./a.out
user@vm101:~$
```

▲ Figure1.6 実行後のコマンドプロンプト

```
#include <stdio.h>

int main(void){
    return 0;
}
```

▲ Program1.1 start.c

Column) バイナリファイル名の指定

先ほどは、プログラムを実行する際に、「./a.out」とコマンドプロンプト上に打ち込みました。なぜ、このコマンドでプログラムが実行したのかというと、コンパイルによって作成されたバイナリファイル名が「a.out」という名前になっているからです。

実は、講義中で取り扱ったコンパイルのコマンドだと、バイナリファイルの名前の指定をしていないので、自動的に「a.out」という名前になるのです。要するに、コンパイルをするたびに、「a.out」は上書きされてしまうこととなります。

これでは、ある特定のバイナリファイルを残しておきたい時に不便ですね。

```
#include <stdio.h>
```

```
int main(void){
```

```
.....
```



gcc ~.c

```
01101110.....
```



a.out

▲ Figure1.7 バイナリファイル名未指定のとき

Column) バイナリファイル名の指定

では、バイナリファイルの指定方法を教えます。コンパイル時のコマンドを以下のように変えるだけです。

`gcc -o バイナリファイル名 テキストファイル名`

例えば、start.cをコンパイルするときに、バイナリファイル名を「result」としたいときは、以下のようなコマンドになります。

`gcc -o result start.c`

```
user@vm101:~$ gcc -o result start.c
```

▲ Figure1.8 コンパイル作業

Column) バイナリファイル名の指定

当然、実行のコマンドも「./a.out」ではいけませんね。
以下のように覚え直しましょう。

./バイナリファイル名

先ほどのようにバイナリファイル名を「result」と指定したときは、以下のようなコマンドになります。

./result

今回の講義は以上です。お疲れさまでした。資料の最後に例題をつけておきましたので、適宜解いてみてください。

```
user@vm101:~$ gcc -o result start.c
user@vm101:~$ ./result
```

▲ Figure1.9 実行作業

Question1-1) コンパイル/実行

右のProgram1.2のようにstart2.cを作成した。

プログラム開発はUnix環境下で行っているものとして、以下の問いに答えよ。

- (1) コンパイル時に、コマンドプロンプトに打ち込むコマンドを答えよ。(バイナリファイル名を指定する必要はない)
- (2) 実行時に、コマンドプロンプトに打ち込むコマンドを答えよ。

```
#include <stdio.h>

int main(void){
    return 0;
}
```

▲ Program1.2 start2.c

Answer1-1) コンパイル/実行

(1) バイナリファイル名の指定が不要なので、

```
gcc start2.c
```

(2) バイナリファイル名を指定せずにコンパイルしたので、自動的にバイナリファイル名は「a.out」になる。したがって実行コマンドは、

```
./a.out
```

Question1-2) バイナリファイル名の指定方法

右のProgram1.3のようにstart3.cを作成した。

プログラム開発はUnix環境下で行っているものとして、以下の問いに答えよ。

- (1) コンパイル時に、コマンドプロンプトに打ち込むコマンドを答えよ。(「answer」をバイナリファイル名にすること)
- (2) 実行時に、コマンドプロンプトに打ち込むコマンドを答えよ。

```
#include <stdio.h>

int main(void){
    return 0;
}
```

▲ Program1.3 start3.c

Answer1-2) バイナリファイル名の指定方法

(1) バイナリファイル名を「answer」にしなければならないので、

```
gcc -o answer start3.c
```

(2) バイナリファイル名が「answer」なので、

```
./answer
```