

C 言語

Computer は 0 と 1 で動きます。0,1 を人間が書いていたらとても大変なので、人間が読みやすい形式で書いて、これをプログラムが翻訳するという形式が取られます。この翻訳前のものがソースコードで、これを翻訳(コンパイル)することで機械語に変換し、実行できるようにします。

この翻訳前(ソースコード)の人間にわかりやすい形式の書き方はプログラミング言語と呼ばれ、この形式のうちの一つが C 言語です。実際他にもいろいろな言語があります。(C++, D, Golang, Fortran, COBOL, など)

C 言語はこの中でも高級言語であり、人が使う言葉に近い表現でプログラムを記述するプログラミング言語であるが、ハードウェアに密着した処理も容易に記述できる、というものです。

また、C++などの言語に影響を与えているので C 言語を学ぶことでそれらの言語を学ぶのがより楽になります。

Hello World

先ずはどの言語でもやる Hello World を端末に表示させて見ましょう。

以下の文章をエディタで書き込んでください。詳細は後ほど解説します。

```
#include <stdio.h>

int main(void) {
    printf("Hello World\n");
    return 0;
}
```

実行すると

Hello World

と表示されるでしょう。

プリプロセッサ(preprocessor)

先ほどの Hello World の解説を行いつつ、C 言語の説明を始めていきます。

1 行目の

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

はプリプロセッサというものが解釈します。pre-processor, つまりコンパイラーの前に処理するものです。

例えば, ある日ものすごい時間をかけてすばらしいプログラムの部品を書いたとします。これを使えば一発であれこれできる, めんどくさいことは全部この中で面倒見てくれるみたいなものです。さて別のプログラムを書いているときに, それが使いたくなったらどうするのでしょうか? いちいちコピー&ペーストしますか?

そんな面倒なことをしないように, include 命令があります。include はあるファイルの中身を読み込み, そこに展開します。これによってコピーしてくるような手間をせずプログラムを使うことができます。

さらにすばらしいことに, C 言語には歴史があって, ものすごい賢いプログラマが, いろんな問題が起きないように考え抜かれて作られた便利な部品をこの include 命令で使うことができます。

この場合, C の標準ライブラリ(このような便利な関数を別の人が使えるようにまとめたものをライブラリといいます。プログラムの図書館ですね)である stdio を読み込んでいます。Standard Input / Outputの略です。この瞬間から, 自分のプログラムにstdioの便利関数を余すところなく使えるようになります。

文(statement)

プログラムの命令の区切り区切りを文(statement)といいます。ステートメントっていう人多いですね。

例えば,

```
printf("Test");
```

や

```
printf("%d¥n", 10 * 10);
```

というふうなものです。文は必ずセミコロン";"で終了します。

制御文字(escape sequence)

'¥n'はエスケープシーケンスと呼ばれるものです。制御文字とも言われます。これは, 印字できない文字を表すやり方です。

例えば, '¥n'は改行文字です。これが印字(print)されると改行されます。試しに¥nを取り除いてみればなるほどという感じになるか試してみると良いかもしれません。

stdio: printf

print

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

で, `stdio` の関数である `printf` に文字列を渡して呼び出しています.

関数とは, 引数(parameter)を与えて, 値が帰ってくる(返り値)ものです.

`printf` は引数に渡された文字を, 端末に印字(print)します.

文字列の豆知識

意外と知られていないのですが, C 言語では連続した文字列リテラルは連結されて一つのリテラルになります. リテラルって何だよって思った方,

```
"Hello World\n"
```

これが文字列リテラルです. これが面白くて,

```
printf("%s", "Hello" " " "World" "\n");
```

は先ほどとまったく同じ意味になります. カンマで区切ってないので文字列リテラルが連続しており, すべて連結されているのです. このため, 上のものは,

```
printf("%s", "Hello World\n");
```

とまったく同じ意味です.

これを利用すると, きれいに書けることがあります. ここで急に2年の情報工学の問題をとってきます.

次のような模様を表示するプログラムを作成せよ.

```
##          ##
# ##      ## #
# #####   #
# #####   #
# ##      ## #
##          ##
```

をこう書くときれいな感じですよ.

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

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

```
    printf("%s", "##          ##\n")
```

```

        "# ## ## #¥n"
        "# #### #¥n"
        "# #### #¥n"
        "# ## ## #¥n"
        "### ##¥n");
    return 0;
}

```

その他の format

とりあえず, %d, %x, %c を紹介しておきます.

```

#include <stdio.h>

int main(void){
    printf("%d : %x : %s : %c¥n",
        100, 255, "test", 'A');
    return 0;
}

```

たくさんの format 指定子(%なんちゃら)を書いておくと, 引数の値が順番に渡されて埋め込まれていきます. ここでは%dに 100, %xに 255, %sに"test", %cに'A'ですね.

d は decimal の d, x は hex の x, s は string の s, c は character の c です.

注意として, "test"は double quote なのに対し, 'A'は single quote なのに注意してください. "なんちゃら"は文字列, 'A'は文字を表します. C 言語では文字と文字列は別物です.

コメント(comment)

コメントというのは, プログラム中で無視される部分を指します. 例えばちょっと変更したり, 後で見たらなんでそうやったのか忘れてそうな難しいところに, comment を書いておくと読みやすくなったりします. 人によっては「コメントを書くのは邪道. 読めばわかるような美しいソースコードを書ければコメントなんてものは…なんていう方もいますが, まあ概ね comment は有用です.

書き方としては一行コメントと, 複数行コメントが C 言語には存在します.

```
#include <stdio.h>

int main(void) {
    // これが一行コメント
    printf("Hello World¥n");

    /*
     *   これが複数行コメント
     */
    printf("Hello World¥n");
    return 0;
}
```

一行コメントは//が始まってから行末まで、複数行コメントは/*から*/までがコメントになります。

<http://www.soft-forum.net/2010/programming/> により詳しく載っています。