

C言語の基礎知識

1. プログラム言語としてのC

・コンピュータとプログラム

ソフトウェア プログラム-人の思考 : プログラム言語
 ハードウェア 機械-コンピュータ : マシン語
 操作 (いろいろな処理)

・C言語の特徴

Pascal COBOL LISP C
 Fortran RPG Prolog
 (数値計算向き) (事務処理向き) (人口知能向き) (万能)

変換
 マシン語

(1) マシンレベルに近い記述能力 マシン(コンピュータ)が良く見える
 他言語ではマシンは見えにくい

- (2) 高速な実行速度
- (3) シンプルな言語仕様
- (4) モジュール化に対応

モジュール化 : プログラムを部品として作成し、プログラム部品を組み合わせることによって大きなプログラムを作っていく方法。

2. C言語の全体像

プログラム言語 文法 She loves you
 S V C
 処理系 翻訳機
 「彼女はあなたが好き」

	C言語	他言語
文法	変数 実行の流れ プログラムの部品化	変数 実行の流れ (プログラムの部品化) 画面出力 キーボード入力 ファイル入出力 etc.
処理系	マシン語への変換 (プログラム部品 画面出力 キーボード入力 ファイル入出力 etc.)	マシン語への変換 (プログラム部品)

3. C言語の歴史

- ・K&R 準拠のC言語
- ・ANSI 準拠のC言語

米国の AT&T ベル研究所で、オペレーティングシステム・UNIX のために開発されたプログラム言語

1970 年 BCPL 言語

B 言語 Ken Thompson

C 言語 Dennis M.Ritchie

C の言語仕様をまとめた「The C Programming Language」共立出版、石田晴久訳

Brian W.Kernighan

Dennis M.Ritchie K&R : C 言語の規格書

1989 年 ANSI:American National Standards Institute 米国国内規格協会

標準規格 ANSI-C (K&R の言語仕様を拡張したもの)

そして、C 言語にオブジェクト指向の考え方を取り入れた C++言語 Bjarme Stroustrup

オブジェクト指向：情報に処理を加えるという考え方ではなく、情報自身が自分で行動する
かのようにプログラムを組み立てること。

4.C 言語習得の心得

(1) コンピュータの仕組みについて理解すること

プログラムが実行される裏にあるコンピュータの仕組みを意識しながら理解する

(2) C 言語はプログラムのチェックを、プログラム実行時に行わない

- ・プログラムの暴走(でたらめな実行)
- ・無限ループ(プログラムの実行を停止できない)
- ・core dump(実行時のメモリの状態を core ファイルに書き出す)

5.C プログラムを実行させるまで

ソースファイル

コンパイラ

コンパイルエラーでソースファイル修正

リンカ

ソースファイルを実行可能ファイルに変換する

実行可能ファイル

プログラムの実行

CC : ラッパー、コンパイラ&リンカ

バグ(bug) : プログラムミスのこと。バグを取り除く作業 : デバッグ

6.C 言語の特徴

a. 小文字をベースにしている

b. 簡潔な表現ができる

Pascal	C
begin end	{ }
a: integer;	int a;
if a=10 then b:=1	b=(a==10)? 1:2;
else b:=2;	
a:=a+1;	a++;
a:=a+b;	a+=b;

c. 演算子が豊富

算術演算子の他に

インクリメント・デクリメント、代入、コンマ、
条件、ポインタ、ビット演算子 etc.

d. ポインタも用いる

ポインタ : 変数の格納されたアドレス

e. データ構造が豊富

char, int, float, short, unsigned, long, double
構造体、共用体

- f. プログラムは関数の集合
- g. 構造化制御文が備わっている
 - if, for, while, do, switch, break, continue
- h. プリプロセッサ機能
 - Pascal の const 文に相当するもの
- i. 入出力機能がない
 - Pascal の write 文に相当するものがない。標準関数 (printf 等) として装備
- j. 特殊文字が扱える
 - ¥n : 改行 ¥t : タブ等エスケープ文字

7.C 言語のルール

最小のプログラム main(){}

- a. プログラムは関数で構成される
 - main(), printf() : プログラム部品
- b. 関数は{で始まり、}で終わる
 - ブロック文複数を{}で囲ったもの
- c. プログラムには main 関数が 1 つ必要である
 - abc をディスプレイに出力する main(){printf("abc")}
- d. 文には ; (セミコロン)が必要である
 - Pascal におけるセミコロンはマルチステートメントの区切り
- e. C には行の概念がない
- f. printf は関数である
- g. 改行はプログラマが指示する ¥n
- h. コメントは /* と */ で囲む
- i. 変数は宣言してから使用する
- j. 変数の値を出力するには書式指定が必要である
 - printf (=Print with format)
 - printf("ans=%d", a);
 - 変数 a の値を 10 進数(decimal)で出力する
 - 書式制御文字 %c : 1 文字出力 %d : 10 進数 %x : 16 進数
 - %f : 浮動小数点 %s : 文字列
- k. C の制御文は、ただ 1 つの文を制御する


```
int i, sum, multi;
sum=0; multi=1;
for (i=1; i<=5; i++){
    sum=sum+i;
    multi=multi*i;
}
printf("%d,%d¥n", sum, multi);
```
- l. 非制御文が 2 つ以上のときはブロック文を用いる

8. プログラムの構成

C プログラム	----- 外部宣言
	+++ main 関数
	+++ 一般関数
外部宣言	#include <stdio.h>
	#define ASMNO 1
main(引数)	main()
引数宣言	{
{	{

```

        内部変数宣言      int a;
        文                a=123;
                          sttmsg(ASMNO);
                          printf("a=%d\n",a);
    }                    }
    関数名(引数)        attmsg(n)
    引数宣言            int n;
    {                    {
        内部変数宣言      int b;
        文                b=456;
                          printf("b=%d\n",b);
                          printf("AsmNo=%d\n",n);
    }                    }

```

9. 演算子

a. 算術演算子 * , / , % , + , -

b. 関係演算子 > , >= , < , <= , == , !=

c. 論理演算子

! 否定 NOT

&& 論理積 AND

|| 論理和 OR

d. インクリメント、デクリメント演算子

++ 1を加算する

-- 1を減算する

a=a+1; => ++a; (前置型)

a++; (後置型)

a=++b; => b=b+1; a=b;

a=b++; => a=b; b=b+1;

e. 代入演算子

記号	例	一般記法
=	a=b	a=b
+=	a+=b	a=a+b
-=	a-=b	a=a-b
=	a=b	a=a*b
/=	a/=b	a=a/b
%=	a%=b	a=a%b