

C言語講座Vol.1

2009年5月15日 CISC

コンピュータの構造について

- 「演算装置、制御装置、記憶装置、入出力装置」によって構成される。
- 記憶装置は、プログラム上、特別意識せずに利用可能な「主記憶装置(メモリ)」と、「補助記憶装置(HDDやCDなど)」に分けられる。

コンピュータ内部のデータ

- 内部でのデータ保持には、電気が流れているか、流れていないかの、2通りしかできない。
=> 内部のデータは全て2進数
- メモリは、内部に多くの箱を持ち、それぞれ番号が振ってある(アドレス)。これを指定すると、データを読み書きできるが、通常は意識する必要はない。

			アドレス	データ
A =>	(0100 0001) ₂	文字も2進数のコード	0x0000	12
B =>	(0100 0010) ₂		0x0002	2.1
C =>	(0100 0011) ₂		0x0004	C
		メモリのイメージ	0x0006	C
			0x0008	S

2進数の基礎知識1

- 2進数1桁をビット(bit)、8桁をバイト(byte)という。
- 2進数を書き表すときは、通常4桁ごとに区切りの空白を入れて書く。 $\Rightarrow (0100\ 1001)_2$
- 4桁毎に、16進数1桁に対応する。
- 10進数への変換は、2のn乗の重みを付ける。
 $(0100\ 1001)_2 = 2^0 + 2^3 + 2^6 = 1 + 8 + 64 = 73$
- 2進数への変換は、2で割って行き、それぞれの余りを記録。これを後ろから読むと2進数になる。

2進数の基礎知識2

- 2進数の負数は、補数表現という形で表される。

通常 : $(0100\ 1001)_2$

1の補数 : $(1011\ 0110)_2$

2の補数 : $(1011\ 0111)_2$

- 2進への変換方法は、全ビット反転で1の補数、それに1加えると2の補数となる。同様の操作をもう一度すると、10進数に戻る。
- 桁数は固定する必要がある。

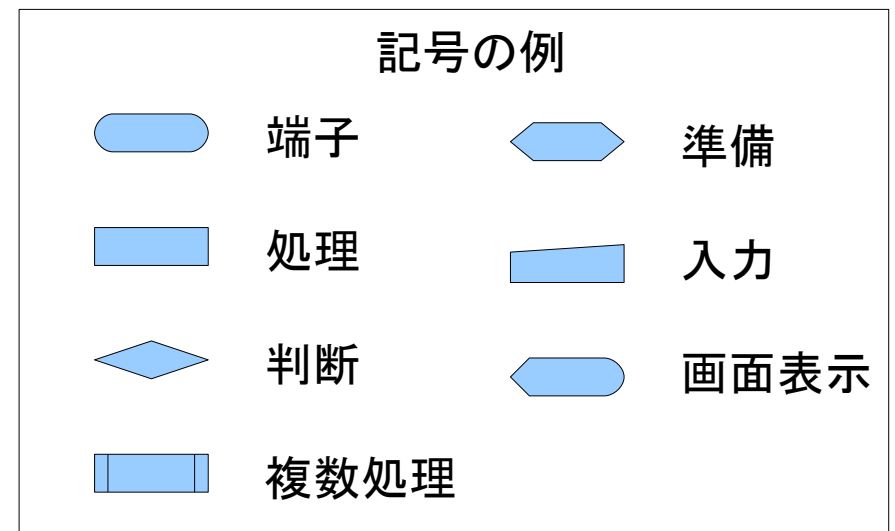
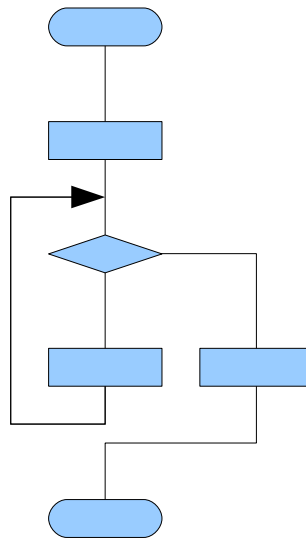
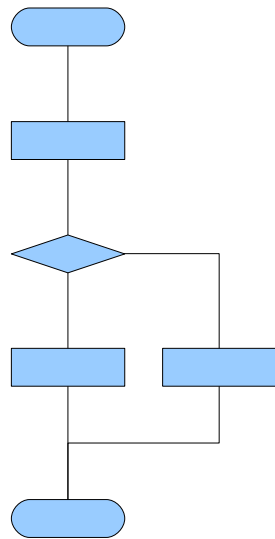
2進数の基礎知識3

- 小数点付の値は、固定小数点数または浮動小数点数により表される。
- 固定小数点数の場合は、任意の位置に小数点があると考え、マイナスの重みをつければよい。
- 浮動小数点数の場合は、 1.52×10^5 のように表す。勿論、実際には2進数で表現されている。よって、 $(0111)_2 \times 2^5$ のようになる。
- いずれにせよ、誤差が発生することが多い。
=> 丸め誤差など

フローチャートについて

- 処理の流れを書き表す。流れ図とも言う。
- プログラムを記述する前に、どのように処理したら解決できるかを記述するために利用する。コードを見るよりフローチャートを見たほうが、一般的には処理の構造を把握しやすい。

- 例



プログラミング言語

- プロセッサが理解できるのは、2進数で書かれた、「機械語」という命令のみで、それぞれの命令は極めて単純なものだけ。
- 人間が読めるよう、機械語に一对一で単語を割り当てたものを「アセンブラ」という。
- 表示、条件分岐、演算など、人間が使いやすい単位まで機械語の処理をまとめ、扱いやすくしたものを高級言語という。
- コンパイラ(ソースコードを一度に機械語へ変換)とインタプリタ(実行時に、一行ごとに解釈する)の2種類がある。前者はC言語が代表であり、後者はBASICが有名。

C言語とは

- 1972年 アメリカはAT&T研究所で誕生
Unixのために作られた？らしい
- 非常に高速である
- 現在使われている各種プログラミング言語に対し、
C言語の作法などが受け継がれている
- 初心者が初めて学ぶ言語に最適であり、またより
深く掘り下げることにも出来る言語
- 移植性の高い仕様ゆえ、グラフィックスとは疎遠。

VisualStudio

- Microsoftの提供する統合開発環境
- 実際の現場でもよく利用されている
- 以前は結構高価な製品だったが、今は無料で手に入れることも可能 (Express Edition)
- 情報画像学科の学生は上位バージョンを学科から借りることも可能 (4月で終了...) だった

VisualStudioの使い方1

- まず、プロジェクトを作らないと話にならない。
[ファイル]->[新規作成]->[プロジェクト]からウィザードを開く
- 初級講座で扱うのは、コンソールアプリケーション
[VisualC++]から[Win32]、そして[Win32コンソールアプリケーション]を選択する
- 名前がないプロジェクト
名前をつける必要がある。プロジェクト名に適宜入力。ファイルの保存場所を確認して[ok]を押す
- 空のプロジェクト
諸事情により、最初は空のプロジェクトで。
チェックボックスにチェックしてから続ける。

VisualStudioの使い方2

- ソースファイルの追加
プログラムを記述するファイルを追加しよう。
[プロジェクト]->[新しい項目]
[コード]->[C++ファイル(.cpp)]
名前を入力(半角英数で)して[追加]をクリック！
ソリューションエクスプローラーのソースファイルに追加されたことを確認
- さあ、次はプログラムの記述だ！

Hello World

- 実際にプログラムを動かそう
下記のプログラムを入力

```
#include <stdio.h>

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

四則演算

- 下記のプログラムを入力

```
#include <stdio.h>

int main(void)
{
    int a, b, wa, sa, seki, syo;

    a=10;
    b=3;

    wa=a+b;
    sa=a-b;
    seki=a*b;
    syo=a/b;

    printf("%d+%d=%d\n", a, b, wa);
    printf("%d-%d=%d\n", a, b, sa);
    printf("%d*%d=%d\n", a, b, seki);
    printf("%d/%d=%d\n", a, b, syo);

    return 0;
}
```

小数

- 下記のプログラムを入力

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

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

```
    int a, b, iSyo;  
    float c, fSyo;
```

```
    a=10;  
    b=3;  
    c=3.0;
```

```
    iSyo=a/b;  
    fSyo=a/c;
```

```
    printf("違いを確認！\n");  
    printf("a/b=%d\n", iSyo);  
    printf("a/c=%f\n", fSyo);
```

```
    return 0;
```

```
}
```

入出力基本

- 下記のプログラムを入力

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

```
int main(void)
```

```
{  
    int year, month, day;
```

```
    printf("誕生年を入力してください\n=>");
```

```
    scanf("%d", &year);
```

```
    printf("誕生月を入力してください\n=>");
```

```
    scanf("%d", &month);
```

```
    printf("誕生日を入力してください\n=>");
```

```
    scanf("%d", &day);
```

```
    printf("あなたの誕生年月日は\n%d年%d月%d日\nですね？\n", year, month, day);
```

```
    return 0;
```

```
}
```


問題1

1. 次の2進数を10進数へ変換せよ

$(1100\ 0001)_2$ $(0100\ 0110)_2$ $(0111\ 1000)_2$

2. 知っているプログラミング言語の名前を書け

3. 身近な処理を二つ以上、フローチャートで表せ
思いつかない場合は、次を用いる。

- ・ 昼食を食べに行く
- ・ 実験レポートを書く

問題2

- 値(a,b)を入力して、次の値を計算し出力するプログラムを作れ。それぞれの計算式において、a,bはカッコ内の意味として用い、5種類を同時に表示し、可能であれば、小数点付の値とする。
 - 三角形の面積(a:底辺 b:高さ)
 - 円の面積(a:半径)
 - 電圧(a:抵抗 b:電流)
 - 抵抗(a:電流 b:電圧)
 - 電流(a:電圧 b:抵抗)

解答

```
#include <stdio.h>

int main(void)
{
    int a, b;
    float san, en, v, r, i;

    printf("aの値¥n=>");
    scanf("%d", &a);
    printf("bの値¥n=>");
    scanf("%d", &b);

    // 小数点以下を残した式にするために、
    // *. *のような記述を加えている。
    // 別の手段もあるが、それは後日。
    san=a*b/2.0;
    en=a*a*3.14;
    v=b*a;
    r=(b*1.0)/a;
    i=(a*1.0)/b;

    printf("三角形の面積%f¥n円の面積%f¥n", san, en);
    printf("電圧%f¥n抵抗%f¥n電流%f¥n", v, r, i);

    return 0;
}
```