

2006年10月5日(木) 実施

コンピュータのハードウェアとプログラム

機械 (ハードウェア) としてのコンピュータの特徴

現在、普通に使われているコンピュータはデジタルコンピュータであり、次のような特徴を備えている。

1. 電子デバイス (主に半導体集積回路を用いた装置) により、データ処理を行う。
2. 2 値論理に基づいてデータを扱う。(電圧の高低, 電荷や窪みの有無, 磁化の向き等)
3. 主記憶装置 (メインメモリ) に一時的に読み込んだプログラムを中央処理装置 (CPU) で解釈・実行することにより、データ処理を行う。 【プログラム記憶方式】

プログラムとは (機械語, アセンブラ, 高級言語)

プログラムとは、コンピュータに処理の手順を指令するものである。コンピュータが動作するための命令の種類とその書式は、CPU の設計段階で決定される命令セット (Instruction Set Architecture) に基づく。この命令セットを用いて作成されたプログラムを機械語プログラムと呼び、命令及びデータは 2 進数で表される。コンピュータが動作可能なプログラムは、機械語で記述されたもののみである。なお、命令セットは CPU ごとに異なるので、ある CPU 用の機械語プログラムは異なる設計の CPU 用には流用できない。

また、2 進数でプログラムを作成していくのは困難なので、命令に名前付けをした、アセンブラ (アセンブリ言語) が用いられる。アセンブラプログラムは機械語プログラムと 1 対 1 に対応するので、やはり CPU の設計に依存する。

ここに、Intel 社の Pentium Pro プロセッサという CPU の命令セットマニュアルより、ADC というアセンブラの命令に関する説明の一部を抜粋する。Opcode 欄は機械語、Instruction 欄はアセンブラによる記述であり、Description 欄はそれらの説明である。

“Intel Architecture Software Developer’s Manual Volume 2: Instruction Set Reference”

(<http://developer.intel.com/design/pentium/manuals/24319101.pdf>) より抜粋：

ADC—Add with Carry

Opcode	Instruction	Description
14 ib	ADC AL,imm8	Add with carry imm8 to AL
15 iw	ADC AX,imm16	Add with carry imm16 to AX
15 id	ADC EAX,imm32	Add with carry imm32 to EAX
	

- **ib, iw, id**—A 1-byte (ib), 2-byte (iw), or 4-byte (id) immediate operand to the instruction that follows the opcode, ModR/M bytes or scale-indexing bytes. The opcode determines if the operand is a signed value. All words and doublewords are given with the low-order byte first
- **imm8**—An immediate byte value. The imm8 symbol is a signed number between -128

and +127 inclusive. For instructions in which imm8 is combined with a word or doubleword operand, the immediate value is sign-extended to form a word or doubleword. The upper byte of the word is filled with the topmost bit of the immediate value.

- **imm16**—An immediate word value used for instructions whose operand-size attribute is 16 bits. This is a number between -32,768 and +32,767 inclusive.
- **imm32**—An immediate doubleword value used for instructions whose operand-size attribute is 32 bits. It allows the use of a number between +2,147,483,647 and -2,147,483,648 inclusive.

なお、AL、AX、EAX はレジスタの名称である。

*レジスタ CPU 内部にある、演算や実行状態の保持に用いる記憶素子。動作が極めて高速だが容量が小さい。

通常のプログラミング（プログラム作成）には、CPU の設計に依存しない、**高級言語**と呼ばれるプログラム言語を用いることが多い。高級言語の多くは命令を英単語やその組み合わせまたはその省略形で表す人工的な言語である。高級言語で記述されたプログラムの各行は機械語に 1 対 1 には対応していないので、翻訳（コンパイル）・編集（リンク）という作業を通じて、コンピュータが実行可能な機械語プログラムに変換される。この作業は大変複雑なため、それぞれコンパイラ、リンカというプログラムを通じて行われる。この場合、元の高級言語プログラムは**ソースプログラム**と呼ばれる。（ソースプログラムの各行を解釈・実行していくインタプリタと呼ばれるプログラムも存在する）

この授業では、高級言語としては古くから存在していた、**手続き型言語**と呼ばれるプログラム言語の中で広く用いられている **C 言語**のプログラミングについて初歩から学ぶことにする。

C 言語の概要

C 言語の特徴

C 言語は UNIX という OS を記述する際に開発されたプログラム言語であり、プログラミングに際しては、次のような特徴を備えている。

1. プログラム中の処理のひとまとまり（モジュール）を**関数**と呼ぶ。
2. データの種別ごとにメモリ上に占めるサイズが固定されている。⇒ データを格納する変数は全て宣言を行い、データ型を明示しなければならない。
3. 関数間のデータの受け渡しはデータの値をコピーして渡す**値渡し**であり、メモリ上のデータの格納場所のアドレスを渡す**アドレス渡し**を行うためには、**ポインタ**を用いなければならない。
4. 標準入力（キーボード）及び標準出力（画面）を基本とする。
5. 基礎的な処理はライブラリ関数として用意されており、その利用に際しては、ヘッダファイルをプログラムに取り込む必要がある。
6. プログラムは 1 バイトコードの文字（画面表示では「半角の英数字」と呼ばれる文字）で記述する。但し、プログラム中の文字列データに関しては、コンパイラが 2 バイトコード（画面表示では「全角文字」と呼ばれる文字）に対応していれば、その利用も可能である。

7. プログラム構造の見易さのために、ブロック毎に字下げ（インデント）を行うことが望ましいが、各行を何桁目から記述するかということに関しては決まりはない。 【フリーフォーマット】
8. 文末にはセミコロン;を用いる。
9. ブロックは中括弧{ }で囲む。

例題 1

次のプログラムを入力し、翻訳・編集して実行形式のファイルを作成し、実行せよ。ここで、ソースプログラム名は prog1-1.c とする。

```
/* prog1-1.c */
#include <stdio.h>

int main(void)
{
    printf("Hello world!\n");

    return 0;
}
```

【解説】

1. /* から */ までは注釈であり、プログラムの翻訳の際には無視され、動作に影響しない。注釈が何行に亘ってもよい。
2. #include は翻訳の際にヘッダファイルをプログラムに取り込むためのプリプロセッサ命令である。
3. stdio.h は standard input/output（標準入出力）の機能を利用するためのヘッダファイル名である。
4. main はプログラム本体のモジュールで、main 関数と呼ばれる。
5. int は integer（整数）のデータ型名で、ここでは main 関数の戻り値を規定している。
6. void は「空」を意味し、ここでは、プログラム実行時に外部から main 関数へのデータ受け渡しが無いことを表す。
7. printf は標準出力すなわち画面に文字列を表示するためのライブラリ関数である。
8. \n は改行を行う制御コードを表す。
9. return はそれに続く式により、関数の戻り値を表す。ここでは、この位置までプログラムが実行された場合、正常終了をシステムに知らせるコード 0 を戻す。

演習 1

例題 1 のプログラムの文字列データを「今日は」に変更し、翻訳・編集して実行形式のファイルを作成し、実行することで 2 バイトコードが利用可能であることを確認せよ。ここで、ソースプログラム名は ex1-1.c とする。