

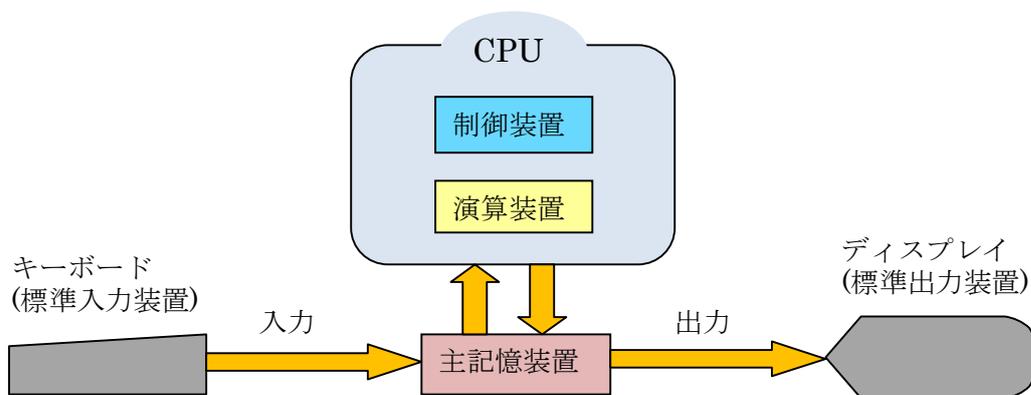
2013 年 4 月 11 日 (木) 実施

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】



注) ディスプレイには、ビデオ RAM (VRAM) に書き込まれたデータが表示される

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順序を指令するもの**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。

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。この内容は今後の授業で順に解説していく。

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** (ディスプレイ) 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 (ディスプレイ表示では「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 (ディスプレイ表示では「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の決まりはない。【**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 (標準出力への文字列の表示)

次のソースプログラムをテキストエディタで入力して、prog1-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し、実行せよ。なお、翻訳・編集及び実行に関しては別の教材を用意してあるので、そちらを参照すること。

```
/* 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、サービス創造学部の〇〇です。」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。