

## 1 コンピュータプログラミング

### 1.1 コンピュータプログラミング

プログラム 計算機に実行させる処理手順をある一定の記述言語で具体的に記した表現。

C のような手続き型プログラミング言語では、「所定の目的を達成するために、何ををどんな手順で実行する必要があるか ( アルゴリズム )」を、記述する必要がある。

プログラミング 計算機のプログラムを作成すること。

### 1.2 機械語とアセンブリ言語

機械 (マシン) 語 コンピュータが直接理解できる「ことば」。0 と 1 の並びでできている。コンピュータ (CPU) の種類によって異なる。

```
1300042774
1400593419
1200274027
```

アセンブリ言語 プログラミング言語の一種。アセンブリ言語の命令と機械語の命令はほぼ一対一の対応をもつ。

```
LOAD BASEPAY
ADD OVERPAY
STORE GROSSPAY
```

アセンブラ アセンブリ言語から機械語への翻訳プログラム

高級言語 (高水準言語) 人間が使う言語に近い要素に基づくプログラミング言語の総称

```
grossPay = basePay + overTimePay
```

コンパイラ 高級言語から機械語への翻訳プログラム

### 1.3 高水準言語

**FORTRAN (1957)** 数値計算, 技術計算

**LISP (1959)** 記号処理, 人工知能

**COBOL (1960)** 事務処理

**BASIC (1965)** 汎用, 初心者向け

**Pascal (1971)** プログラミング教育用

**C (1971)** システム記述, 汎用

1989 年 ANSI (American National Standards Institute; 米国標準規格協会) による C 言語の標準化

Prolog (1973) 論理プログラミング, 人工知能, 自然言語の構文解析

Ada (1983) 信頼性や保守の容易性に優れる。システムプログラム向き

C++ (1985) オブジェクト指向プログラミングが容易になるように C を拡張

この他にもたくさんのプログラミング言語があります。

## 2 C プログラムの作成から実行まで

以下は UNIX 系 OS における一例であるが, Windows のコマンドプロンプト等から行う場合でも基本的な流れは同じである。

1. エディタを使ってソースコード (source code) を書き, ファイルに保存する。このファイルをソースファイル (source file) という。C のソースファイルでは, ファイル名の終りに `.c` を含める。

エディタに GNU Emacs を使う場合の実行例は次のとおり。 *filename* は適当なソースファイル名に置き換えること。

```
emacs [-nw] [filename] &
```

2. ソースファイルをコンパイル (compile) して実行可能ファイル (executable file) を作る<sup>1</sup>。コンパイルを行うプログラム (コマンド) をコンパイラ (compiler) という。この授業で使うコンピュータには GNU C compiler がインストールされており, 起動のためのコマンド名は `gcc` または `cc` である。

```
gcc filename
```

コンパイルの途中でエラーメッセージが出たら, 1. に戻ってソースファイルを修正する。

エラーが無ければ, `a.out` という実行可能ファイルができあがる。

3. プログラムを実行し, 実行結果を確認する。

```
./a.out
```

プログラム実行時にエラーが発生したり, 実行結果に不具合があれば, 1. に戻ってソースファイルを修正し, 2. のコンパイルも行う。

なお, プログラムに潜む欠陥をバグ (bug) といい, バグを取り除く (プログラムを修正する) 作業をデバッグ (debug) という。

---

<sup>1</sup>正確には, コンパイルの後にリンク (link) が行われて実行可能ファイルが出来上がるのであるが, ここでの「コンパイル」という言葉には, リンクのステップも含むこととする。