

第1章 はじめに

1.1 プログラミング言語としての C

1.1.1 プログラミング言語とは

プログラミング言語とは、プログラムを書くための形式、つまりハードウェアに指示を与えるための形式である。日本語・英語などの日常言語と同じように、言語の種類毎に定まった文法（約束ごと）に従って記述する。

プログラミング言語には多くの種類があるが、もっとも大まかな分類は、_____と_____にわけけるものである。

機械語

_____ともいう。_____が直接理解出来る形式である。人間にとっては、そのままでは使いにくい。また、当然 CPU の種類に依存する。

高級言語

人間にとって使いやすいように設計された言語である。ただし、CPU は直接理解できないので何らかの方法で機械語に翻訳する必要がある。翻訳の方法には _____（一括翻訳方式）と _____（逐次翻訳方式）がある。前者の方式の場合、プログラムを実行する前に、翻訳（_____）という作業を行なう必要がある。C 言語は通常このコンパイラ方式を用いる。

1.1.2 C 言語の歴史

C 言語は 1970 年代に _____ というオペレーティングシステムを記述するために AT&T Bell 研究所の D. M. Ritchie によって開発された。

系統的には Algol という言語（Fortran とならぶ最古の高級言語）の血を引き、Pascal とは兄弟か従兄弟のような関係である。

1978 年に Ritchie と B. W. Kernighan による “The C Programming Language”（通称 K&R）という本が出版され、一般に広まった。

1988 年 ANSI により標準化された。この時に安全性の強化のため、いくつかの改訂がなされた。また、1980 年代前半に AT&T の B. Straustrap により C のオブジェクト指向拡張として _____ が開発された。C++ は C に対して上位 互換 である。つまり、原則として C のために書かれたプログラムは C++ のコンパイラでもコンパイルすることができる。

問 1.1.1 C 言語以外のプログラム言語について、名前を挙げよ。

.....

1.2 C 言語プログラムを実行させるまで

このプリントでは、C 言語処理系として Borland 社のフリーの C コンパイラ bcc32 を用いる。
 最初にまず、画面に「こんにちは」や「Hello!」などの文字列を表示するだけのプログラムを作成する。

プログラムの作成は次のような手順で行なう。

1. エディタ（_____・_____ など）でソースプログラム（ソースファイル）を作成する。C のソースファイルの拡張子には _____ をつける。
2. C コンパイラでオブジェクトプログラム（実行可能ファイル）を作成する（コンパイルする）。Windows 環境の場合、できた実行可能ファイルには _____ という拡張子がつく。
3. プログラムを実行する。

コンパイル時や実行時に文法上の間違いや _____（不具合）が見つかった場合（見つかる方が普通）は、ソースプログラムの作成に戻る。

ソースプログラムの作成

次のような内容のファイルをエディタで作成し、hello.c という名前で保存する。

ファイル hello.c

```
#include <stdio.h>

main()
{
    printf("こんにちは、皆さん!¥n");
    return 0;
}
```

注意: stdio は standard I/O（標準入出力）のことである。

注意: ¥（円記号）は、機種や環境によって \（バックスラッシュ）と表示されることもある。

printf(...); は、画面に文字列などを出力するための命令である。「"」（ダブルクォート）と「"」に囲まれた部分は _____ を表す。上の例では「こんにちは、皆さん」という文字列が画面に出力される。なお、このような命令文の最後には必ず「;」（セミコロン）がつくことに注意する。

プログラムのコンパイル

まず、「スタート」—「すべてプログラム」—「アクセサリ」—「コマンドプロンプト」で、「コマンドプロンプト」という名前のウインドウを開く。

このままでは、作成したソースファイルをコンパイラが見つけれられないので、ソースファイルを作成したフォルダに移動する。フォルダを移動するには `__` というコマンドを用いる。

```
cd path
```

パスの中に空白が含まれている場合は、パス全体をダブルクォーテーション「`"`」で囲む必要がある。

例えば「`C:¥foo`」というフォルダに移動したい場合は、

「`cd "¥foo"`」と入力する。(C:は不要である。)

コンパイルには、`_____` というコマンドを用いる。

```
bcc32 source-file
```

例えば `hello.c` をコンパイルするためには `bcc32 hello.c` と入力する。他の C コンパイラ (Visual C++ や LSI-C など) を用いる場合には、コマンド名が異なる。

プログラムに間違いがなければ、`bcc32` の場合、`hello.exe` というファイルが出来ているはずである。つまり、`bcc32` の場合は、実行可能ファイルは、ソースファイルの名前の拡張子 (.以降の部分) を `exe` に変えたものになる。

`_____` というコマンドで、現在のフォルダにどのようなファイルがあるか確かめることができる。次のような出力がされれば、コンパイルは成功している。

```
C:¥foo>dir
```

```
ドライブ C: のボリュームラベルはありません。
ボリュームシリアル番号は XXXX-XXXX です
```

```
C:¥foo のディレクトリ
```

```
YYYY/MM/DD HH:mm <DIR>      .
YYYY/MM/DD HH:mm <DIR>      ..
YYYY/MM/DD HH:mm              72 hello.c
YYYY/MM/DD HH:mm             52,224 hello.exe
YYYY/MM/DD HH:mm              478 hello.obj
YYYY/MM/DD HH:mm             393,216 hello.tds
      4 個のファイル              445,990 バイト
      2 個のディレクトリ    X,XXX,XXX,XXX バイトの空き領域
```

この出力結果は `hello.c` というファイルの他に `hello.obj` と `hello.exe` というファイルが出来ていることを示している。`obj` ファイルはコンパイルの中間生成物である。

一方、`bcc32` を実行したときに “Error...” とか “Warning...” のようなメッセージが出ている場合は、ソースプログラム中に間違いがあって、コンパイルに失敗している可能性がある。もう一度ファイルの編集からやり直す。(Warning (警告) だけの場合は、コンパイルは一応出来ている。)

プログラムの実行

hello.exe を実行するには、その hello.exe のあるディレクトリ（フォルダ）で、単に hello と入力すれば良い。（.exe は不要である。）

```
C:\My Documents\My Programs>hello
こんにちは、皆さん!
```

問 1.2.1 プログラムの中の文字列を別の文字列に変更してコンパイルし実行せよ。

問 1.2.2 文字列中の「\n」はどのような意味があるか？この部分を取り除いてコンパイル・実行し結果を比較せよ。（\n の他に \ から始まる特殊な意味を持つ文字（エスケープシーケンスともいう）には、\t（水平タブ）、\a（警報）、\b（バックスペース）、\'（' という文字）、\"（" という文字）、\\（\ という文字）などがある。）

.....
.....

キーワード:

一般:

機械語、高級言語、コンパイラ、インタプリタ、UNIX、エディタ、MS-DOS プロンプト、cd、dir、C 言語に関するもの:

K&R、\n、