

## PC の環境設定/printf

今回からソフトゼミ A が始まります。今日はソフト班の活動において基本となる「C 言語」を自分のパソコンで使うために、パソコンの環境設定を行います。また、文章を表示させる命令である「printf」の解説もします。

### ➤ コンパイラの取得

---

C 言語をパソコンで使用するには、コンパイラというものがが必要です。コンパイラ取得の手順は以下の通りです。

① 次のサイトにアクセス ↓

<http://www.embarcadero.com/jp/products/cbuilder/free-compiler>

② 「C++コンパイラのダウンロード」の中の「こちらのページ」をクリック。

③ そうすると「ダウンロード登録フォーム」に飛ぶので、そこに個人の名前やメールアドレス等を登録。

④ 登録するとダウンロードページに行くのでそこでダウンロードする。

また、登録したメールアドレスにダウンロードしたファイルの解凍に必要なパスワードが送られてくるのでチェックしておく。

⑤ ダウンロードしたファイル(freecommadlinetoos.zip)を解凍し、その中の「freecommadlinetoos.exe」を実行する。

実行すると、「C:\¥borland¥bcc55」というフォルダができると思うので、フォルダが作られたか確認する。

ここまでできればコンパイラの取得は成功です。

### ➤ 拡張子の表示

---

「拡張子」とは、ファイルを見たときに、そのファイルがどのようなファイルかを判断するための文字列です。先ほどのファイル(freecommadlinetoos.exe)を例に挙げると、「.exe」の部分が拡張子に当たります。自分が作ったファイルがコンパイルできているかどうかを確認するのにも使えるのですが、Windows のデフォルト設定では見えなくなっているため、見えるようにします。

#### ■ Windows XP

「マイコンピュータ」→「ツール」→「フォルダオプション」→「[表示]タブ」→「詳細設定」の項目の下から 3 番目にある[登録されている拡張子は表示しない]

い]のチェックを外す→「すべてのフォルダに適用」→「OK」→「OK」

#### ■ Windows Vista,7

「コンピューター」(デスクトップにない場合はスタートメニューから、もしくは Windows キーと「E」キーを同時押し)→メニューバーの「ツール」(表示されていない場合は ALT キーを押すと表示されます)→「フォルダオプション」→「[表示]タグ」→「詳細設定」の項目の下から 3 番目にある[登録されている拡張子は表示しない]のチェックを外す→「すべてのフォルダに適用」→「OK」→「OK」

#### ■ Windows 8

デスクトップを開く→「コンピューター」(デスクトップにない場合は Windows キーと「E」キーを同時押し)→開いた窓上側「表示」リボンをクリック→右側の「表示/非表示」の欄の中の「ファイル名拡張子」にチェックを入れる

### ➤ 環境設定ファイル・環境変数の設定

---

メモ帳などのテキストエディタを開き、C:\Borland\bcc55\Bin のところ  
(=コンパイラ導入の時にできたやつ)に

```
-I"C:\borland\bcc55\Include"  
-L"C:\borland\bcc55\Lib"
```

という内容の「bcc32.cfg」というファイルを作ってください。(拡張子に注意)

また、同じ C:\Borland\bcc55\Bin のところに、

```
-L"C:\borland\bcc55\Lib"
```

という内容の「ilink32.cfg」というファイルも作ってください。

次に、環境変数をいじります。これをミスるといろいろと大変なことになるので慎重にやってください。使用している Windows のバージョンによって出し方が異なるので注意してください。

#### ■ Windows XP 以前

デスクトップの「マイコンピュータ」を右クリック→「プロパティ」→「詳細設定」→「環境変数」

#### ■ Windows Vista

「スタート」→「コントロールパネル」→「クラシック表示」→「システム」

→画面左側「タスク」内「システムの詳細設定(A)」→「続行」→「詳細設定」  
→「環境変数」

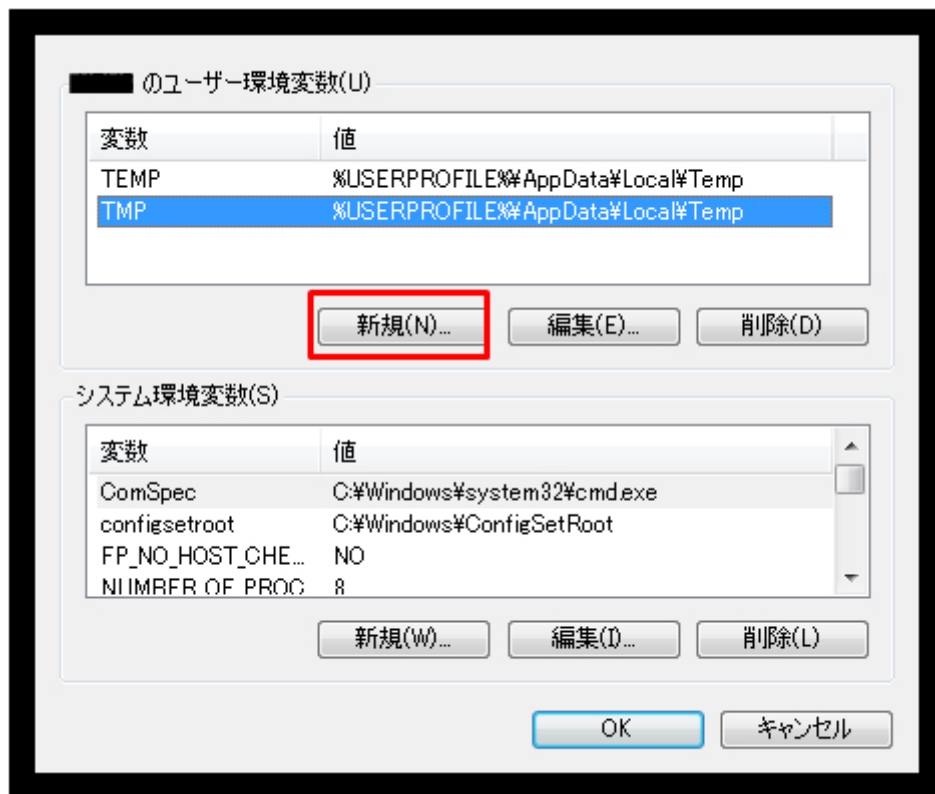
#### ■ Windows 7

「スタート」→「コントロールパネル」→「システムとセキュリティ」→  
「システム」→画面左側「タスク」内「システムの詳細設定(A)」→「詳細設定」  
→「環境変数」

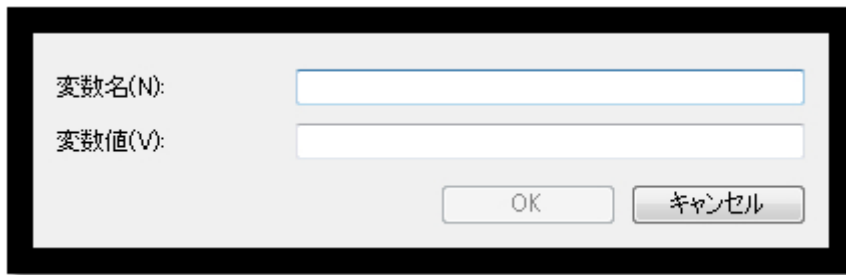
#### ■ Windows 8

「スタート」を右クリック→「システム」→画面左側「システムの詳細設定」  
→「詳細設定」→「環境変数」

すると、下の画面が出てきます。



赤い四角で囲ってある「新規(N)」をクリックしてください。  
すると、次ページにあるような画面が出てきます。



「変数名(N)」に

PATH

「変数値(V)」に

C:\borland\bcc55\Bin

と入力して「OK」をクリックしましょう。

※すでに PATH が 1 つ以上登録されている場合は「;(セミコロン)」で区切って書いてください。

そのあと環境変数の丸の上のところに[PATH]があるかどうか、正しく打たれているかどうか確認してから「OK」を押せば設定完了です。何か間違いがあったら「キャンセル」をクリックすれば元に戻るのもう一回やり直しましょう。

## ➤ コマンドプロンプト

---

今までの設定が正常にできているかどうかを確認します。

「スタート」→「すべてのプログラム」→「アクセサリ」→「コマンドプロンプト」をクリックする、または Windows キーと「R」キーを同時押しして出てきたウィンドウに「cmd」と入力すると、コマンドプロンプトが開きます。(今後よく使うので、素早く出せる校舎の出し方を覚えておくと便利です。)コマンドプロンプト上で「bcc32」と打って「Enter」を押します。

すると、次のような画面が出てくるとと思います。こうなれば設定は無事成功です。これ以外が出てきたら、どこかで失敗しているので今までの操作を確認しながらやり直します。

```
C:\windows\system32\cmd.exe
C:\Users\tanno_kazuho>bcc32
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
Syntax is: BCC32 [ options ] file[s]      * = default; -x- = turn switch x off
-3      * 80386 Instructions                -4      80486 Instructions
-5      Pentium Instructions              -6      Pentium Pro Instructions
-Ax     Disable extensions                -B      Compile via assembly
-C      Allow nested comments            -Dxxx   Define macro
-Exxx   Alternate Assembler name        -Hxxx   Use pre-compiled headers
-Ixxx   Include files directory         -K      Default char is unsigned
-Lxxx   Libraries directory              -M      Generate link map
-N      Check stack overflow              -Ox     Optimizations
-P      Force C++ compile                 -R      Produce browser info
-RT     * Generate RTTI                   -S      Produce assembly output
-Txxx   Set assembler option             -Uxxx   Undefine macro
-Vx     Virtual table control             -X      Suppress autodep. output
-aN     Align on N bytes                  -b      * Treat enums as integers
-c      Compile only                      -d      Merge duplicate strings
-exxx   Executable file name             -fxx    Floating point options
-gN     Stop after N warnings             -iN     Max. identifier length
-jN     Stop after N errors               -k      * Standard stack frame
-lx     Set linker option                 -nxxx   Output file directory
-oxxx   Object file name                 -p      Pascal calls
-tWxxx  Create Windows app                -u      * Underscores on externs
-v      Source level debugging            -wxxx   Warning control
-xxxx   Exception handling                -y      Produce line number info
```

※上の図では印刷の都合上白地に黒という配色になっていますが、デフォルト設定では黒字に白い文字が出るようになっていていると思います。色を変えたい場合は **color** コマンドを使いますが、知らなくて今後困るというものではないため説明は割愛します。

ここで少しコマンドプロンプトについて説明します。

カーソルがある行の左側の部分(上では **C:\Users\tanno\_kazuho**)のところは今見ているフォルダ(カレントディレクトリ)の場所(フォルダへのパス)を示しています。カレントディレクトリを移動するには、

```
cd (移動先の場所)
```

とやると移動できます(間に半角スペースが入るので注意)。特に、一つ上のフォルダに移動するときは、

```
cd ..
```

(ピリオド2つ)と書きます。

自分で書いたプログラムをコンパイルするときには絶対使うので覚えておきましょう。

## ▶ 変数について

---

プログラムを作成するときには変数を設定し、それに数値や文字、式などを参照して使用します。変数というのは値を保管する場所ですが、その場所にはどんなものでも入れられるわけではありません。変数を宣言するにあたってどのような変数を保管する場所なのかを決めておく必要があります。

データ型名	保管できるもの	数値の範囲
int	整数	-2147483648 ~ 2147483697
float	実数	1.18E-38 ~ 3.40E+38
double	実数	2.23E-308 ~ 1.79E+308
char	文字(整数)	半角英数字1文字 (-128~127)

※変数の最大値・最小値は Windows 環境下におけるものです。

## ▶ 変数の宣言について

---

変数の宣言をするときにはデータ型名を宣言し、空白の後に変数を宣言します。例えば、

```
int a;
```

と入力すれば整数の変数 **a** を宣言することができます(半角スペースとセミコロンを忘れないように気を付けてください)。これにより変数 **a** が使えるようになりますが、宣言していない変数は使用することができないので注意してください。また、整数として宣言した変数に実数を入れるとプログラムが正しく動かなくなるので注意してください。

まとめて変数を宣言したい場合は、

```
int a,b;
```

のように1つ目の変数の後ろに「,」を入れるとその後にまた変数を宣言できます。実数や文字を使いたい場合は対応するデータ型名を使えばいいので、上の表を参考にしてください。

## ▶ プログラミングの基本

---

基本的に、それぞれの処理の最後尾には「;」を1個つけます。セミコロンをつけ忘れるとエラーが起きます。意外と見落としやすいので注意しましょう。

※あくまで私の感覚ですが、プログラミングのエラーの原因の大半がこの類のものだったりします。コンパイル前には見直しをしっかりとしましょう。

プログラムのソース中にメモとして文章を入れたい時はコメントという機能を使います。コンパイル時には、コメントは無視されます。

コメントの書き方は2種類あります。

```
//(文章)
```

「//」以降の1行を全てコメントにします。

(使えない場合もあります。なお、C++の場合は必ず使えます。)

```
/*(文章)*/
```

「/\*」と「\*/」の間に挟まれている部分をコメントにします。

コメントはメモとしても使えますが、「実装予定だけれど該当部分が未完成だからそこだけ飛ばしてコンパイルしたい」というような場合に、飛ばす部分のプログラムをコメントにしておく、という使い方もできます。今後のゲーム制作において便利なので、ぜひ覚えておいてください。

## ➤ 演算子について

C言語の演算子には、普段算数や数学で使うものとは異なる記号もあります。

演算子	意味
+	加算
-	減算
*	乗算
/	除算
%	除算の余り

## ➤ printf について

ここからいよいよプログラミングについて学んでいきます。

今回説明する `printf` は、冒頭でも述べたとおり文字を画面上に表示させる命令です。プログラムの例としては、

- `a01.c` ← C言語なので拡張子は `.c` です。

```
#include<studio.h>
int main (void){
    printf("私はエレ研部員です。¥n");
    printf("%d + %d = %d¥n",1,2,1+2);
    return 0;
} [7]
```

というような感じです。実際にメモ帳などのテキストエディタで打ってみましょう。また、**Tab** キーを打つか半角スペースを入れるとプログラムが見やすくなります。ただし、**全角スペースはエラーが起きるので絶対に入れないください。**

今回は「**Source**」というフォルダを作り、そこに保存しましょう。

■ コマンドプロンプトを開いてカレントディレクトリを移動する

プログラムを書けたらコマンドプロンプトで[**cd C:¥Source**](**cd** 自分のプログラムを入れておいたところ)と入力して実行します。そうすると前述のとおりカレントディレクトリの移動ができます。存在しないフォルダを指定した場合はエラーとなります。

■ コンパイルする

[**bcc32 a01.c**](**bcc32** 自分のプログラム名)と入力し、**Enter** キーを押します。この時、エラーがあるとエラーメッセージが表示されます。

■ 実行する

[**a01**](自分のプログラム名から.cをとったもの)と打つと実行されます。今回のプログラムを実行した場合、成功すればこのような形で表示されます。

```
私はエレ研部員です。
```

```
1+2=3
```

プログラムの説明ですが、

```
#include<studio.h>
int main (void){
    printf("私はエレ研部員です。¥n");
    printf("%d + %d = %d¥n",1,2,1+2);
    return 0);
}
```

**網掛け部分**は今後説明します。今回はプログラムを書くのに必要なもの程度に考えてくれればいいと思います。

**printf**は **printf("打ち出したい文章",数値 1,数値 2,...,数値 n);**

のように使われます。数値は今回は**%d** に対応しており、左から **1,2,...,n** となり、**%d** に数値が入力されます。また、“打ち出したい文章”の部分に数式を入力しても、それはあくまで「文章」としてとらえられてしまうため計算されません。計算したい場合は“数値”の部分に入力すると計算結果が表示されます。(上記の例で言うと、**1+2** と書いた部分は **3** と書いた場合と同じように表示されます。)



今回出てきた記号の説明をすると、

<code>%d</code>	出てきた順番に数値(10進数の整数)を表示する。
<code>%f</code>	出てきた順番に数値(実数)を表示する。
<code>%c</code>	出てきた順番に文字を表示する。
<code>¥n</code>	改行

※書く回数は後ろの数値の数と同じにすること。

また、「¥」は環境によっては「/」と表示されますが意味は変わりません。

### ➤ 練習問題

---

- 1.自分のクラス、番号、名前を `printf` で出力するプログラムを作ってください。
- 2.次の計算の答えを `printf` で出力するプログラムを作ってください。

$2578 + 1205$

$168 \times 21$

以上です。お疲れ様でした。