

## 第 1 回 PC の環境設定/printf

ソフトゼミ A では結構有名なプログラミング言語、「C 言語」を学んでいきます。第 1 回目は、この「C 言語」を自分の PC でも使えるようにする環境の設定の仕方と、最も簡単な関数の `printf` の使い方を説明していきます。

### 拡張子の表示

---

拡張子というのはファイルを見た時に、それがどのようなファイルなのか判断するために書いてある文字列のことです。自分で作ったプログラムがコンパイルできているかどうかの確認にも使えるのですが、Windows のデフォルト設定では拡張子が見えないので、見えるよう設定します。

**Xp:** 「マイコンピュータ」 → 「ツール」 → 「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] のチェックをはずす」 → 「すべてのフォルダに適用」 → 「OK」 → 「OK」

**Vista:** 「コンピュータ」 → 「整理」 → 「フォルダと検索のオプション」 → 「表示」 → 「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] のチェックをはずす」 → 「OK」

で設定完了です。

### コンパイラーを手に入れる

---

まず <http://www.codegear.com/jp/downloads/free/cppbuilder> の下から二番目にある **C++Compiler 5.5/Turbo Debugger** でアンケートなどに答える（メールアドレス以外はでたらめで構いません）などの手順を踏んで「freecommandlinetools2.exe」という exe ファイルを適当なところに落としてください。そして、その exe ファイルを実行していくと **C:\¥borland¥bcc55** という exe ファイルができると思います。ここまでできればコンパイラーの入手は成功です。

ちなみにコンパイラーとは C 言語で書いたプログラムを PC の理解できる形に変換する仕事をするのがコンパイラーです。

## 環境設定ファイル・環境変数をいじる

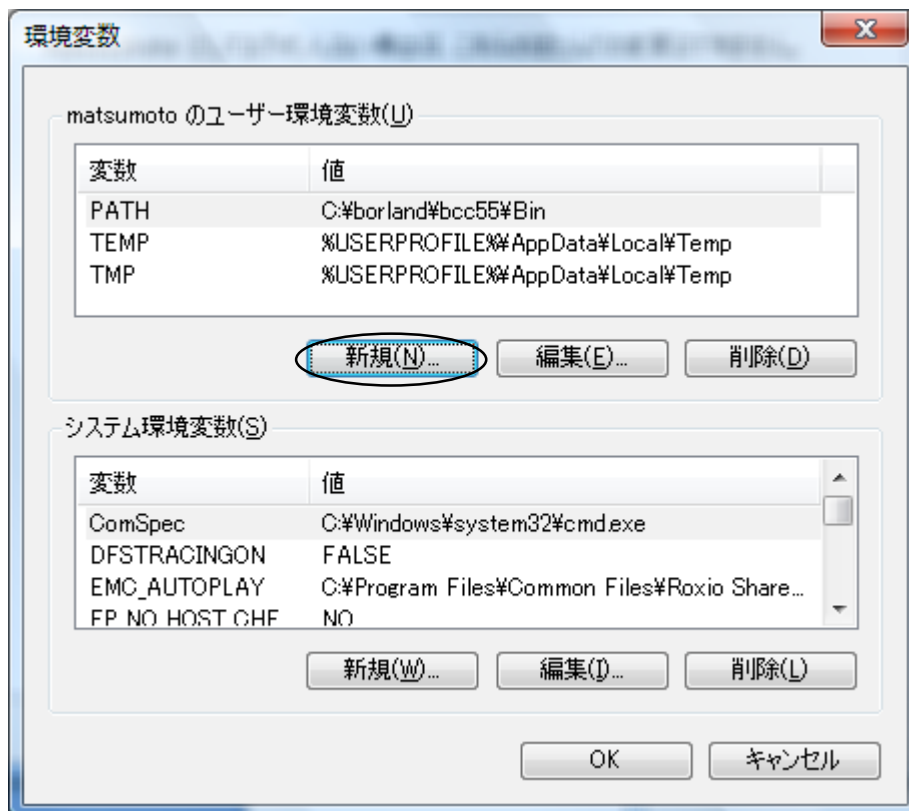
c:\¥Borland¥Bcc55¥Bin のところに

```
-I"c:\¥Borland¥Bcc55¥include"  
-L"c:\¥Borland¥Bcc55¥lib"
```

という内容の「bcc32.cfg」というファイルをメモ帳等で作ってください（拡張子に注意）。  
そして、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。

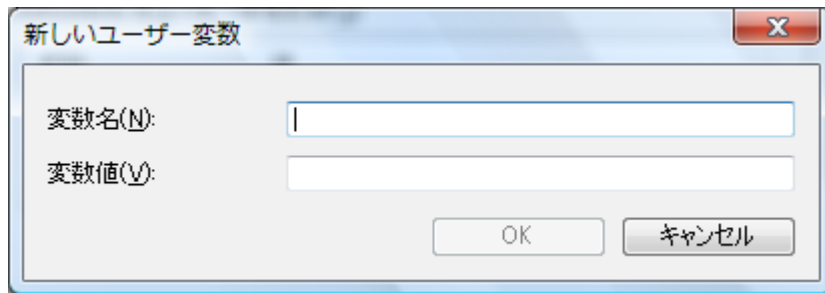


つぎに環境変数をいじります。

これを間違えるといろいろと大変なので慎重に行ってください。

「マイ コンピュータ」を右クリック → 「詳細設定」 → 「環境変数」  
とやると上のようなのができます。

丸で囲ってある「新規 (N)」というのをクリックしてください。



すると上のウインドウがでできます。

そこで

変数名 (N) に **PATH**

変数値 (V) に **C:¥borland¥bcc55¥Bin**

とうち「OK」をクリックしてください。

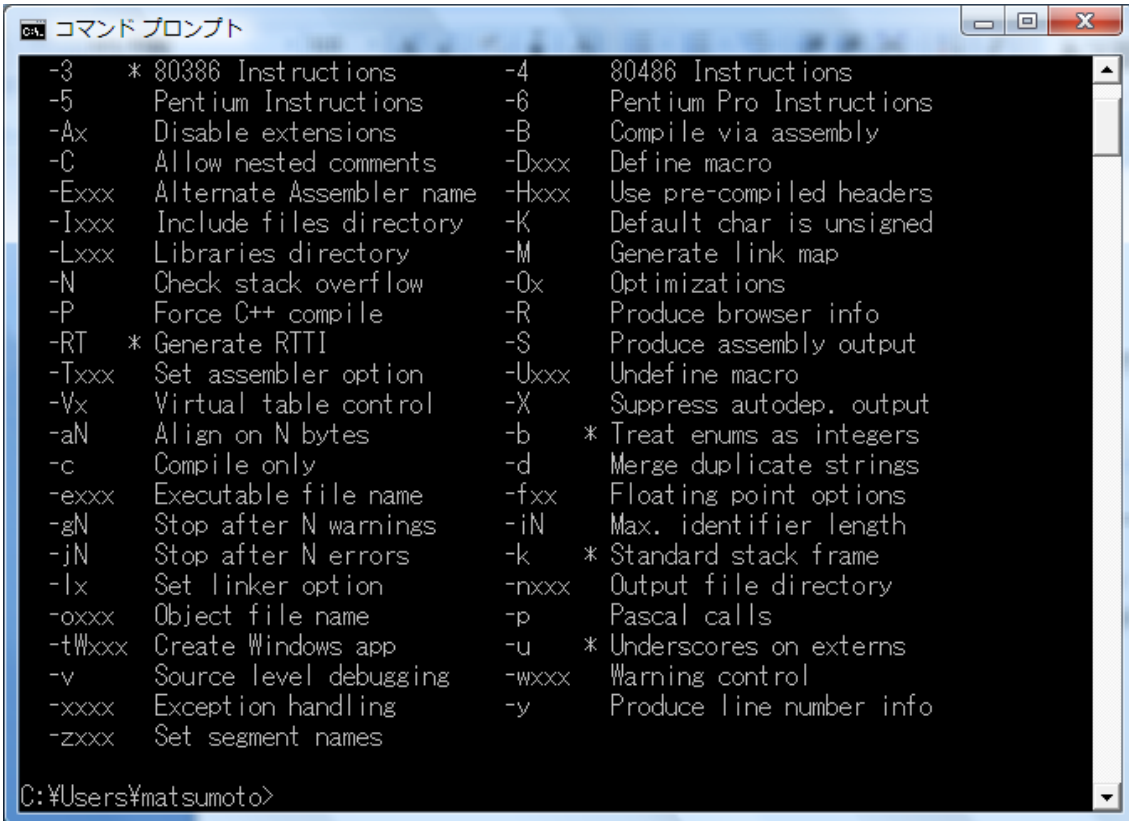
そのあと環境変数の丸の上のところに「**PATH**」があるかどうか、正しく打たれているかを確認し「OK」をクリックすれば設定完了となります。

なにか間違いがあっても「キャンセル」をクリックすれば元に戻るのもう一度やり直してください。

## コマンドプロンプト

とりあえず今までの設定が正しくできているかを調べてみましょう。

「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」とやるとコマンドプロンプトが開きます。ここで**[bcc32]**と打つと



```
コマンドプロンプト
-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
-zxxx Set segment names

C:¥Users¥matsumoto>
```

なかなか訳のわからない文字が出れば設定は無事成功しています。これ以外が出てきているようなら、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てください。

ここで少しコマンドプロンプトについて説明していきます。

```
C:¥Users¥matsumoto>
```

カーソルがある行の左側の下線部のところが今見ているフォルダ（カレントディレクトリ）のアドレスを示しています。フォルダを移動するには**[cd (移動先のアドレス)]**とやると移動できます。

自分で書いたプログラムをコンパイルする時に必ず使うので覚えておきましょう。

## printf

---

printf とは文字を画面上に表示する関数です。とりあえずサンプルプログラムを実行するところまでを順を追って書いておきます。

- ・ lesson01.c (←\*注意 C 言語なので拡張子は.c です)

```
#include <stdio.h>

int main( void){
    printf( “エレ研です。よろしく¥n”);
    printf( “%d+%d=%d¥n”,1,1,1+1 );

    return 0;
}
```

を適当なところに保存する。(例では c ドライブに **Source** というフォルダを作ってその中に保存します。なお、プログラム内容の意味は後で説明します。)

- ・ 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

コマンドプロンプトで[cd c:¥Source] (cd 自分のプログラムをおいたところ) と打ち実行。

```
C:¥Users¥matsumoto>cd c:¥Source
c:¥Source>
```

- ・ コンパイルする

[bcc32 lesson01.c] (bcc32 自分のプログラム名) と入力して実行。

```
c:¥Source>bcc32 lesson01.c
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
lesson01.c:
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland
c:¥Source>
```

### \*注意

ここでエラーがでたらプログラムが間違っていることになるので要見直し

- ・ 実行する

[lesson01] (自分のファイル名から.cをとった状態) と打つ。

```
c:¥Source>lesson01
エレ研です。よろしく
1+1=2
c:¥Source>
```

以上で終了です。

ここで先ほどのプログラムの説明をしていきます。

```
#include <stdio.h>

int main( void){
    printf(“エレ研です。よろしく¥n”);
    printf(“%d+%d=%d¥n”,1,1,1+1);

    return 0;
}
```

塗ってある部分は今のところ理解しなくていいところです。

書き出しはこのようなするものだと覚えておいてください。

よって今回理解するのは `printf` という関数だけです。

この `printf` は

```
printf(“ 打ち出したい文章 ”計算式 1,計算式 2,.....,計算式 n);
```

という使い方です。

あと、サンプルの「打ち出したい文章」のところにあった `%d` や `¥n` の意味がわからない文字は

`%d` … 出てきた順番に計算式 1 からの答え(10 進数)を表示する。

\*注意 書く回数は後ろの計算式の数と同じにすること

`¥n` … 改行

という意味です。

このような特殊な文字はいくつかあり

`%x` … 計算式の答え (16 進数) を表示する

`%o` … 計算式の答え (8 進数) を表示する

`%%` … %を表示する      `¥¥` … ¥を表示する

などなど、他にもあるので暇なときにでも自分で調べてみましょう。

## プログラムの基本

---

・それぞれ処理の最後尾に「; (セミコロン)」を一個つけます。忘れやすいことなので注意しましょう。

・プログラムのソース中にメモとして文章を入れたいときにはコメントアウトと言うのを使います。コンパイルするときにはコメントアウトされた部分は無視されます。

コメントアウトの方法は二種類あります。

`// (文章) ]` … `//` のあとの一行すべてをコメントアウトします。

`/* (文章) */` … `/*` と `*/` の間に挟まれている部分をコメントアウトします。