

ソフトゼミ A 第 1 回 PC の環境設定/printf

注意

環境によっては記号「¥」(エンマーク)は「\」(バックスラッシュ)と表示されることがありますが、同じ意味です。ゼミ A, B, C の資料では「¥」(エンマーク)で統一します。

■ はじめに

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(ひとかたまりの命令)の一つである「printf」の使い方を説明していきます。

■ コンパイラを手に入れる

まず下記のアドレスのサイトに行きます。

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

「C++コンパイラのダウンロード」の項目の中にある「こちらのページ」をクリックすると「C++Compiler/Turbo Debugger ダウンロード登録フォーム」に飛ぶのでここで個人の名前やメールアドレス等を登録させます。登録するとダウンロードできるページに行けます。(ダウンロードしたファイルの名前は `freecommandlinetools.zip` のはずです。)

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を開くためのパスワードが送られるのでチェックしましょう。ダウンロードした `zip` ファイルを解凍すると出てくる「`freecommandlinetools2.exe`」という `exe` ファイルを適当なところに落とします。そしてその `exe` ファイルを実行していくと(標準では) `C:\¥borland¥bcc55` というフォルダができると思います。ここまでできればコンパイラの入手は成功です。

ちなみにコンパイラは、C 言語で書いたプログラムを PC が理解できる形に変換する仕事(コンパイル)をします。

■ 拡張子の表示

「拡張子」というのはファイルをみたときに、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するのに使えますが、Windows のデフォルト設定では拡張子が見えないので、見えるように設定します。

- 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」

で設定完了です。

- Windows Vista, 7

「コンピュータ(一)」を開く(デスクトップにない場合は Windows キーと「E」キーを同時押し)→メニューバー(ファイル (F) 編集(E) ...っていうやつ)が表示されていない場合には alt キーを押して表示させる→「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」

■ 環境設定ファイル・環境変数をいじる

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)
あと、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」

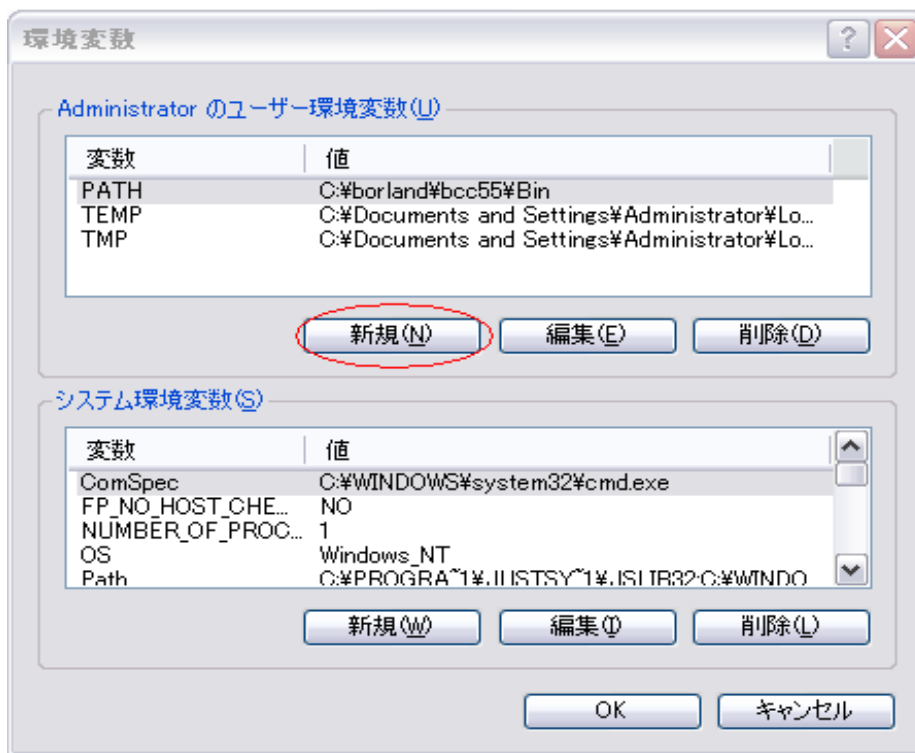
- Windows Vista, 7

デスクトップの[コンピュータ(一)]を右クリック→「プロパティ」→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」

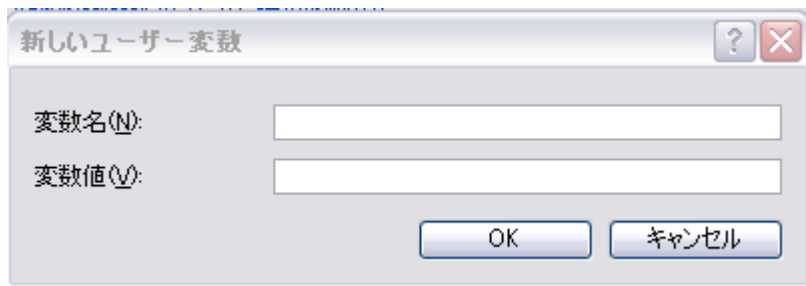
※ デスクトップに「コンピュータ(一)」のアイコンがない場合には、
Vista) 「スタート」→「コントロールパネル」→「クラシック表示」→「システム」→画面左側「タスク」内「システムの詳細設定(A)」→「続行」→「詳細設定」→「環境変数」

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)」→「続行」→「詳細設定」→「環境変数」

すると、次ページにあるような画面が出てきます。



赤い丸で囲ってある「新規 (N)」をクリックしましょう。(画面は WindowsXP の場合の一例です。環境によって微妙に異なります。)



すると、上のウィンドウが表示されます。

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。

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、ただしく打たれているか確認してから「OK」を押せば設定完了です。なに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りますのでもう一回やり直しましょう。

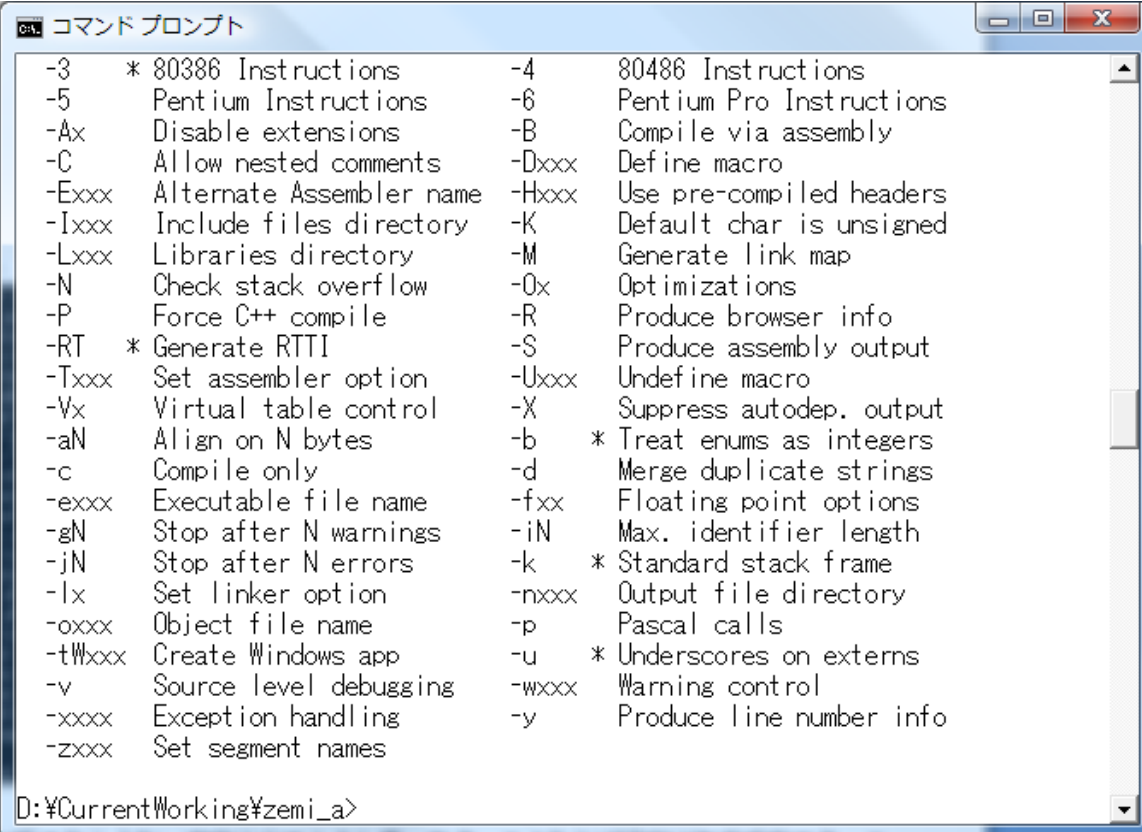
この操作は、コマンドプロンプトからコンパイラを扱うための手続きで、これで、コマンドプロンプトがコンパイラのありか(パス:path)をわかるようになりました。これによって、どんなフォルダの階層からでもコンパイルができるようになりました。

■ コマンドプロンプト

※ゼミ A の配布資料では、印刷の都合上、コマンドプロンプトの画像は全て白地に黒文字を採用しています。Windows のデフォルト設定では、黒地に白文字です。自分の見やすい配色で使ってください。

今までの設定が正常にできているか調べてみます。

「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で「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、素早く出せる後者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。



```
cmd 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
-xxx    Exception handling         -y      Produce line number info
-zxxx   Set segment names

D:¥CurrentWorking¥zemi_a>
```

すると、上の画面のような表示が出てくると思います。こうなれば設定は無事成功です。これ以外がでてきたら、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。

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。

-v	Source level debugging	-wxxx	Warning control
-xxxx	Exception handling	-y	Produce line number info
-zxxx	Set segment names		

C:¥Documents and Settings>■

カーソルがある行の左側の部分(上では「C:¥Documents and Settings」)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。特に、一つ上のフォルダに上がる場合は

```
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絶対使うので覚えておきましょう。

■ printf

では、いよいよプログラムを書いていきます。printfとは文字を画面上に表示する命令です。とりあえずサンプルプログラムを実行する所までを順を追って書いておきます。

・a01.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;
}
```

とテキストエディタ(メモ帳など)で書いてみてください。

なお、網掛け部分の行(int main(void){ ~ }の間の行)の先頭でTabキーを打つか、半角スペースを入れるかすると、見やすいプログラムになります。(全角スペースは絶対に打たないこと)

今回は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しょう。

(プログラムの内容の意味は後述するので今はスルーします。)

・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 自分のプログラムおいたところ)と入力して実行します。(Cドライブ下にSourceフォルダがない場合にはエラーとなります。)

C:¥Documents and Settings>cd C:¥Source■

※移動前のフォルダは人によって異なります。上のものは一例です。

- ・コンパイルする

[bcc32 a01.c] (bcc32 自分のプログラム名) と入力して **Enter** キーを押します。

```
C:¥Source>bcc32 a01.c
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
a01.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見直しましょう。

- ・実行する

「a01」（自分のファイル名から.cをとった状態）と打つ。

```
C:¥Source>a01
エレ研です。よろしく
1 + 1 = 2

C:¥Source>
```

以上で終了です。

ここで先程のプログラムの説明をします。

```
#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;
}
```

網掛け部分は今のところ理解しなくていいので、こういう風に書き出さなきゃいけないって覚えおきましょう。

よって今回理解するのは printf(“ ~ ”)の部分だけです。

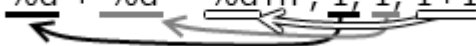
この printf は

```
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。

今回の例では、このようになります。↓

```
printf( "%d + %d = %d\n", 1, 1, 1+1 );
```



サンプルプログラムの打ち出したい文章に表記された%d や\n等の文字は

%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。

こういうなんか特殊な文字かはいくつかあります。

%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 \\\n…\nを表示

などなど、他にもあるので自分で暇だったら調べてみましょう。

■ Cプログラミングの基本

・基本、それぞれ処理の最後尾には「;」(セミコロン)を一個つけます。

・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と言うものを書きます。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二種類です

…[/] (文章)] これは[/]のあとの 1 行すべてをコメントにします。

(この方法は使えない場合があります。なお、C++の場合は必ず使えます。)

…[/* (文章) */] これは[/*]と[/*]の間に挟まれている部分をコメントにします。

例えば、

```
#include <stdio.h>
/* zemi_a 01 “say hello” */
int main( void ){
    printf(“Hello, World!\n”); /* Hello, World!と出力 */
    return 0;
}
```

というソースをコンパイルした時、網掛け部分は無視されます。

■ 練習問題

先に作ったサンプルの「a01.c」を改造して、以下の問いに答えよ。

1. 「明治大学エレクトロニクス研究部」と表示、改行して終了するプログラムを書け。
2. $333 + 301$ の計算結果を出力、改行して終了するプログラムを書け。