

ソフトゼミ A2016 第 1 回

PC の環境設定/printf

今回からソフトゼミ A が始まります。今日はソフト班活動において基本となる「C 言語」を使いこなすために、パソコンの環境設定を行います。また文章や数値を出力する命令「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。コンパイラ取得方法は以下の通りです。

(windows10 の方は.net 3.5 をインストールして適用したあとに行ってください)

- ① 次のサイトにアクセス
→<http://www.embarcadero.com/jp/free-tools>
- ② 「C++コンパイラのダウンロード」の「こちらのページ」をクリック。
- ③ 「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を登録。
- ④ ダウンロードページに行くので、そこでダウンロード
(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はず)←忘れないように
- ⑤ ダウンロードしたファイル(freecommadlinetoos.zip)を解凍、その後
(freecommadlinetoos.exe)を実行
- ⑥ 実行後(C:\¥borland¥bcc55)というフォルダが作られた確認する!!!!

ここまでで何かわからないことや出来ないことがあったら、すぐに周りのエレ研部員にきいてください。

★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。先程のファイルを例に挙げると(.exe)の部分が拡張子です。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もにも使えますが、Windows のデフォルト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 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、10

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、

```
-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

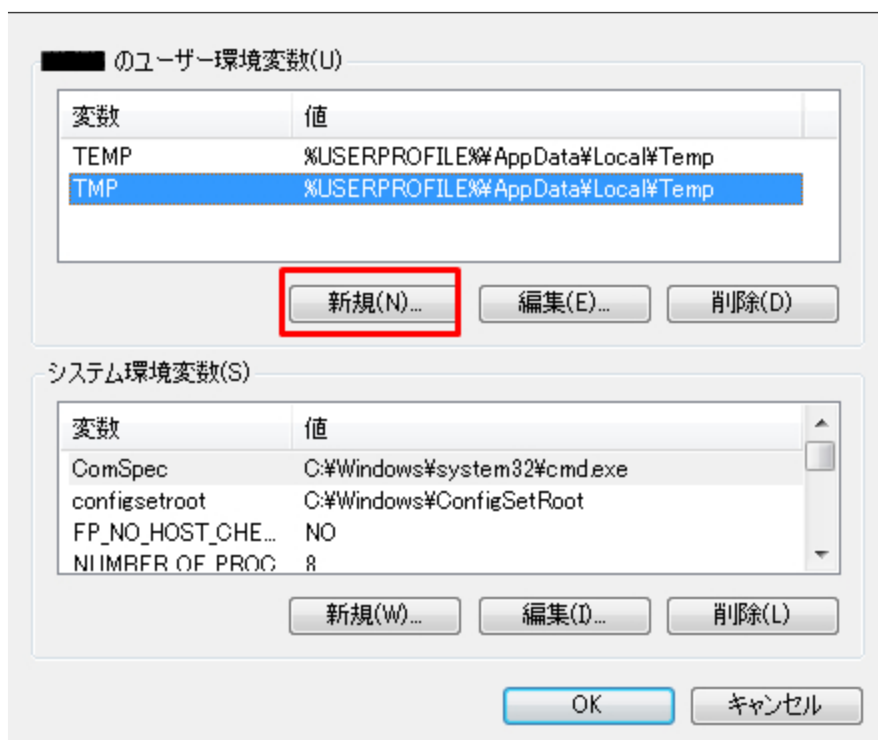
「スタート」→「コントロールパネル」→「システムとセキュリティ」→

「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」

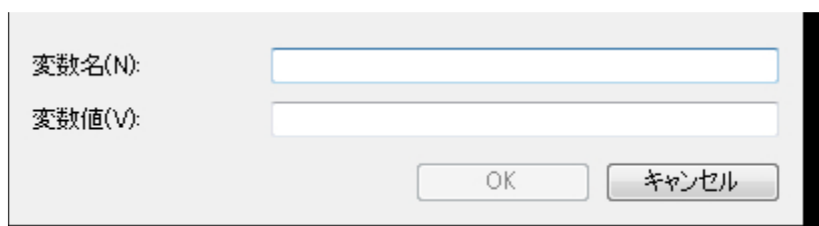
■ Windows 8、10

「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。

```
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
```

ここで少しコマンドプロンプトについて説明します。

カレントディレクトリを移動するには

cd (移動先の場所)

とすると移動できます(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 ..

と書きます。

よくわからない人は部員に聞いてみましょう!!!!

この動作はよく使うので覚えておきましょう。

★では!!実際にプログラミングしていきましょう!!★

★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の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はデータ型名を宣言して、空白の後に変数を宣言します。

```
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★

まずプログラミングが出来る友達を見つけましょう!!という冗談はおいといて基本中の基本をおさえていきます。

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。

プログラムのソース中にメモとして文章を入り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行をすべてコメントにします。

```
 /*(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、普段数学で使うものと異なる記号もあります。

- 「+」 . . . 加算
- 「-」 . . . 減算
- 「*」 . . . 乗算
- 「/」 . . . 除算
- 「%」 . . . 除算の余り

では、実際プログラミングしてみましょう!!

★printfについて

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。まず適当なファイルをつくってメモ帳やエディタで適当な名前を書くところをひらきましょう。(このとき拡張子は.cにすることに注意)

例 a01.c

みたいな感じです。

↓サンプルコード

```
#include<stdio.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;
}
```

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でください。

★コマンドプロンプトからのカレントディレクトリの移動

プログラムが書けたらコマンドプロンプトで
cd C:¥(自分のプログラムを作成したところ)
と入力してエンター。

(例 cd C:¥eleken)

★コンパイル

コマンドプロンプトで

bcc32 (プログラム名)と入力してエンター!!

(例 bcc32 a01.c)

なおこの際エラーがあると、エラーメッセージが出ます。

よくわからない場合は部員にきいてみてください。優しく教えにいきます!!

★実行する

コマンドプロンプトで自分のつくったプログラムから「.c」をとったものを
打つと実行されます。

ちなみにさっきの実行結果は

私はエレ研部員です。

1 + 2 = 3

と、なります。

さきほどのサンプルコードですが

```
#include<stdio.h>、int main(void)、return 0
```

は、今はプログラミングに必要なんだなと思ってくれれば大丈夫です。

printfはprintf(“出力したいもの”);のように使います。

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。また出力したい部分に数式を入力しても計算されません。

また今回でてきた記号の説明をしますと

%d . . . 出てきた順番に数値を表示する。

%f . . . 出てきた順番に数値(実数)を表示する。

%c . . . 出てきた順番に文字を表示する。

¥n . . . 改行

また¥と/の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. 次の計算の答えを出力するプログラミングを作ってください。

$$3188 + 9407$$

$$128 * 64$$

以上です。

わからなかったり、質問がある場合はすぐ部員にききましょう!!

★printfの補足

printfでの出力できましたが、行数をそろえたい場合やdouble型の変数を出力するときには小数点以下が長々と出力されるのをふせぐ方法を教えたいと思います。

まず行数をそろえる場合ですが、例えば改行して10と100の値を出力する際に右揃えにする場合です。

```
printf("a = %3d¥n b = %3d¥n", 10, 100);
```

このように%とdの間に揃えたい桁数を入力することで簡単にそろえることができます。あまりメリッとがないように思えますが、今後for文などで知らないうちに桁数が上がることもあるので、その時に役立ちます。

double型の変数を出力する際に%fで出力することを習いましたが、実際に出力してみると、

```
1.000000000
```

のように出力されることがあります。そこで精度指定という技術が必要になってきます。これの使い方は

```
double d = 3.14
```

```
printf("%4.2f¥n", d);
```

と出力します。この中の%4.2fですが、始めの4は出力する全文字数を、後ろの2は小数点以下の出力する数を示しています。

練習問題

57 ÷ 7 の結果は小数点第2位まで表示してください。

変数についての補足説明

前回のゼミで変数についての説明が不十分だったため、補足説明をしたいと思います。まずは以下のサンプルコードをみてください。

```
#include<stdio.h>
int main(void){
    int a; //変数 a の宣言(int 型のため a は整数)
    double b; //変数 b の宣言(double 型のため b は実数)

    a = 123; //a に 123 という値を代入
    b = 1.23; //b に 1.23 という値を代入

    printf("%d¥n",a); //a(=123)という変数に代入した値の出力
    printf("%.2f¥n",b); //b(=1.23)という変数に代入した値を下 2 桁で出力

    return 0;
}
```

今、自分で宣言した変数にそれぞれ値を代入しています。よって%d や%f に代入するものを変数にしてもその変数に値が代入されているのでその数字の出力がされます。