

## MinGW による C 言語によるプログラミング環境の作り方 (Windows 環境)

ここでは、MinGWというライブラリを使って、無償のC言語のコンパイラgcc とC++言語のコンパイラg++ をご自宅などのPCにインストールし、C言語/C++言語のプログラムの作成・コンパイル・実行を行う方法について、手順を追って解説します。 なお、リンク先の情報は最新かどうか分かりませんし、内容を保証するものでもありません。あくまでも参考程度とお考え下さい。

### 1) C言語によるプログラミングについて

まずソースプログラムをエディタで作成します。ここではエディタとしてTeraPad を用いることにします。TeraPad のインストールは、窓の社というフリーソフトのサイトをお勧めしておきます：

<http://forest.watch.impress.co.jp/library/software/terapad/>

ダウンロードした実行型ファイルをクリックして、指示に従ってキーを押してインストールして下さい。デスクトップにショートカットを作ると良いでしょう。

### 2) 練習用のディレクトリ(フォルダ)を作る

プログラミングを行うディレクトリ(フォルダ)を作りましょう。例として、C:¥algorithm とします。フォルダが出来たらダブルクリックして開いておいてください。

### 3) テストプログラム hello.c の作成

C言語のテストプログラムを作りましょう。1)でインストールしたTeraPadを、デスクトップのショートカットをダブルクリックして起動して次のプログラムを入力しましょう。「ファイル」の「名前を付けて保存」をクリックし、フォルダ algorithm に hello.c という名前で保存します。

```
#include<stdio.h>
int main(void)
{
    printf("Hello, world.¥n");
    return 0;
}
```

図1 C言語のサンプルプログラム hello.c

これにより、2)で作ったフォルダ中に hello.c というファイルが作られます。C言語では元のプログラムを拡張子に .c を付けて作ります。このように、人がわかる仕様で書かれた元のプログラムをソース(source)プログラムと呼びます。

### 4) C言語コンパイラ gcc とC++言語コンパイラ g++ のダウンロードと準備

3)で作ったソースプログラムは、人が分かる言語で書かれていますが、コンピュータに計算を行

わせるためには**コンパイラ (compiler)**と呼ばれる翻訳プログラムを使って、ソースプログラムを機械語に翻訳し、**●●.exe** あるいは **●●.obj** を作ります。

現在、無償で利用することができる**C言語/C++言語のコンパイラ**はいくつかあります。統合環境を提供している **Visual Studio** が便利ではありますが、統合環境に馴染みが薄い人も多いと思いますので、ここでは、もっと**シンプルなコマンドラインを用いる方法**をご紹介します。

**MinGW**と呼ばれるものがあります。これが最もシンプルだと思います。MinGWは、次のサイトを参考にとすると良いでしょう：

<http://yohshiy.blog.fc2.com/blog-entry-292.html> ← サイトA

ここには**MinGW**の説明と、インストールの方法が書いてあります。

そこに書いてありますように、**MinGW | Minimalist GNU for Windows** のホームページ：

<http://www.mingw.org/>

に行きます(英語のサイトです)。



図2 MinGWのホームページ

この右上の赤色の部分をクリックしましょう。ダウンロードが始まります。

ダウンロードが無事に終了したら、ダウンロードしたファイル (**mingw-get-setup.exe**) を実行しましょう。上記の**サイトA**に書いてあるように、基本的にデフォルトの設定で問題ありません。最後の方で**Run**を押すといくつかのファイルが展開されてから次の画面が出ます(もし違うようなら左側の**[Basic Setup]**をクリックしてください)。

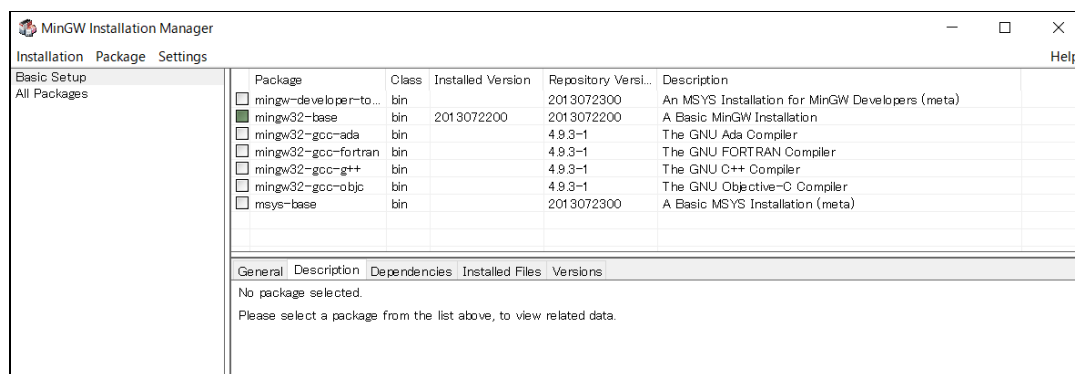


図3 インストールするプログラムを選ぶ画面

このとき、C言語 と C++言語 を使うだけなら、mingw32-base と、mingw32-gcc-g++ の2つを選ぶだけで充分です。これら2つを選んで下さい。具体的には、それぞれのプログラムでマウスの右クリックをして出たメニューで[Mark for Installation]を選んで下さい。選び終わったら、メニューの[Installation]→[Apply Changes]を実行します。

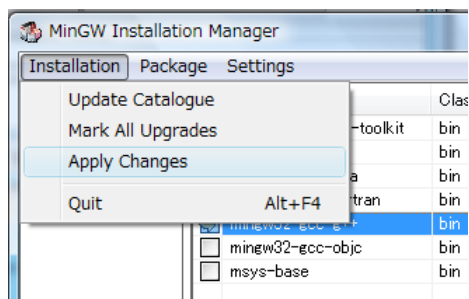


図4 インストールを実行するメニュー

次の画面で[Apply]をクリックするとファイルの展開とインストールが始まります。

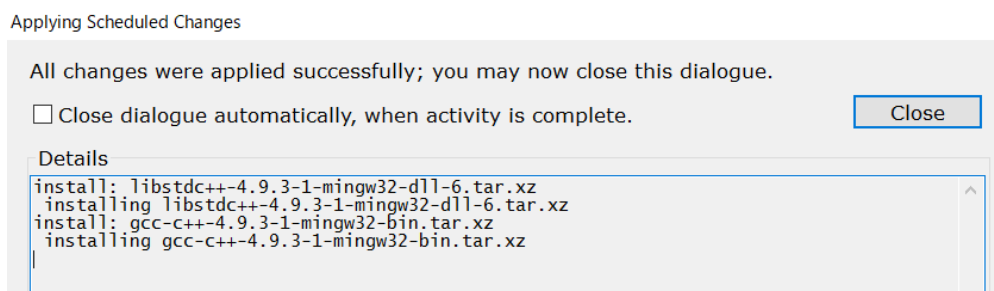


図5 最後に表示される確認画面

上の画面になればインストール成功ですので、[Close] を押して図5の画面を消して下さい。最後に図3の画面の [Installation]→[Quit] をクリックしてインストールは終了です。

以上の操作によって、(デフォルトの設定の場合は) C:¥MinGW に必要なファイルがインストールされます。

Cドライブを開けると、MinGWというディレクトリ(フォルダ)が作られていることと思います。「コンピュータ」→「Windows(C:)」をダブルクリックして確認してみてください。MinGWのフォルダも開けると、次のようなフォルダがその中にあることがわかります：

- 名前
- bin
  - include
  - lib
  - libexec
  - mingw32
  - share
  - var

図6 C:¥MinGWの下に作られるフォルダ

binフォルダをダブルクリックすると、binフォルダの中に gcc を始めとするプログラムがあることがわかります。このため、後でこれらのプログラムをシステムに知らせるために、path を設定する必要があります。そうしないと、gcc などが「コマンドプログラム」であることがシステムに伝わらないからです。システムがコマンドプログラムを探しに行く場所のリストを示すものが、環境変数 PATHです。環境変数というのはシステムが参照する重要なパラメータの1つです。

一般に、環境変数をいじるのは慎重にやる必要があります。不用意にいじると、OSの起動時に重大な問題が起きたりします。十分に注意してやって下さい。 比較的簡単に行う方法がネット上などで説明されています。

いま「環境変数 path 設定」で googleで検索したら、ちょうど、pathの設定が悪いので gccがない、と言われたというQ&Aがあったので次に示しておきます：

<http://note.chiebukuro.yahoo.co.jp/detail/n129063>

この回答のずっと下の方に、「PATHを設定する」という部分があります。「コントロールパネル」→「ユーザーアカウント」→「ユーザーアカウント」に進み、左側の「環境変数の変更」をクリックしましょう。環境変数の設定画面が開かれます。Path で「編集」を押して編集できます。ですが、この方法はお勧めしません。間違えると大変だからです。そこで、フリーソフトで簡単に環境変数を設定できる次のプログラムをお勧めしておきます：

<http://forest.watch.impress.co.jp/library/software/redmondpath/>

Redmond Path というものです。RdemondPath.zip がダウンロードされたら、そのファイルをクリックすると、Redmond Path.exe と Redmond Path.html が作られますので、Redmond Path.exe をダブルクリックして下さい。そして、「はい」で実行すると、次の画面が出ます。

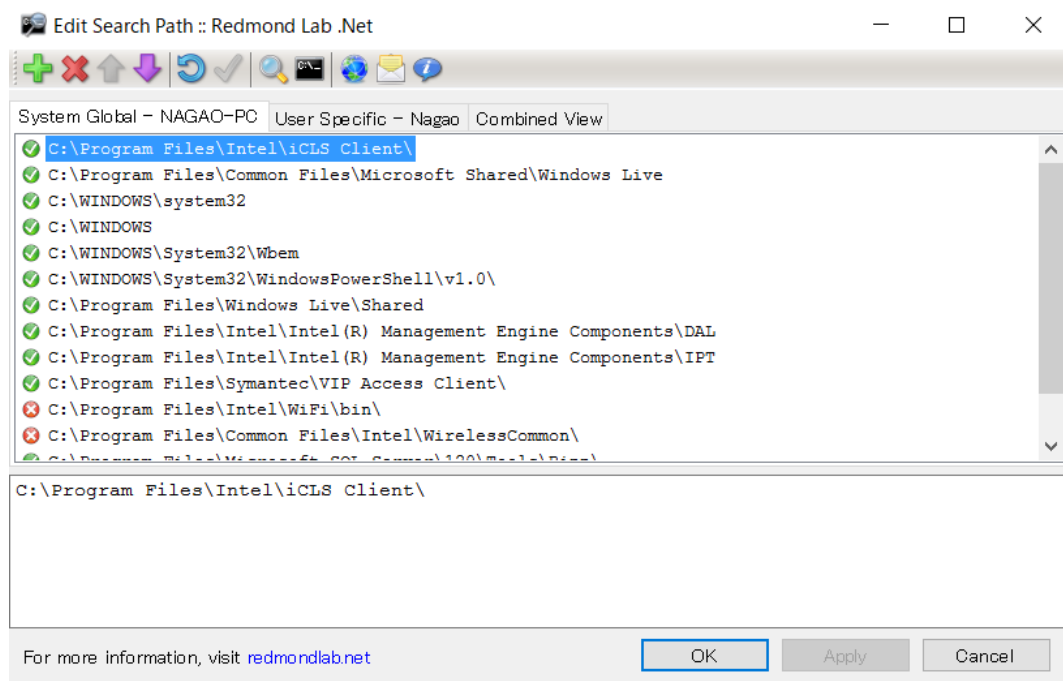



図7 環境変数PATHを変更するためのフリーソフトRedmond Pathの実行画面

メニューの  をクリックし、「フォルダーの参照」が出たら、「コンピュータ」→「Windows(C:)」→「MinGW」→「bin」で「OK」を押してください。そうすると、上の画面の最後に、今回加えるべき

フォルダである `C:\MinGW\bin` が追加されたことと思います。そうしたら「OK」を押して画面を閉じて下さい。これで設定は終わりです。MinGWによる、C言語およびC++言語のコンパイルが実行できるようになりました。

## 5)コンパイラ gcc を使ったC言語のプログラムのコンパイルと実行方法

今回は簡単のため、統合環境ではなく コマンドラインでコンパイラを実行します。Windows の付属プログラムの「コマンドプロンプト」を使います。コマンドプロンプトは、Windows 10 ではちょっとわかりづらい場所にあります。一番左下のマークをクリックしてすべてのプログラムを表示した際、「Windowsシステムツール」の下にあります。「Windowsシステムツール」をマウスで左クリックして表示される「コマンドプロンプト」です。これから何度も使うものなので、「コマンドプロンプト」でマウスの右クリックをし、「その他」→「タスクバーにピン留めする」を左クリックして、タスクバーに出しておくくと便利です。これで、すぐに「コマンドプロンプト」を実行することができます。では、タスクバーの「コマンドプロンプト」を左クリックして実行しましょう。

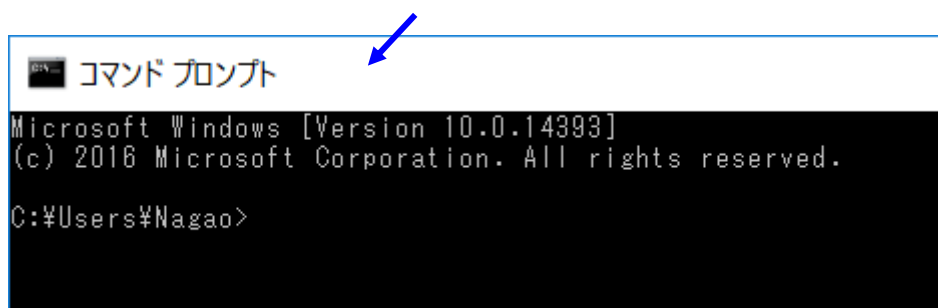


図8 コマンドプロンプトの初期実行画面

ここで `C:\User\ユーザ名>` となりますが、これが初期のフォルダです。画面の文字が小さい場合は、図8の青い矢印の部分で右クリックし、「プロパティ」押して「"コマンドプロンプト"のプロパティ」を開けて、「フォント」の「サイズ」を16くらいにすると良いでしょう。見易くなります。

このフォルダ(`C:\User\ユーザ名`)は、練習用に2)で作った `C:\algorithm` ではありません。この状態から、次のコマンド `cd` (change directory) で、練習用のフォルダ `C:\algorithm` に移動できます。

```
C:\User\Nagao>cd C:\algorithm ↓ (↓はEnterキー)
```

これによって、次のような表示になり、練習用のフォルダに移ったことがわかります。

```
C:\algorithm> ■
```

ですが、これを毎回やるのは面倒ですね？ その場合はコマンドラインの作業用フォルダを変更すれば、毎回それをやる必要がなくなります。「Windowsシステムツール」をマウスで左クリックし、「コマンドプロンプト」でマウスの右クリックをし、「その他」→「ファイルの場所を開く」を左クリックして、ショートカットが保存されているフォルダを開けます。

名前	更新日時	種類
PC	2016/07/16 17:25	ショートカット
エクスプローラー	2016/07/16 17:25	ショートカット
コマンド プロンプト	2017/02/12 19:28	ショートカット
コントロール パネル	2016/07/16 17:25	ショートカット
デバイス	2016/07/16 17:25	ショートカット
ファイル名を指定して実行	2016/07/16 17:25	ショートカット
既定のプログラム	2016/07/16 17:25	ショートカット

図9 コマンドプロンプトのフォルダ

この「コマンド プロンプト」を右クリックし、「プロパティ」を左クリックします。この「作業フォルダ」の中に図10のように「C:¥algorithm」とキーボードから入力して「OK」を押して下さい。

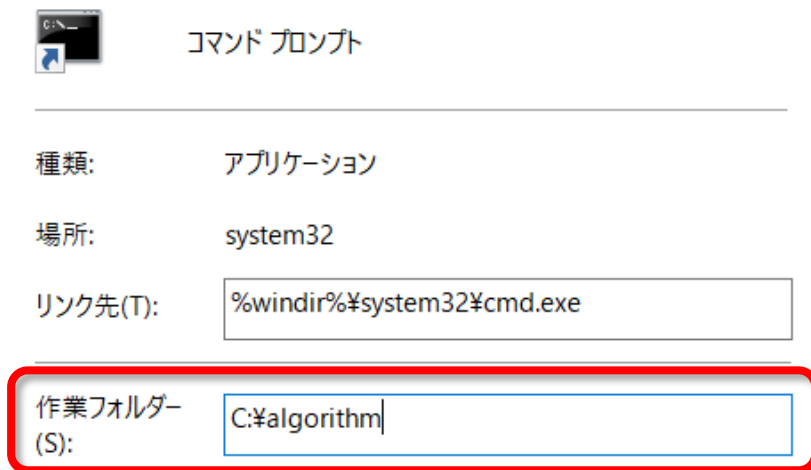


図10 コマンドプロンプトのプロパティ設定画面

これにより、起動時の作業フォルダが C:¥algorithm になります。次に、図9の状態ですべての「コマンドプロンプト」を右クリックし「タスクバーからピン留めを外す」を左クリックして、一旦、タスクバーから外して下さい。再び、「コマンドプロンプト」を右クリックし「タスクバーにピン留めする」を左クリックして下さい。そして、試しにタスクバーのコマンドプロンプトを左クリックして起動すると、最初から作業フォルダが C:¥algorithm になっていることがわかります。これで毎回移動する必要がなくなりました。

以上でコンパイルの準備は終了です。

さて、サイトAの下の方の「コンパイル」以下に具体的なコンパイル方法が書いてありますが、次に説明します。

### ● C言語のソースプログラム ●●.c の gcc によるコンパイル方法:

例えば、hello.c をコンパイルし、実行型ファイルである hello.exe を作るためには、次のようにコマンドラインで入力します。



```
C:¥algorithm>gcc -o hello.exe hello.c↓ (↓はEnterキー)
```

ここで、**-o hello.exe** を省略すると、デフォルトの **a.exe** というファイルができますが、どのソースプログラムをコンパイルしたときも **a.exe** という同じ名前になってしまいますので、ハイフンオー、スペースに続いて出力ファイル名(今回は **hello.exe**)をいつも入れるようにしましょう。

先ほど、TeraPad で **hello.c** を作って保存しましたので、上のコマンドをそのまま打ち込むと、**hello.exe** が作られるはずです。algorithm フォルダにどのようなファイルがあるかを確認するためには、通常の Windows のフォルダを見るように、「コンピュータ」の下の「Windows(C:)」の下の「algorithm」をクリックして表示します。または、次のように中身をコマンド **dir** で表示することもできます。

```
C:¥algorithm>dir↓
```

**hello.c** に入力ミスがなければ、コンパイルが正常に終わって、**hello.exe** が出来ているはずですので、次のようにコマンドから入力するか、

```
C:¥algorithm>hello↓ (.exeの入力は不要)
```

あるいはフォルダ中の **hello.exe** をダブルクリックして実行して下さい。なお、ファイルをダブルクリックした場合、プログラムが終了した時点でいったん表示を止めるようなプログラムになっていないと、一瞬だけ画面が出て、それで終わって消えてしまいます。このため、コマンドラインで実行型ファイル名を入力して実行した方が良いです。上記のように入力すると、画面に次のように表示されるはずです。

```
C:¥algorith>hello
Hello, world.

C:¥algorith>
```

このようにして、色々なC言語のプログラムをコンパイルし、実行することができます。

ご参考までに、C++言語のソースプログラム(拡張子は.cpp)をコンパイラ **g++** でコンパイルする方法を次に示しておきます。

### ● C++言語のソースプログラム ●.cpp の g++ によるコンパイル方法:

先ほどのテストプログラムに対応する C++言語のサンプルプログラムを次に示します。エディタのTeraPadで作ってから、「名前を付けて保存」をクリックし、フォルダ **algorithm** に **hello.cpp** という名前で保存します。

```
#include<iostream>
using namespace std;

int main(void)
{
    cout << "Hello, world." << endl;
    return 0;
}
```

図11 C++言語のサンプルプログラム hello.cpp

C++言語のソースプログラム(拡張子は.cpp)をコンパイルするときは、gcc ではなくて、g++ という名前のコンパイラプログラムを使います。例えば、hello.cpp という名前のソースプログラムをコンパイルするときは次のように入力します。

```
C:¥algorithm>g++ -o hello.exe hello.cpp ↓
```

hello.exe を実行するときは、次のようにコマンドから入力しましょう。

```
C:¥algorithm>hello ↓ (.exeの入力は不要)
```

#### 【C言語・C++言語の関係のファイル(拡張子:.c, .h, .cpp)を開くプログラムについて】

フォルダ中のC言語・C++言語の関係のファイル ●●.c, ●●.h や ●●.cpp を開くデフォルトプログラムとして TeraPad を指定しておくこと、これらのソースファイルやヘッダファイルをダブルクリックするだけで、TeraPad が起動するようにできます。これには、例えば algorithm フォルダを開けた状態で、hello.c を右クリックして「プログラムから開く」→「別のプログラムを開く」→「常にこのアプリを使って.c ファイルを開く」にチェックを入れた状態で、TeraPad を選択して「OK」を押します。これによって、以降は .c ファイルをダブルクリックすると TeraPad を開けるようになります。ヘッダファイル .h やC++言語のソースファイル .cpp についても同様に TeraPad をデフォルトのプログラムとして登録しておくことをお勧めします。

仮にプログラムにミスがあった場合は、コンパイル時、あるいは場合によってはプログラム実行後にエラーが起きますので、エディタ(TeraPad)でソースプログラムを修正して下さい。プログラムのエラーは色々な場合に生じます。ソースプログラムが文法として正しくない場合はもちろん、開こうとしたファイルが現在のフォルダや指定したフォルダ内にない場合や、リンクするオブジェクトプログラムがない場合、PCのメモリが足りなくなった場合など、実に様々です。このため、プログラムエラーが発生したときにどこにバグや問題があるのかを探すデバッグが極めて重要です。コンパイラが出すエラーメッセージに手がかりがある場合もありますが、ない場合も多いです。このデバッグは「慣れるしかありません」。経験を積んで、恐らくここであろう、ということがすぐにピンとくるようになるまでは苦勞することになりますが、それを経ないとプログラミングが上手にはなりません。

説明は以上です。