

C プログラミング研修実習環境セットアップガイド

NEC マネジメントパートナー
人材開発サービス事業部

(1) はじめに

本ガイドの対象コースではプログラミング実習があります。プログラミング実習を実施いただくためには、研修でご利用いただく PC で、本ガイドにて紹介しているソフトウェアが正しく動作することが必要です。お申込み前にご確認ください。

なお、ソフトウェアのインストールについては、PC 環境やセキュリティポリシーに留意いただき、問題がないことを確認のうえ実施いただきますようお願い申し上げます。

対象コース

- LN819 C プログラミング 1ー基本文法ー
- LN820 C プログラミング 2ー実践的プログラム開発技法ー
- LN056 C++ プログラミング基礎

(2) ソフトウェア・研修データ

研修で利用するソフトウェアや研修データは以下の URL からダウンロードできます。研修データは圧縮・暗号化されています。解凍パスワードは研修実施時に別途お伝えします。

●ソフトウェア

以下の URL でダウンロードした圧縮ファイルを任意の場所に解凍してください。

<https://nmpcdata.blob.core.windows.net/download/software.zip>

以下のファイルが含まれています。

コンパイラ	mingw-w64-install.exe
テキストエディタ	tpad108.exe
動作確認用ソースファイル	array.c

●研修データ（サンプルプログラム）

C プログラミング 1	https://nmpcdata.blob.core.windows.net/download/cpro1.zip
C プログラミング 2	https://nmpcdata.blob.core.windows.net/download/cpro2.zip
C++ プログラミング基礎	https://nmpcdata.blob.core.windows.net/download/cpp2.zip

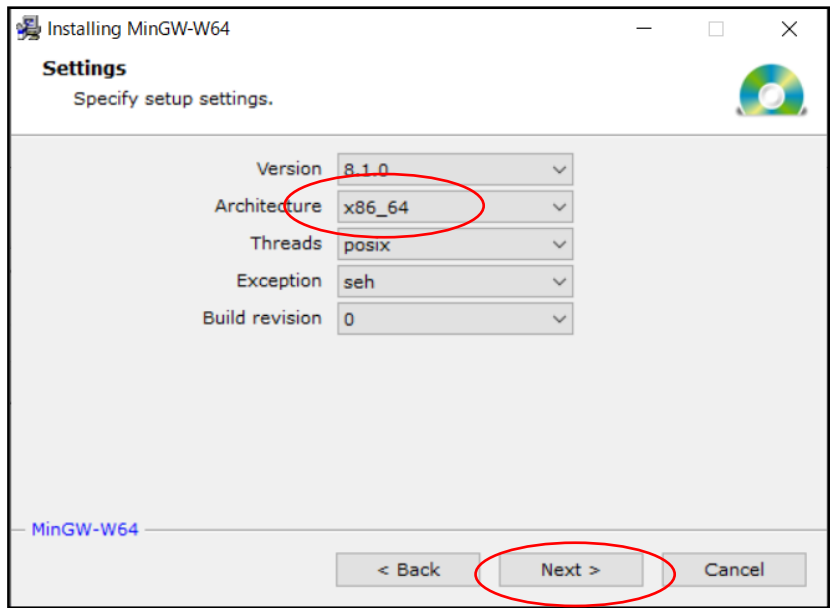
(3) コンパイラ

コンパイラは mingw-w64 を利用します。コースお申込み前に、本コンパイラが正しく動作することをご確認ください。なお、PC の環境によっては本ガイドの通りに動作しない場合がありますので、あらかじめご了承ください。

1. ダウンロードされたファイルを実行します。以下のような画面が表示されますので、「Next」ボタンをクリックします。



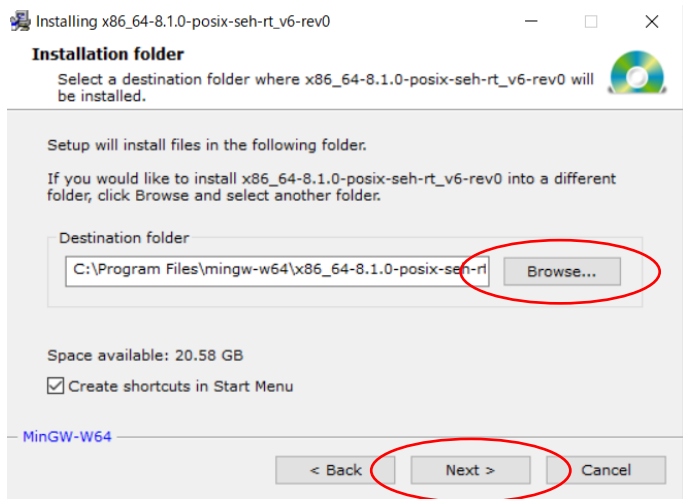
2. インストール設定をします。以下の表のように設定します。



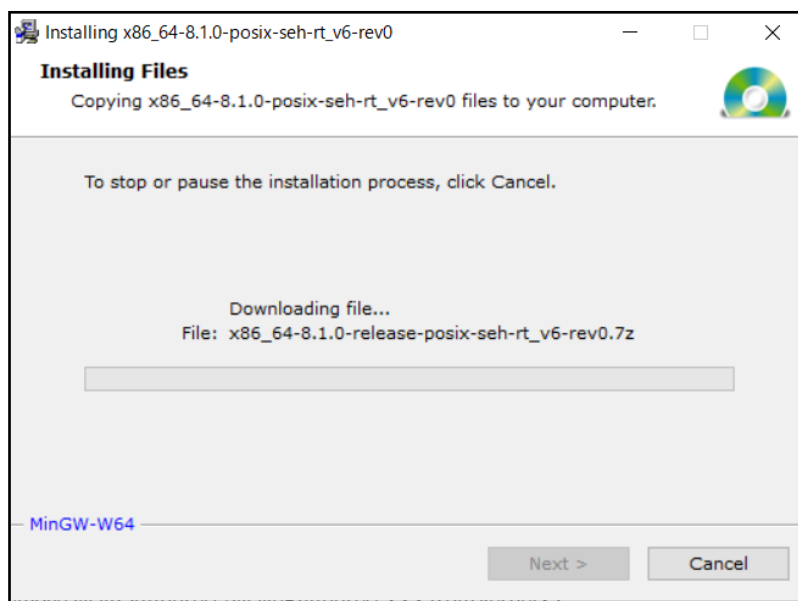
Version	最新バージョンが表示されます。
Architecture	64bit 環境では x86_64 に変更します。
Thread	posix
Exception	seh
Build revision	0

設定したら「Next」ボタンをクリックします。

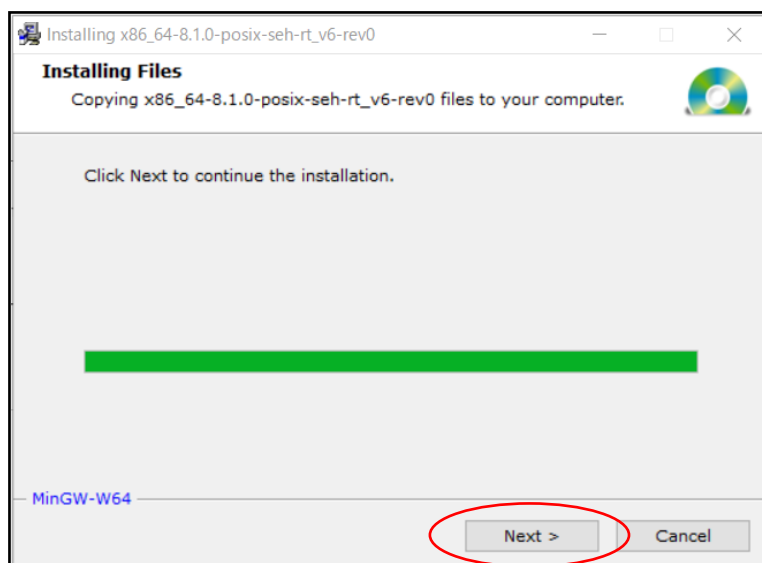
3. 次の画面が表示されるので、必要に応じてインストール先を変更し「Next」ボタンをクリックします。



4. 以下のようにインストールが始まります。



5. インストールが終了したら「Next」ボタンをクリックします。



6. 「Finish」 ボタンをクリックします。



7. コマンドプロンプトを起動します。インストールしたフォルダ内の bin フォルダに移動します。プロンプトに以下のコマンドを入力して、下記画面が表示されていればインストールが成功しています。

`gcc -v`

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17763.1217]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin>gcc -v
Using built-in specs.
COLLECT_GCC=gcc
Target: x86_64-w64-mingw32
Configured with: .././src/gcc-8.1.0/configure --host=x86_64-w64-mingw32 --build=x86_64-w64-mingw32 --target=x86_64-w64-mingw32 --prefix=/mingw64 --with-sysroot=/c:/mingw64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64 --enable-shared --enable-static --disable-multilib --enable-languages=c,c++,fortran,lto --enable-libstdc++-time=yes --enable-threads=posix --enable-libgomp --enable-libatomic --enable-lto --enable-graphite --enable-checking=release --enable-fully-dynamic-string --enable-version-specific-runtime-libs --disable-libstdc++-pch --disable-libstdc++-debug --enable-bootstrap --disable-rpath --disable-win32-registry --disable-nls --disable-werror --disable-symvers --with-gnu-as --with-gnu-ld --with-arch=nocore --with-tune=core2 --with-libiconv --with-system-zlib --with-gmp=/c:/mingw64/prerequisites/x86_64-w64-mingw32-static --with-mpfr=/c:/mingw64/prerequisites/x86_64-w64-mingw32-static --with-mpc=/c:/mingw64/prerequisites/x86_64-w64-mingw32-static --with-isl=/c:/mingw64/prerequisites/x86_64-w64-mingw32-static --with-pkgversion='x86_64-posix-seh-rev0, Built by MinGW-W64 project' --with-bugurl=https://sourceforge.net/projects/mingw-w64 CFLAGS='-O2 -pipe -fno-ident -I/c:/mingw64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c:/mingw64/prerequisites/x86_64-zlib-static/include -I/c:/mingw64/prerequisites/x86_64-w64-mingw32-static/include' CXXFLAGS='-O2 -pipe -fno-ident -I/c:/mingw64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c:/mingw64/prerequisites/x86_64-zlib-static/include -I/c:/mingw64/prerequisites/x86_64-w64-mingw32-static/include' LDFLAGS='-pipe -fno-ident -L/c:/mingw64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/opt/lib -L/c:/mingw64/prerequisites/x86_64-zlib-static/lib -L/c:/mingw64/prerequisites/x86_64-w64-mingw32-static/lib'
Thread model: posix
gcc version 8.1.0 (x86_64-posix-seh-rev0, Built by MinGW-W64 project)

C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin>
```

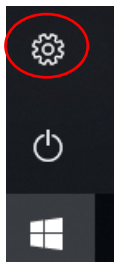
※ コマンドプロンプトはWindows メニューの「Windows システムツール」から選択できます。

（４）環境変数の設定

環境変数の PATH にインストールしたフォルダ内の bin フォルダを登録すると、コマンドプロンプトでは、インストールしたフォルダ内の bin フォルダに移動しなくても実行することができます。

環境変数の変更方法は PC の状況により異なります。以下に設定例を紹介しますが、ご利用の環境にあった手順については、関連の書籍や Web サイトからご確認ください。

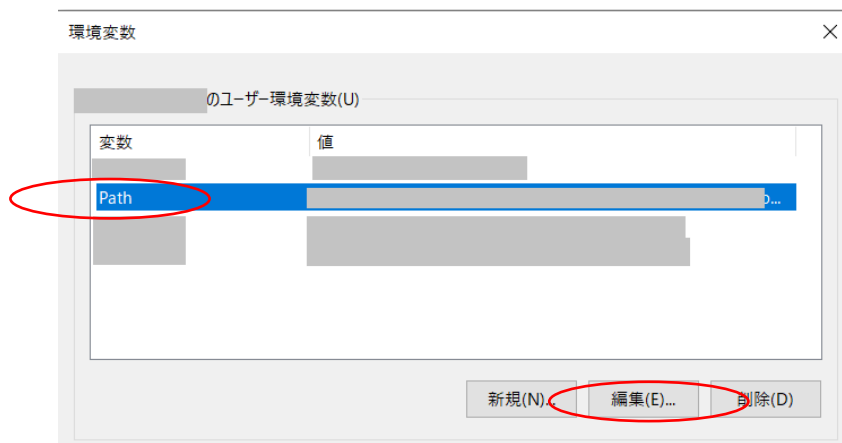
1. Windows メニューの設定ボタンを選択します。



2. Windows の設定画面で「環境変数」と入力し、「環境変数を編集」を選択します。

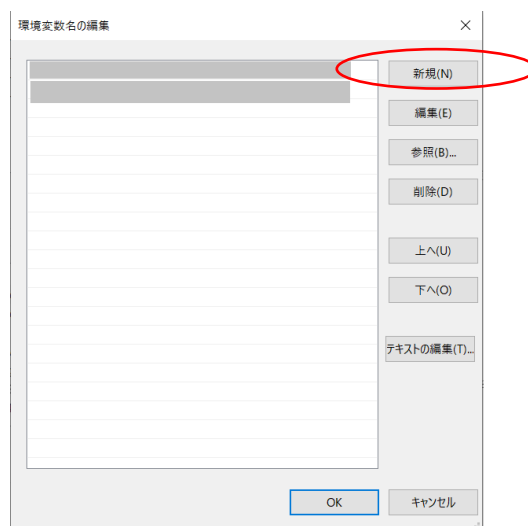


3. ユーザー環境変数から「Path」を選択します、編集ボタンをクリックします。

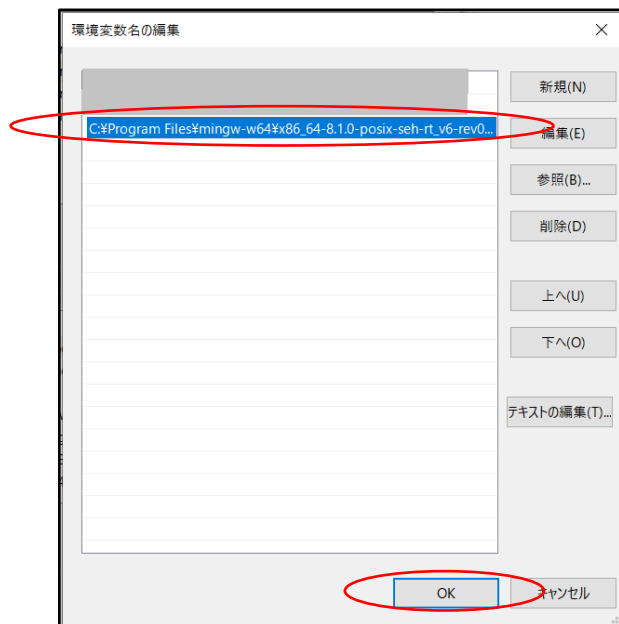


(※ 「Path」が項目にない場合は、補足1に進んでください。)

4. 新規ボタンをクリックします。

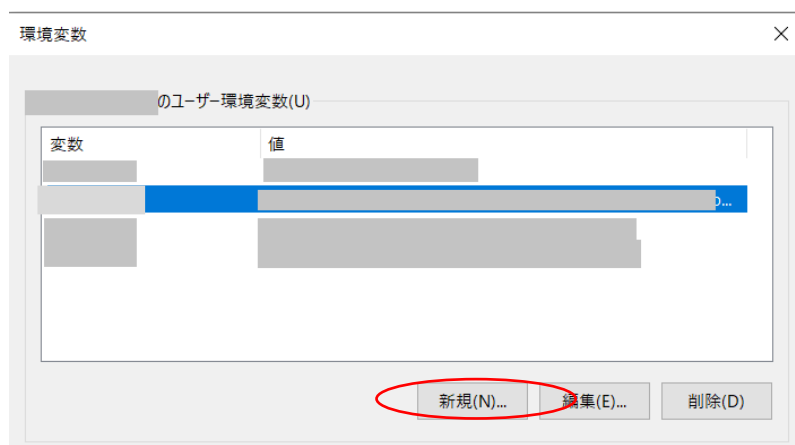


5. インストールしたフォルダ内の bin フォルダ (例: C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin) を入力し、OK ボタンをクリックします。



6. 元の環境変数の画面に戻るので OK ボタンをクリックすれば設定終了です。

補足 1. 上記 3 で、「Path」が無い場合は、新規ボタンをクリックします。



補足 2. 以下の画面が表示されるので、変数名に「Path」を変数値にインストールしたフォルダ内の bin フォルダ（例：C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin）を入力し、



補足 3. 元の環境変数の画面に戻るので OK ボタンをクリックすれば設定終了です

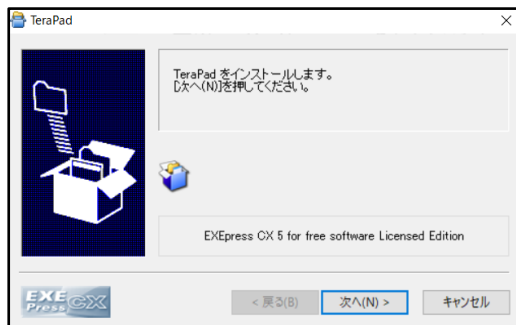
（５）テキストエディタ

テキストエディタは、TeraPad を利用します。

１．インストールファイルの実行

tpad109.exe を実行します。

２．インストールウィザードが起動します。



表示される画面すべてで「次へ」をクリックします。最後に、TeraPad が起動されて、「はじめに」が表示されるので、読んだ後に終了します。

(6) 動作確認

1. 「array1.c」を任意のフォルダにコピーします。
2. コマンドプロンプトを起動します。コマンドプロンプトはWindows メニューの「Windows システムツール」から選択できます。
3. CD コマンドを利用して、(1) の「array1.c」がコピーされてフォルダに移動します。
例) コピー先が、C:\work の場合以下のように入力します。

```
>CD C:\work
```

4. 「gcc -v」と入力し、以下のようなメッセージが表示されることを確認します。

```
>gcc -v
Using built-in specs.
COLLECT_GCC=gcc
( . . . 省略 . . . )
```

5. 以下のコマンドを入力し、コンパイル、実行ができるかを確認します。

```
>gcc -o array1 array1.c    ※コンパイル
```

```
>array1
```

```
No.1 ... 1                ※実行 (「数字を入力して Enter」を 5 回繰り返してください)
```

```
No.2 ... 2
```

```
No.3 ... 3
```

```
No.4 ... 4
```

```
No.5 ... 5
```

```
No.1 : 1
```

```
No.2 : 2
```

```
No.3 : 3
```

```
No.4 : 4
```

```
No.5 : 5
```

```
Total = 15
```