VSCode環境構築

執筆者 : 59 期生　片野 凱介

初稿 : 2022 年 5 月 2 日

第一改稿 : 2022 年 5 月 6 日

第二改稿 : 2022 年 5 月 25 日

目次

0. 事前準備(任意)

拡張機能(Japanese Language Pack for Visual Studio Code) のインストール

- 英語でも良い人は飛ばして構わない。

手順

- 起動時に聞かれるので大人しくクリックしておしまい。

  勝手にソフトの再起動までしてくれる。

HackGen のインストール

- 標準のフォントでも良い人は飛ばして構わない。

手順

1. こちら からダウンロード。
2. ダウンロードしたものを解凍。
3. 以下の 2 種類(合計 4 ファイル)のみを抽出(それ以外消していい)。
    i. HackGen Console
    ii. HackGen
4. エクスプローラーで C:\Windows\Fonts を開き、先程抽出した 4 ファイルをペースト。
5. 自動でインストールされるのでインストールが終わるまで待つ。

    できたら、2 種類のフォルダ？（項目？）ができており、

    その中にそれぞれの Regular 版 と Blod 版 が入っているはず。

- 以上。解凍したファイルは消しても良いが、起動する度にリセットされるので、残しておいたほうが吉。

1. VSCode に拡張機能のインストール

拡張機能(C/C++)のインストール

- 拡張機能検索のところで C/C++ 入れて検索、一番上のインストール。
- プレリリースバージョン は不要。

拡張機能(C/C++ Extension Pack)のインストール

- 一応入れる。方法は同上。

2. settings.json の編集(フォントなどの設定)(任意)

- 設定を変更しない人や、フォントを変えない人は飛ばして構わない。

手順

1. settings.json を開く(方法は後述)。
2. 以下を記述。
    - なお、　// から始まる コメントアウト は記述しなくても良い。
    - また、不要な行は　記述しない　又は、　コメントアウト　を各自で行う。

```
    // 起動時の情報を非表示にする
    "workbench.startupEditor": "none",
    // 一般的な(コードなど)フォントの指定
    "editor.fontFamily": "HackGen, monospace",
    // コンソールのフォントの指定
    "debug.console.fontFamily": "HackGenConsole",
    // フォントサイズの設定
    "editor.fontSize": 24,
    // スペース可視化
    "editor.renderWhitespace": "boundary",
```

settings.json の開き方

1. VSCode を開き、左上のファイルを開く(又は、 Alt → F)。
2. プルダウンメニューが展開されるので、ユーザー設定 → 設定 を開く。
3. 右上の ウィンドウを閉じるボタン の下にある、3 つのボタンのうち、

   一番左のボタン(紙を裏返すようなアイコン)をクリック。

3. コンパイル(gcc の実行)

- 私が、VSCode 上でコンパイルできる方法を確立させたので、処理 B は不要になりました。

  よって、処理 C を実行してください。
  - 私が、コンパイル実行環境を完成させたので、処理 A は不要になりました。

    ~~よって、処理 B を実行してください。~~
    - ~~処理 A を実行してください。~~

処理 C

- tasks.json を下記に書き換えてください。

  なお、tasks.json がない場合は後述するいずれかの方法で作成してください。

```
{
        "version": "2.0.0",
        "tasks": [
                {
                        "type": "cppbuild",
                        "label": "C/C++: cl.exe アクティブなファイルのビルド",
                        "command": "C:/Xilinx/14.7/ISE_DS/ISE/gnu/MinGW/Beta/nt64/bin/gcc.exe",
                        "args": [
                                "-g",
                                "-o" ,
                                "${fileDirname}\\${fileBasenameNoExtension}.exe",
                                "${file}"
                        ],
                        "options": {
                                "cwd": "${fileDirname}"
                        },
                        "problemMatcher": [
                                "$msCompile"
                        ],
                        "group": "build",
                        "detail": "コンパイラ: cl.exe"
                }
        ]
}
```

お手軽 tasks.json 作成(推奨)

1. VSCode で、コンパイルしたいプログラムを開く。

   例) HelloWorld.c を開く。
2. Ctrl+Shift+P を押す。
3. タスク:タスクの構成 を選択。
4. C/C++:cl.exe アクティブなファイルのビルド を選択。
5. tasks.json が自動で生成され、開かれる。

自力で tasks.json 作成

1. プログラムがある階層に、 .vscode ファイルを作成。
2. .vscode に tasks.json という名前の空ファイルを作成。

処理 B (不要)

- **現在、この処理は不要です。**

手順

1. 配布した Compiler.bat をクリックして起動してください。
2. 指示に従って実行してください。

念のため処理を記述しておきます。

```
1.起動後、コンパイルしたいファイルのファイル名(拡張子無し)　を入力し、Enter。
2.そのファイルの、絶対ディレクトリ(ファイル)パス　を入力し、Enter。
3.何かキーを押したら、ウィンドウが閉じる。
```

処理 A (不要)

- **現在、この処理は不要です。**

手順

- コマンドプロンプトを起動し、以下を実行。

```
cd プログラムのディレクトリパス(ファイルパス)
```

- その後以下を実行。

```
C:\Xilinx\14.7\ISE_DS\ISE\gnu\MinGW\Beta\nt64\bin\gcc.exe -g -o プログラム名 プログラム名.c
```