

Εγκατάσταση προγραμματιστικού περιβάλλοντος ανάπτυξης εφαρμογών για C και C++ σε Windows

VS Code και Mingw-w64

Χρήστος Γκόγκος, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Π.Ι.(Άρτα)

30/9/2019

Οι παρακάτω οδηγίες διαμορφώνουν το **Visual Studio Code** έτσι ώστε να χρησιμοποιεί τον **GCC compiler** για **C** και **C++** με σκοπό τη δημιουργία προγραμμάτων που μεταγλωττίζονται και εκτελούνται σε περιβάλλον **Windows**.

Εγκαταστάσεις λογισμικών

- Εγκατάσταση του [mingw-w64](#) σε έναν κατάλογο που δεν έχει κενά στη διαδρομή του (όχι στην προκαθορισμένη διαδρομή που προτείνεται κατά την εγκατάσταση **c:\Program Files**). Για παράδειγμα μπορεί να εγκατασταθεί στο **c:\mingw-w64**. Αν ο υπολογιστής είναι **64-bit** τότε επιλέξτε κατά την εγκατάσταση ως **architecture** το **x86_64**, αλλιώς αν είναι **32-bit** επιλέξτε **i686**.
- Προσθήκη της διαδρομής του καταλόγου **bin** του **Mingw-w64** (π.χ. του **c:\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin**) στη μεταβλητή περιβάλλοντος (environment variable) **PATH** των Windows [2].
- Εγκατάσταση του [Visual Studio Code](#).
- Εγκατάσταση της επέκτασης [C++ extension for VS Code](#).
- Εγκατάσταση της επέκτασης [Code Runner](#).
 - Πιέστε **Ctrl+,** (πλήκτρο Control και πλήκτρο κόμμα) έτσι ώστε να εμφανιστεί η καρτέλα **Settings** (εναλλακτικά η καρτέλα **Settings** εμφανίζεται επιλέγοντας τα μενού **File > Preferences > Settings**). Στα **extensions** εντοπίστε το **Run Code Configuration** και επιλέξτε το **Run In Terminal** και το **Save All Files Before Run**.

Δημιουργία ενός χώρου εργασίας (workspace)

- Σε ένα νέο παράθυρο γραμμής εντολών, δημιουργήστε έναν άδειο κατάλογο **projects** στον οποίο θα τοποθετηθούν όλα τα VS Code projects. Στη συνέχεια δημιουργήστε έναν υποκατάλογο με όνομα **helloworld**. Οι εντολές που θα χρειαστεί να δοθούν είναι οι ακόλουθες:

```
mkdir projects
cd projects
mkdir helloworld
cd helloworld
```

- Ανοίξτε τον φάκελο **helloworld** με το **VS Code** (το **VS Code** έχει τη δυνατότητα εκτός από αρχεία να ανοίγει και καταλόγους οπότε απεικονίζει όλα τα αρχεία και τους υποκαταλόγους που περιέχονται).

Προσθήκη πηγαίου κώδικα σε C

- Στο VS Code επιλέξτε το μενού **File > New File** και ονομάστε το αρχείο **hello1.c**.
- Γράψτε τον ακόλουθο πηγαίο κώδικα στο αρχείο **hello1.c**:

```
#include <stdio.h>
```

```
int main()
{
    printf("hello, world!\n");
    return 0;
}
```

- Πραγματοποιήστε αυτόματη μορφοποίηση του κώδικα (**Ctrl+Alt+F**).
- Αποθηκεύστε (**Ctrl+S**) και εκτελέστε (**Ctrl+Alt+N**) τον κώδικα.
- Το αποτέλεσμα θα πρέπει να είναι:

```
hello, world!
```

Προσθήκη πηγαίου κώδικα σε C++

- Στο VS Code επιλέξτε το μενού **File > New File** και ονομάστε το αρχείο **hello2.cpp**.
- Γράψτε τον ακόλουθο πηγαίο κώδικα στο αρχείο **hello2.cpp**:

```
#include <iostream>
#include <vector>

int main()
{
    std::vector<std::string> msg{"hello", " ", "world!"};
    for (auto s : msg)
    {
        std::cout << s;
    }
    std::cout << std::endl;
    return 0;
}
```

- Πραγματοποιήστε αυτόματη μορφοποίηση του κώδικα (**Ctrl+Alt+F**).
- Αποθηκεύστε (**Ctrl+S**) και εκτελέστε (**Ctrl+Alt+N**) τον κώδικα.
- Το αποτέλεσμα θα πρέπει να είναι:

```
hello, world!
```

Αναφορές

1. [Using Mingw-w64 in VS Code](#)
2. [How to set the path and environment variables in Windows](#)
3. [Principles of Algorithmic Problem Solving, Johan Sannemo, 2018](#)