**GCC C-MAKE FRAMEWORK**

Tác giả : Long Dao | <https://louisvn.com>

1. **Giới thiệu**

* GCC C-Make Framework được thiết kế để quản lý các dự án phần mềm C/C++.
* Cho phép người dùng tạo, xóa, biên dịch, chạy và xuất báo cáo thử nghiệm cho một dự án bất kỳ.
* Để quản lý được một dự án, tôi cần bạn cài đặt một số phần mềm sau để sử dụng framework.
  + [**Toolchain GCC**](https://winlibs.com/). Đây là một trình biên dịch đảm nhiệm quá trình dịch từ mã C sang mã máy để chương trình bạn viết ra chạy được trên PC của bạn. Hãy tải xuống phiên bản **GCC 13.2.0 (with POSIX threads - UCRT) 32bit** để thống nhất về mặt môi trường.
  + Makefile. Đây là công cụ thay thế cho việc nhập liệu các câu lệnh phức tạp thành các câu lệnh đơn giản trên cửa sổ điều khiển Terminal của PC.
  + [**Cygwin**](https://www.cygwin.com/install.html). Nếu bạn là người dùng hệ điều hành Window, bạn cần cài đặt công cụ này để giả lập một môi trường Linux giúp cho Makefile và các công cụ khác hoạt động ổn định.
  + [**VSCode**](https://code.visualstudio.com/download). Đây là phần mềm soạn thảo văn bản cực kỳ phổ biến khi bạn viết code. Nó giúp bạn hiển thị màu sắc, nhắc lệnh khi viết, hỗ trợ cấu hình gỡ lỗi, …
  + [**Python**](https://www.python.org/downloads/). Trong Python, một công cụ [**Gcovr**](https://gcovr.com/en/stable/) cho phép bạn tạo bản báo cáo về độ bao phủ của mã nguồn (code coverage).

1. **Cài đặt GCC**

* Để cài đặt GCC, hãy truy cập đường dẫn sau (Github): [**Nhấn vào đây**](https://github.com/brechtsanders/winlibs_mingw/releases/download/13.2.0-16.0.6-11.0.0-ucrt-r1/winlibs-i686-posix-dwarf-gcc-13.2.0-llvm-16.0.6-mingw-w64ucrt-11.0.0-r1.zip)
* Sau khi tải xong, nhấn **Window + E**. Tại cửa sổ Quản lý tệp, vào ổ **C**, tạo một folder là “**Toolchain**”.
* Sau đó giải nén file vừa tải xuống vào folder vừa tạo. Sau khi giải nén, bạn sẽ có một đường dẫn : **C:\Toolchain\mingw32**
* Nhấn chọn **Window** và tìm kiếm “**environment**”. Bạn sẽ thấy ứng dụng sau:



* Nhấn chọn ứng dụng. Chọn Tab **Advanced** > **Environment variables…**
* Một cửa sổ mới hiện lên. Tại bảng **System variables** > **Path** > **Edit…**
* Một cửa sổ mới hiện lên. Nhấn **New** để thêm mới các đường dẫn sau:
  + **C:\Toolchain\mingw32\bin**
  + **C:\Toolchain\mingw32\i686-w64-mingw32\bin**
* Sau khi thêm xong, hãy nhấn **OK** lần lượt để thoát.
* Xong! Để kiểm tra xem bạn đã cài đặt thành công hay chưa, hãy nhấn **Window** và tìm kiếm “**cmd**”



* Mở ứng dụng, nhập “**gcc -v**” vào cửa sổ. Nếu bạn thấy thông báo như sau là đã thành công.



1. **Cài đặt VSCode**

* Để cài đặt VSCode, hãy truy cập đường dẫn sau: [**Nhấn vào đây**](https://code.visualstudio.com/download)
* Sau khi tải xuống, nhấn mở file setup. Chọn “**I accept the agreement**” > Chọn “**Create a desktop icon**” > **Next** > **Install**
* Đợi cài đặt xong là đã thành công!
* Tiếp theo, bạn cần mở ứng dụng VSCode vừa cài lên để cài đặt các gói công cụ cần thiết.
* Nhấn tổ hợp phím **Ctrl + Shift + X**. Một tab “Extensions” hiện lên.
* Gõ vào thanh tìm kiếm và tải xuống lần lượt các gói sau:
  + **C/C++** ; **C/C++ Themes** ; **C/C++ Extension Pack**
  + **Code Runner**
  + **Doxygen Documentation Generator**

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

* Sau khi cài đặt xong, nhấn tổ hợp phím [ **Ctrl + ,** ] để mở tab “Settings”.
* Nhập “**Code-runner: Run In Terminal**” vào thanh tìm kiếm. Sau đó nhấn chọn vào ô vuông theo hình bên dưới:



1. **Cài đặt Cygwin**

* Để cài đặt Cygwin, hãy truy cập đường dẫn sau: [**Nhấn vào đây**](https://www.cygwin.com/setup-x86_64.exe)
* Giống như bước 2 của cài đặt GCC, Bạn vào ổ **C**, tạo một folder là “**Cygwin**”.
* Tiếp theo, copy file setup vừa tải xuống vào folder vừa tạo: **C:\Cygwin**
* Nhấp mở ứng dụng vừa được copy. Chọn **Next** > **Install from Internet**
* Tại cửa sổ mới, nhập “**C:\Cygwin**” vào “**Root Directory**”. Phía dưới chọn **All Users** > **Next**
* Tại cửa sổ mới, nhập “**C:\Cygwin**” vào “**Local Package Directory**” > **Next**
* Chọn Use System Proxy **Settings** > **Next**
* Chọn vào 1 đường link bất kỳ để tải xuống dữ liệu. (Nên chọn link thứ 4) > **Next**
* Đợi 1 lúc đến khi một cửa sổ “**Select Packages**” hiện ra. Tại **View** chọn **Full**
* Search các gói sau và chọn các phiên bản mới nhất : **make**, **zip**, **unzip**, **sed**, **cygrunsrv, bc, tree**

Lưu ý, nhấn vào ô vuông có mũi tên trỏ xuống để lựa chọn phiên bản. Ví dụ như hình sau:

A screenshot of a computer

Description automatically generated with medium confidence

* Sau khi chọn xong phiên bản cài đặt cho tất cả các gói được liệt kê phía trên, nhấn **Next** > **Next** để tiến hành cài đặt.
* Chờ đến khi cài đặt xong, nhấn **Finish**.
* Sau đó, bạn mở “**Environments**” tương tự như bước 4+ của cài đặt GCC và thêm vào các đường dẫn sau:
  + **C:\Cygwin\bin**
  + **C:\Cygwin\sbin**
  + **C:\Cygwin\usr\sbin**
* Sau khi thêm Environments, bạn **kích chuột phải** vào biểu tượng ứng dụng vừa cài đặt. Chọn “**Run as administrator**” và chờ cho ứng dụng dụng hiện lên.
* Nhập vào “**cygserver-config**” > **Enter** >… nhập “**yes**”. Chờ chạy xong và đóng cửa sổ.
* **Lưu ý** : Nếu bạn có cài đặt Git trên PC. Vui lòng di chuyển các đường dẫn của Cygwin lên phía trên các đường dẫn của Git trong Enviroments.
* Khởi động lại PC của bạn - bắt buộc.

1. **Cài đặt Python**

* Để cài đặt Python, hãy truy cập đường dẫn sau: [**Nhấn vào đây**](https://www.python.org/downloads/)
* Nhấn vào nút “**Download Python …**” để tải xuống ứng dụng.
* Nhấn **Window + E**. Tại cửa sổ Quản lý tệp, vào ổ **C**, tạo các folder lồng nhau như sau: “**Toolchain > Python > Python311**”. (Thay “311” thành phiên bản của Python của bạn).
* Sau khi tải xong, Click mở ứng dụng. Tại cửa sổ cài đặt, chọn “**Customize installation**”. Tiếp theo nhấn chọn **tất cả** các lựa chọn trong “**Optional Features**”, đặc biệt là “**Pip**”. Sau đó nhấn “**Next**”.
* Tại cửa sổ “**Advanced Options**” cũng nhấn chọn tất cả các lựa chọn. Sau đó tại mục “**Customize install location**”, nhập vào đường dẫn đến folder đã tạo trước đó “**C:\Toolchain\ Python\ Python311**” rồi chọn “**Install**”.
* Chờ việc cài đặt hoàn tất và nhấn “**Close**”.
* Sau khi cài đặt xong, mở “**Path**” trong “**Eviroments**”. Nếu bạn không thấy 2 đường dẫn tương tự được đặt ở trên cùng (như hình bên dưới), hãy copy đường dẫn chính xác trên PC của bạn và thêm nó vào (di chuyển lên trên cùng).

A screenshot of a computer program

Description automatically generated

* Sau đó, mở “**cmd**” và nhập “**python --version**” để kiểm tra phiên bản của Python. Nếu có thông tin phiên bản được in ra tức là bạn đã cài đặt thành công.

A black text on a white background

Description automatically generated

* Tiếp theo, tại cửa sổ “**cmd**”, nhập “**pip install gcovr**” để cài đặt công cụ **Gcovr**. Chờ đến khi công cụ được cài đặt thành công, bạn sẽ thấy dòng chữ “Successfully installed gcovr-…”. Nhập “**gcovr --version**” để kiểm tra phiên bản của nó.



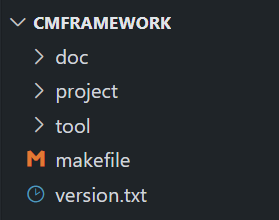
* Lưu ý. Phiên bản **Python 3.11.4** và phiên bản **Gcovr 6.0** đang chạy ổn định trên Framework này.

1. **Sử dụng Framework**

* Bạn cần tải và giải nén Framework trước khi làm việc với nó.
* Để sử dụng Framework, trước tiên hãy mở trình soạn thảo **VSCode** lên.
* Để thêm Framework vào trình soạn thảo, nhấn liên tục 2 tổ hợp **Ctrl + K** và **Ctrl + O**.
* Một cửa sổ “**Open Folder**” hiện lên, chọn đường dẫn đến folder “**CMFramework**” đã được giải nén trước đó. Sau đó chọn **Select Folder**
* [**Nếu**] một cửa sổ tương tự hình bên dưới hiện ra, hãy **nhấn chọn** vào ô vuông và nhấn “**Yes**”.



* Bạn sẽ nhìn thấy ở tab “**Explorer**” hiển thị toàn bộ Framework vừa được thêm vào.
* **CẤU TRÚC FRAMEWORK**



* + **doc** : lưu trữ tài liệu sử dụng chung cho mọi dự án. File “**Readme.vi.docx**” này cũng nằm trong thư mục này.
  + **project** : quản lý tất cả các dự án của bạn. Lưu ý, tôi đã cung cấp sẵn một dự án mẫu “**~temp**” làm nền tảng để tạo ra các dự án khác. Bạn buộc không được xóa nó bằng mọi giá.
  + **tool** : bao gồm các thư viện chung và các công cụ tạo báo cáo thử nghiệm, …
  + **makefile** : đây là tệp quản lý toàn bộ các tính năng xương sống nhằm đảm bảo Framework hoạt động. Bạn cũng **không** được tùy chỉnh file này ngoại trừ giá trị của các biến bên trong mục “**Setting**”.
  + **version.txt** : file này ghi lại lịch sử các phiên bản và phiên bản hiện tại của Framework.
* **CẤU TRÚC CỦA PROJECT TRONG FRAMEWORK**



Lấy ví dụ trên dự án mẫu “**~temp**” được cung cấp sẵn. Lưu ý bạn **không** được chỉnh sửa hay xóa bỏ dự án này. Bạn cần tạo dự án mới và thực thi trên nó.

* + **doc** : lưu trữ các tài liệu của dự án này.
  + **inc** : chứa các file header của dự án (**.h**)
  + **dev** : chứa các file source của dự án được phát triển để thử nghiệm về độ bao phủ của mã (**.c .cpp**)
  + **src** : chứa các file source của dự án (**.c .cpp .o**)
  + **user\_cfg.mk** : một makefile cho phép người dùng cấu hình đường dẫn hoặc cấu hình cho từng dự án cụ thể.
* **SỬ DỤNG CÁC LỆNH MAKE**

Trước tiên, mở một cửa sổ Terminal trên VSCode. Vào **Terminal** > **New Terminal**

A screenshot of a phone

Description automatically generated with medium confidence

Tại tab **TERMINAL**, nhập các yêu cầu của bạn. Lưu ý, đường dẫn gốc trên terminal phải là đường dẫn đến “**CMFramework**”. Nếu không đúng, sử dụng lệnh “**cd**” để di chuyển đến.

* + **make setup** : khởi tạo các tool, câu lệnh này nên chạy 1 lần khi bạn mới sử dụng Framework này lần đầu. Nó sẽ tạo ra các file thực thi bên trong thư mục “**tool/bin**”.
  + **make** | **make all** : 2 câu lệnh này cho phép chạy {**make clean build run**}, xem cụ thể bên dưới.
  + **make clean** : dùng để xóa thư mục đầu ra, nơi lưu trữ các file object và file thực thi. Nó cũng xóa các bản báo cáo thử nghiệm (nếu có).
  + **make build** : dùng để biên dịch các file (**.c**) sang các file object (**.o**), sau đó liên kết nó thành file thực thi (**.exe**).
  + **make merge** : dùng để gộp các file object lại thành một file object duy nhất.
  + **make run** : dùng để chạy file thực thi (**.exe**) trên terminal.
  + **make report** : dùng để tạo bản báo cáo thử nghiệm nếu bạn sử dụng thư viện “**utest.h**” (sẵn trong **tool**) trong chương trình C để viết testcase.
  + **make pack** : dùng để đóng gói dự án, cho phép bạn chia sẻ nó cho bất kỳ ai.
  + **make vsinit** : dùng để tạo ra các file cấu hình trong VSCode, giúp phần mềm liên kết đúng các đường dẫn file trên **giao diện** hiển thị. Khi bạn move sang một project khác, lệnh này sẽ **tự động** được chạy. Bạn chỉ cần chạy lại lệnh này khi bạn **update** các đường dẫn trong makefile (**user\_cfg.mk**).
  + **make move** : dùng để di chuyển đến một dự án bất kỳ. Tự động thêm mới nếu dự án không tồn tại.
    - Ví dụ : **make move.proj1** : sẽ hiểu là di chuyển tới dự án “**proj1**”.
    - Hoặc : **make move.group1:proj1** : di chuyển đến dự án “**group1/proj1**”. Bạn có thể lồng trong nhiều group, sử dụng dấu “**:**” để phân cách.
  + **make remove** : dùng để xóa bỏ một dự án bất kỳ. Nếu xóa dự án hiện tại, nó sẽ tự chuyển về dự án mẫu “**~temp**”. Cách dùng giống như lệnh “**move**”.
  + **make import** : dùng để thêm một dự án mới từ một file zip được chia sẻ từ lệnh “**pack**” của người khác. Cách dùng giống như lệnh “**move**”. Tuy nhiên, biến “ZIP” sẽ cho biết đường dẫn đến file zip.
    - Ví dụ : **make** **import.proj1 ZIP=path/to/file.zip**
    - Lưu ý : biến “ZIP” phải viết liền với dấu “=” và đường dẫn. Không sử dụng khoảng trắng ở giữa.
  + **make list** : dùng để liệt kê tất cả các file và folder đang có trong dự án.
  + **make print** : dùng để in ra giá trị của các biến được sử dụng bên trong makefile.
    - Ví dụ : **make print.VAR1.VAR2** : in ra giá trị của 2 biến là **VAR1** và **VAR2**.
    - Tương tự nếu in nhiều biến sẽ phân cách nhau bởi dấu chấm.
    - Câu lệnh này chỉ sử dụng cho Admin hoặc nhà phát triển Framework.
* **LƯU Ý**
  + Tất cả các đường dẫn, tên file đều phải được viết liền. Tức là không có khoẳng trắng ở giữa. Bạn nên lưu ý khi đặt tên cho bất kỳ một tệp hoặc thư mục nào.
  + Không được phép đặt tên các source file hay header file trùng với tên của dự án. Điều này có thể xảy ra xung đột khi tạo các file đầu ra.
  + Trong một dự án, mỗi một tên file là duy nhất ở toàn bộ các thư mục. Bạn không được phép biên dịch 2 tên file giống nhau vì nó cũng có thể xảy ra xung đột tại đầu ra.

===================== **HẾT** =====================

Tác giả : Long Dao | <https://louisvn.com>