# GIỚI THIỆU

## Giới thiệu về C

**Lập trình C** là ngôn ngữ lập trình máy tính mệnh lệnh, có mục đích chung, được phát triển vào năm 1972 bởi Dennis M. Ritchie tại Phòng thí nghiệm Điện thoại Bell để phát triển hệ điều hành UNIX. C là ngôn ngữ máy tính được sử dụng rộng rãi nhất. Nó tiếp tục dao động ở mức độ phổ biến số một cùng với ngôn ngữ lập trình Java, cũng phổ biến không kém và được sử dụng rộng rãi nhất trong số các lập trình viên phần mềm hiện đại.

### Tại sao nên học lập trình C?

**Ngôn ngữ lập trình C** là PHẢI dành cho sinh viên và các chuyên gia đang đi làm để trở thành một Kỹ sư phần mềm tuyệt vời, đặc biệt khi họ đang làm việc trong Miền phát triển phần mềm. Tôi sẽ liệt kê một số ưu điểm chính của việc học lập trình C:

* Dễ học
* Ngôn ngữ có cấu trúc
* Nó tạo ra các chương trình hiệu quả
* Nó có thể xử lý các hoạt động cấp thấp
* Nó có thể được biên dịch trên nhiều nền tảng máy tính

### Sự thật về C

* C được phát minh ra để viết một hệ điều hành gọi là UNIX.
* C là sự kế thừa của ngôn ngữ B được giới thiệu vào khoảng đầu những năm 1970.
* Ngôn ngữ này được chính thức hóa vào năm 1988 bởi Viện Tiêu chuẩn Quốc gia Hoa Kỳ (ANSI).
* Hệ điều hành UNIX được viết hoàn toàn bằng C.
* Ngày nay C là Ngôn ngữ lập trình hệ thống được sử dụng rộng rãi và phổ biến nhất.
* Hầu hết các phần mềm hiện đại đã được triển khai bằng C.

### Ứng dụng của lập trình C

Ban đầu C được sử dụng cho công việc phát triển hệ thống, đặc biệt là các chương trình tạo nên hệ điều hành. C đã được sử dụng làm ngôn ngữ phát triển hệ thống vì nó tạo ra mã chạy nhanh gần bằng mã được viết bằng hợp ngữ. Một số ví dụ về việc sử dụng C là -

* Các hệ điều hành
* Trình biên dịch ngôn ngữ
* thợ lắp ráp
* soạn thảo văn bản
* Bộ đệm in
* trình điều khiển mạng
* Chương trình hiện đại
* cơ sở dữ liệu
* Thông dịch viên ngôn ngữ
* tiện ích

## Cài đặt môi trường để chạy

Chương trình AC có thể dài từ 3 dòng đến hàng triệu dòng và phải được ghi vào một hoặc nhiều tệp văn bản có phần mở rộng là **".c"** ; ví dụ: *xin chào. c* . Bạn có thể sử dụng **"vi"** , **"vim"** hoặc bất kỳ trình soạn thảo văn bản nào khác để viết chương trình C của bạn vào một tệp.

Hướng dẫn này giả định rằng bạn biết cách chỉnh sửa tệp văn bản và cách viết mã nguồn bên trong tệp chương trình.

Nếu bạn muốn thiết lập môi trường cho ngôn ngữ lập trình C, bạn cần có sẵn hai công cụ phần mềm sau trên máy tính của mình, (a) Trình soạn thảo văn bản và (b) Trình biên dịch C.

### Soạn thảo văn bản

Điều này sẽ được sử dụng để gõ chương trình của bạn. Ví dụ về một số trình soạn thảo bao gồm Windows Notepad, lệnh Chỉnh sửa hệ điều hành, Tóm tắt, Epsilon, EMACS và vim hoặc vi.

Tên và phiên bản của trình soạn thảo văn bản có thể khác nhau trên các hệ điều hành khác nhau. Ví dụ: Notepad sẽ được sử dụng trên Windows và vim hoặc vi có thể được sử dụng trên windows cũng như trên Linux hoặc UNIX.

Các tệp bạn tạo bằng trình chỉnh sửa của mình được gọi là tệp nguồn và chúng chứa mã nguồn chương trình. Các tệp nguồn cho các chương trình C thường được đặt tên với phần mở rộng " **.c** ".

Trước khi bắt đầu lập trình, hãy đảm bảo rằng bạn có sẵn một trình soạn thảo văn bản và bạn có đủ kinh nghiệm để viết một chương trình máy tính, lưu nó vào một tệp, biên dịch và cuối cùng là thực thi nó.

### Trình biên dịch C

Mã nguồn được viết trong tệp nguồn là nguồn có thể đọc được của con người đối với chương trình của bạn. Nó cần được "biên dịch" thành ngôn ngữ máy để CPU của bạn thực sự có thể thực thi chương trình theo hướng dẫn đã cho.

Trình biên dịch biên dịch mã nguồn thành các chương trình thực thi cuối cùng. Trình biên dịch miễn phí và được sử dụng thường xuyên nhất là trình biên dịch GNU C/C++, nếu không, bạn có thể có các trình biên dịch từ HP hoặc Solaris nếu bạn có các hệ điều hành tương ứng.

Phần sau đây giải thích cách cài đặt trình biên dịch GNU C/C++ trên các hệ điều hành khác nhau. Chúng tôi tiếp tục đề cập đến C/C++ cùng nhau vì trình biên dịch GNU gcc hoạt động cho cả ngôn ngữ lập trình C và C++.

### Cài đặt trên UNIX/Linux

Nếu bạn đang sử dụng **Linux hoặc UNIX** , hãy kiểm tra xem GCC đã được cài đặt trên hệ thống của bạn hay chưa bằng cách nhập lệnh sau từ dòng lệnh –

$ gcc -v

Nếu bạn đã cài đặt trình biên dịch GNU trên máy của mình, thì nó sẽ in ra một thông báo như sau -

Using built-in specs.

Target: i386-redhat-linux

Configured with: ../configure --prefix=/usr .......

Thread model: posix

gcc version 4.1.2 20080704 (Red Hat 4.1.2-46)

Nếu GCC chưa được cài đặt, thì bạn sẽ phải tự cài đặt nó bằng hướng dẫn chi tiết có sẵn tại <https://gcc.gnu.org/install/>

Hướng dẫn này đã được viết dựa trên Linux và tất cả các ví dụ đã cho đã được biên soạn dựa trên phiên bản Cent OS của hệ thống Linux.

### Cài đặt trên Mac OS

Nếu bạn sử dụng Mac OS X, cách dễ nhất để lấy GCC là tải xuống môi trường phát triển Xcode từ trang web của Apple và làm theo hướng dẫn cài đặt đơn giản. Khi bạn đã thiết lập Xcode, bạn sẽ có thể sử dụng trình biên dịch GNU cho C/C++.

Xcode hiện có sẵn tại [developer.apple.com/technologies/tools/](https://developer.apple.com/technologies/tools/) .

### Cài đặt trên Windows

Để cài đặt GCC trên Windows, bạn cần cài đặt MinGW. Để cài đặt MinGW, hãy truy cập trang chủ MinGW, [www.mingw.org](https://www.mingw-w64.org/downloads/) và theo liên kết đến trang tải xuống MinGW. Tải xuống phiên bản mới nhất của chương trình cài đặt MinGW, có tên là MinGW-<version>.exe.

Trong khi cài đặt Min GW, ở mức tối thiểu, bạn phải cài đặt gcc-core, gcc-g++, binutils và thời gian chạy MinGW, nhưng bạn có thể muốn cài đặt thêm.

Thêm thư mục con bin của bản cài đặt MinGW vào biến môi trường **PATH** để bạn có thể chỉ định các công cụ này trên dòng lệnh bằng tên đơn giản của chúng.

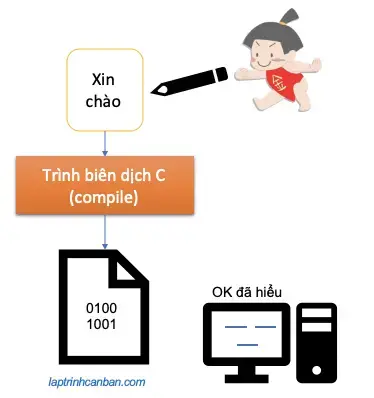
Sau khi cài đặt hoàn tất, bạn sẽ có thể chạy gcc, g++, ar, ranlib, dlltool và một số công cụ GNU khác từ dòng lệnh của Windows.

## Trình biên dịch để chạy và các câu lệnh để chạy

### Biên dịch chương trình là gì trong C

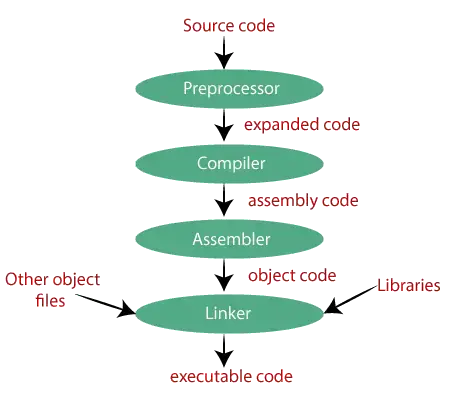
Trong các bài trước chúng ta đã biết, C là một ngôn ngữ bậc cao và nội dung viết trong đó rất giống với cách viết và suy nghĩ của con người. Tuy nhiên thì trong máy tính chỉ có 2 trạng thái tồn tại đó là **có dòng điện chạy qua** và **không có dòng điện chạy qua**, tương ứng với 2 số 0 và 1, do đó máy tính **không thể hiểu** được nội dung chúng ta đã viết trong mã nguồn của chương trình C đâu.

Và để cho máy tính có thể hiểu ngôn ngữ con người, chúng ta cần phải **biên dịch** nội dung đã viết sang dạng 1, 0 cho máy tính hiểu. Công việc này được gọi là biên dịch chương trình, hay còn gọi là **compile** chương trình, và **công cụ sử dụng** để biên dịch chương trình C được gọi là **trình biên dịch**.



### Quá trình biên dịch trong chương trình C

Để biên dịch một chương trình viết bởi ngôn ngữ C, chúng ta cần trải qua 4 bước sau đây. Các bước này được thực thi tự động trong **trình biên dịch của C**.



#### Preprocessor (tiền xử lý)

Đây là bước đầu tiên trong quá trình biên dịch chương trình. Tại đây sẽ **thực hiện các công đoạn chuẩn bị** trước khi chúng ta bắt đầu xử lý chính trong chương trình C.

Nói một cách đơn giản thì nếu việc biên dịch mã nguồn C là công việc nấu cơm thì tại Preprocessing (tiền xử lý) chúng ta sẽ tiến hành chuẩn bị gạo, rửa rau cắt thịt v.v.. trước khi bắt đầu nấu cơm vậy.

Có thể kể đến một số xử lý ở Preprocessing (tiền xử lý) trong biên dịch C như sau:

* Load và đọc các library cần thiết sử dụng trong chương trình
* Mở rộng các marcro được định nghĩa sau từ khóa define
* Xử lý trước các lệnh bắt đầu sau ký tự #
* Xóa comment trong mã nguồn, và biên dịch trước một số bộ phận trong mã nguồn.

#### Compiler (biên dịch)

Tiếp theo Preprocessor chính là Compiler (biên dịch) - xử lý chính trong trình biên dịch chương trình C.

Dựa vào compiler, mã nguồn được viết trong file C từ ngôn ngữ bậc cao mà con người hiểu được sẽ được biên dịch sang **ngôn ngữ assembly** ở dạng các mã assembly code. Ngôn ngữ assembly ở đây là **ngôn ngữ bậc thấp**, là ngôn ngữ trung gian giữa **ngôn ngữ bậc cao** và **ngôn ngữ máy tính**, có tác dụng chuyển ngôn ngữ bậc cao sang dạng các chỉ thị 1 đối 1 cho máy tính.

Compiler ở đây theo nghĩa hẹp có nghĩa là quá trình biên dịch mã nguồn C sang ngôn ngữ assembly. Tuy nhiên thông thường thì chúng ta cũng sử dụng **Compiler** theo nghĩa rộng chính là toàn bộ quá trình biên dịch từ Preprocessor (tiền xử lý) đến Linker (liên kết).

#### Assembler (tập hợp)

Tiếp theo, trình tập hợp Assembler sẽ chuyển đổi các mã **assembly code** đã dịch ở Compiler ở trên thành các **mã máy tính** - loại ngôn ngữ mà máy tính có thể hiểu được. Các mã máy tính này được biểu diễn bởi số 0 và số 1, và được tập hợp trong một file máy tính.

Mặc dù tại giai đoạn này, mã nguồn của chương trình C đã được chuyển thành một file ở dạng mà máy tính có thể hiểu được, nhưng ở giai đoạn này do chúng ta chưa liên kết đủ đủ thông tin trong file, nên file này chưa thể thực thi một cách bình thường được.

#### Linker (liên kết)

Đây là bước cuối cùng trong biên dịch chương trình trong C. Tại đây, chúng ta sử dụng trình liên kết **Linker** để liên kết các thông tin còn thiếu như các thư viện (library) chẳng hạn vào file máy tính đã tạo ở Assembler.

Việc liên kết thông tin cuối cùng đã hoàn thành được file máy tính, và chúng ta có thể chạy file này một cách bình thường.

### Các trình biên dịch trong C

Để biên dịch chương trình C, chúng ta cần cài đặt và sử dụng các **trình biên dịch**. Có rất nhiều trình biên dịch miễn phí mà chúng ta có thể sử dụng trong ngôn ngữ C như sau:

#### Trình biên dịch trong Visual Studio

Sau khi chúng ta cài đặt Visual Studio Community vào máy tính, một **trình biên dịch C** cũng sẽ được cài đặt kèm theo, và chúng ta có thể sử dụng ngay trình biên dịch này thông qua **Developer Command Prompt** được tích hợp sẵn sau khi cài để biên dịch C với cú pháp sau:

COPY

|  |
| --- |
| cl sample.c |

Ngoài ra thì bạn cũng có thể sử dụng Visual Studio để biên dịch một chương trình C++ như sau:

|  |
| --- |
| cl sample.cpp |

#### Trình biên dịch MinGW:

MinGW là một trình biên dịch ngôn ngữ C miễn phí trên Windows. Sau khi chúng ta cài đặt MinGW vào máy tính, chúng ta có thể sử dụng trình biên dịch này để compile chương trình C bằng lệnh sau đây:

|  |
| --- |
| gcc - o filename filename.c |

Ngoài ra thì bạn cũng có thể sử dụng MinGW để biên dịch một chương trình C++ như sau:

|  |
| --- |
| g++ - o filename filename.cpp |

#### Trình biên dịch C++ Compiler:

C++ Compiler cũng là một công cụ miễn phí tạo môi trường lập trình C. Sau khi chúng ta cài đặt C++ Compiler vào máy tính, một **trình biên dịch C** cũng sẽ được cài đặt kèm theo, và chúng ta có thể sử dụng ngay trình biên dịch này với lênh sau đây:

COPY

|  |
| --- |
| bcc32c sample.c |

#### Trình biên dịch trong Dev C++:

Dev C++ là một IDE miễn phí tạo môi trường lập trình C. Sau khi chúng ta cài đặt Dev C++ vào máy tính, chúng ta có thể viết chương trình, cũng như biên dịch và chạy trực tiếp các chương trình C và C++ trên IDE này.

## Lưu đồ

### Quy trình xử lý cơ bản của máy tính gồm I-P-O.

Shape

Description automatically generated

|  |  |
| --- | --- |
| **VD:** Xác định Input, Process, Output của việc làm 1 ly nước chanh nóng. | |
| Input | : ly, đường, chanh, nước nóng, muỗng. |
| Process | : - cho hỗn hợp đường, chanh, nước nóng vào ly.  - dùng muỗng khuấy đều. |
| Output | : ly chanh nóng đã sẵn sàng để dùng. |

|  |  |
| --- | --- |
| **VD:** Xác định Input, Process, Output của chương trình tính tiền lương công nhân tháng 10/2002 biết rằng lương = lương căn bản \* ngày công. | |
| Input | : lương căn bản, ngày công |
| Process | : nhân lương căn bản với ngày công |
| Output | : lương |

|  |  |
| --- | --- |
| **VD:** Xác định Input, Process, Output của chương trình giải phương trình bậc nhất ax + b = 0. | |
| Input | : hệ số a, b |
| Process | : chia – b cho a |
| Output | : nghiệm x |

|  |  |
| --- | --- |
| **VD:** Xác định Input, Process, Output của chương trình tìm số lớn nhất của 2 số a và b. | |
| Input | : hệ số a, b |
| Process | : Nếu a > b thì Output = a lớn nhất |
| Output | : Ngược lại Output = b lớn nhất |

* **Bài tập**: Xác định Input, Process, Output của các chương trình sau

**1**. Đổi từ tiền VND sang tiền USD.

**2**. Tính điểm trung bình của học sinh gồm các môn Toán, Lý, Hóa.

**3**. Giải phương trình bậc 2: a + bx + c = 0

**4**. Đổi từ độ sang radian và đổi từ radian sang độ (công thức α/π = a/180, với α: radian, a: độ)

**5**. Kiểm tra 2 số a, b giống nhau hay khác nhau.

### Sử dụng lưu đồ (Flowchart)

Để dễ hơn về quy trình xử lý, các nhà lập trình đưa ra dạng lưu đồ để minh họa từng bước quá trình xử lý một vấn đề (bài toán).

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

### Bài tập

Vẽ lưu đồ cho các chương trình sau:

1. Đổi từ tiền VND sang tiền USD.
2. Tính điểm trung bình của học sinh gồm các môn Toán, Lý, Hóa.
3. Giải phương trình bậc 2: a + bx + c = 0
4. Đổi từ độ sang radian và đổi từ radian sang độ (công thức α/π = a/180, với α: radian, a: độ)