# LẬP TRÌNH C CƠ BẢN

# A. PHẦN CƠ BẢN

## I. TỔNG QUAN VỀ CHƯƠNG TRÌNH C

### 1. Giới thiệu chương trình cơ bản.

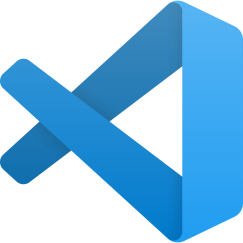
- Các chương trình thường gặp trong ngày.

 + Hệ điều hành:





+ Các ứng dụng:



- Nhìn sơ qua thì chúng ta sẽ thấy từ **chương trình** đến **hệ điều hành điều** có một nguyên lý cơ bản là như sau:  
 + **Phần chương trình chính** : Luôn có một nhân để xử lý trong một hệ thống.

+ **Phần tài nguyên có sẵn**: Là phần tài nguyên có sẵn để có thể xử lý khi nhận dữ liệu đầu vào:

+ **Phần dữ liệu đầu vào**: Dữ liệu mà ta muốn xử lý chúng.  
 + **Phần thông tin** :Phần thông tin đưa ra sau khi sử dụng dữ liệu.

Tài nguyên sẵn có

Thông tin

Dữ liệu

*Hình 1. Cơ bản của chương trình*

- Một ví dụ cơ bản là bạn giải một diện tích hình chữ nhật với **chiều dài = 15**, **chiều rộng = 10**. Thì bạn chắc **phải có** công thức tính trong đầu là **S = chiều dài x chiều rộng**.

Dữ liệu là chiều dài = 15, chiều rộng = 10. Cùng tài nguyên sẳn có ( công thức tính) , bạn sẽ đưa ra thông tin, diện tích là 150. Nguyên lý hoạt động của máy tính cơ bản là như vậy, sẽ có đầu vào, đầu ra, phần xử lý, thông tin.

### 2. Lập trình máy tính căn bản là gì?

**Lập trình máy tính** hay **lập chương trình máy tính** thường gọi tắt là lập trình ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): **Computer programming**, thường gọi tắt là **programming**) là việc lập ra [chương trình làm việc cho máy](https://vi.wikipedia.org/wiki/Ch%C6%B0%C6%A1ng_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh) có [bộ xử lý](https://vi.wikipedia.org/wiki/CPU), nói riêng là [máy tính](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh), để thực thi nhiệm vụ xử lý thông tin nào đó. Chương trình này gồm dãy các chỉ thị (hay lệnh) hợp lý để máy thực hiện theo trình tự thời gian.

*(Nguồn: https://vi.wikipedia.org/wiki/Lập\_trình\_máy\_tính)*

Một cách dễ hiểu nữa, Lập trình máy tính chính là chúng ta sử dụng các công cụ cần thiết để giao tiếp với máy tính ( ra lệnh, yêu cầu, nhờ vả) máy tính để chúng thực hiện công việc thay cho chúng ta, với một kịch bản soạn thảo trước đó.

### 3. Lập trình c là gì?

- Lập trình ngôn ngữ C chính là dùng ngôn ngữ c để lập trình máy tính.

- **C là một ngôn ngữ lập trình cấp trung**, ngôn ngữ lập trình C được phát triển bởi **Dennis M. Ritchie** để phát triển hệ điều hành UNIX tại Bell Labs. C được thực thi lần đầu tiên trên máy tính DEC PDP-11 vào năm 1972.

- Được coi là ngôn ngữ mẹ của các ngôn ngữ khác, là nền tảng của các ngôn ngữ khác.

- Điều kiện tiên quyết học ngôn ngữ c hay lập trình tốt hơn:  
 + Hiểu biết ngôn ngữ lập trình.

+ Biết được ưu nhược điểm của c.

+ Trước khi học lập trình, bạn phải có kiến thức căn bản về máy tính.

Viết chương trình trên IDE

Code được chuyển sang dạng mã máy để máy tính đọc hiểu và thực thi

Những dòng code thường thấy

Máy tính thực thi

*Hình 2: Sơ đồ máy tính thực thi từ coding đến running*

*- Chương trình c biên dịch và thực thi như thế nào?*

Quy trình dịch là quá trình chuyển đổi từ ngôn ngữ bậc cao (NNBC) (C/C++, Pascal, Java, C#…) sang ngôn ngữ đích (ngôn ngữ máy) để máy tính có thể hiểu và thực thi. Ngôn ngữ lập trình C là một ngôn ngữ dạng biên dịch. Chương trình được viết bằng C muốn chạy được trên máy tính phải trải qua một quá trình biên dịch để chuyển đổi từ dạng mã nguồn sang chương trình dạng mã thực thi. Quá trình được chia ra làm 4 giai đoạn chính:

- Giai đoàn tiền xử lý (Pre-processor)

- Giai đoạn dịch NNBC sang Asembly (Compiler)

- Giai đoạn dịch asembly sang ngôn ngữ máy (Asember)

- Giai đoạn liên kết (Linker)



*Hình 2.1 Quá trình biên dịch*

**\*** HOẠT ĐỘNG

Giai đoạn tiền xử lý – Preprocessor  
Giai đoạn này sẽ thực hiện:

* Nhận mã nguồn
* Xóa bỏ tất cả chú thích, comments của chương trình
* Chỉ thị tiền xử lý (bắt đầu bằng #) cũng được xử lý

*Ví dụ*: chỉ thị #include cho phép ghép thêm mã chương trình của một tệp tiêu để vào mã nguồn cần dịch. Các hằng số được định nghĩa bằng #define sẽ được thay thế bằng giá trị cụ thể tại mỗi nơi sử dụng trong chương trình.

Cộng đoạn dịch Ngôn Ngữ Bậc Cao sang Assembly

* Phân tích cú pháp (syntax) của mã nguồn NNBC
* Chuyển chúng sang dạng mã Assembly là một ngôn ngữ bậc thấp (hợp ngữ) gần với tập lệnh của bộ vi xử lý.

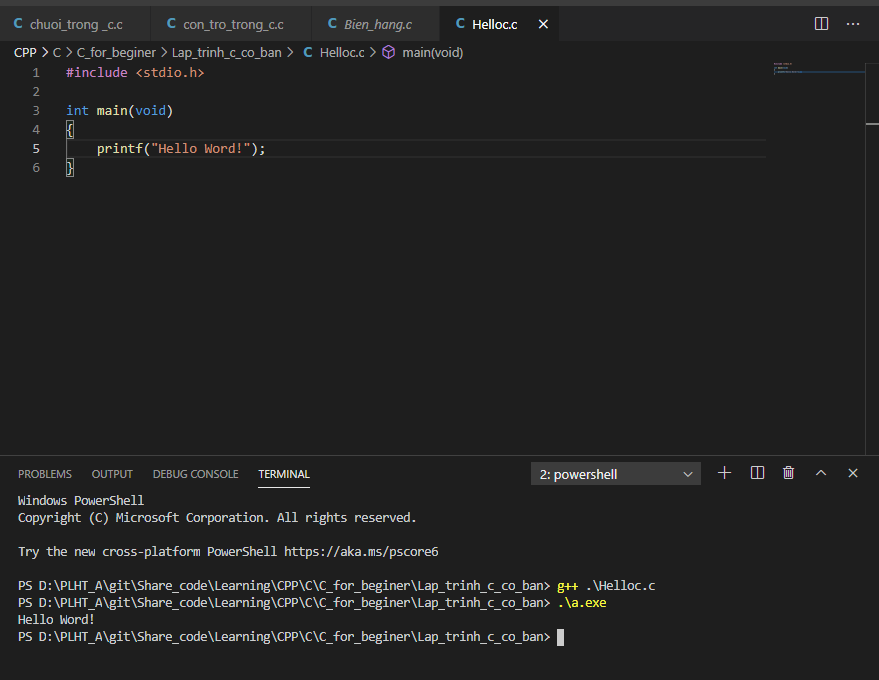
Công đoạn dịch Assembly

* Dich chương trình => Sang mã máy 0 và 1
* Một tệp mã máy (.obj) sinh ra trong hệ thống sau đó.

Giai đoạn Linker

* Trong giai đoạn này mã máy của một chương trình dịch từ nhiều nguồn (file .c hoặc file thư viện .lib) được liên kết lại với nhau để tạo thành chương trình đích duy nhất
* Mã máy của các hàm thư viện gọi trong chương trình cũng được đưa vào chương trình cuối trong giai đoạn này.
* Chính vì vậy mà các lỗi liên quan đến việc gọi hàm hay sử dụng biến tổng thể mà không tồn tại sẽ bị phát hiện. Kể cả lỗi viết chương trình chính không có hàm main() cũng được phát hiện trong liên kết.

-> Kết thúc quá trình tất cả các đối tượng được liên kết lại với nhau thành một chương trình có thể thực thi được (executable hay .exe) thống nhất.



Phần code mà chúng ta thường hay viết

Chạy file .exe để kiểm thử

Biên dịch code thành file .exe trên windows

*Hình 3. Giới thiệu chương trình c*

## TỔNG KẾT

- Vậy một chương trình từ hệ điều hành đến chương trình con điều gồm những phần cơ bản sao:  
 + Phần yêu cầu dữ liệu cần được xử lý.

+ Phần tài nguyên hiện có.

+ Phần NHÂN để xử lý ( não bộ).

+ Phần đưa ra thông tin.

- Chương trình C cũng có những phần tương tự.  
 + Phần hỗ trợ mà trình biên dịch cung cấp ( thư viện, lệnh tiền xử lý, kiểu dữ liệu, hàm), tôi gọi đây là phần tài nguyên sẳn có.

+ Phần yêu cầu đề bài cần giải quyết, tôi gọi đây là phần yêu cầu dữ liệu cần được xử lý.

+ Trình biên dịch, đây là phần mà chúng ta build hay debug một source bằng g++, g++ sẽ thu thập toàn bộ code, thư viện của chúng ta để biên dịch thành một bộ mã mà máy tính có thể hiểu.  
 + File .exe được tạo (nếu bạn đang sài hệ điều hành windows), đây chính là phần đưa ra thông tin, thông tin chính là một gói chương trình được đưa ra.

- Khi chúng ta soạn thảo Code chương trình C cũng gồm 4 phần:

+ Phần yêu cầu bài toán ( Đây là phần chúng ta cần phải xử lý dữ liệu.

+ Phần tiền xử lý ( Đây là nơi ta khai báo thư viện, tiền lệnh, … là chúng ta đang cung cấp tài nguyên cho chương trình).

+ Phần nhân ( hàm main) phần lỗi của chương trình, xử lý tất cả những phần xử lý.

+ Hàm và biến ( Đây được coi như phần sử dụng dữ liệu thành kiểu dữ liệu có sẳn, xử lý những phần nhỏ để phần main có thể tổng hợp xử lý chung.

+ Phần đưa ra thông tin (Nơi đây chúng ta dùng comment để cho biết đoạn code mà chúng ta code nó thực hiện chức năng gì? Đưa ra thông tin để người trong nhóm có thể nắm bắt được.

Tóm lại, ở đây các bạn cần lưu lý, Một chương trình cơ bản của máy tính; Đặc biệt là phân biệt rõ lập trình c (Từ soạn code đến thành file .exe) và lập trình c ( Quá trình soạn code). Chúng điều có 4 phần nhưng ở mỗi gian đoạn, 4 phần thực hiện ở 1 lĩnh vực khác.

## II. CẤU TRÚC CỦA MỘT CHƯƠNG TRÌNH C

### 1. Tiền xử lý.

**Bộ tiền xử lý trong C** ở đây không phải là một phần của bộ biên dịch, nhưng có những bước riêng rẽ trong quá trình biên dịch. Theo cách hiểu cơ bản nhất, bộ tiền xử lý trong ngôn ngữ C là các công cụ thay thế văn bản và hướng dẫn trình biên dịch không yêu cầu tiền xử lý trước khi được biên dịch. Chúng tôi hướng đến bộ tiền xử lý C như CPP.

Tất cả các lệnh tiền xử lý bắt đầu với ký thự “**#”**. Nó ít nhất không phải là ký tự trắng, để dễ dàng đọc.

Ví dụ minh họa:

Phân tích các ví dụ sau để hiểu các directive đa dạng.

#define DO\_DAI\_MANG\_TOI\_DA 20

Tiền xử lý này thông báo cho trình biên dịch C thay thế DO\_DAI\_MANG\_TOI\_DA với 20. Sử dụng *#define* cho các hằng số làm tăng khả năng đọc của chương trình.

#include <stdio.h>

#include "header.h"

Tiền xử lý này thông báo cho trình biên dịch lấy thư viện stdio.h từ **Thư viện hệ thống** và thêm vào mã nguồn hiện tại. Dòng kế tiếp thông báo cho trình biên dịch lấy tệp **header.h** từ thư mục máy tính và thêm nội dung và mã nguồn hiện tại.

#undef KICH\_CO\_FILE

#define KICH\_CO\_FILE 42

Tiền xử lý này thông báo cho trình biên dịch vộ hiệu hóa biến KICH\_CO\_FILE và định nghĩa mới có giá trị 42.

## Dưới đây là danh sách các thẻ tiền xử lý quan trọng trong C:

|  |  |
| --- | --- |
| **Directive** | **Miêu tả** |
| #define | Thay thể cho bộ tiền xử lý macro |
| #include | Chèn một header đặc biệt từ file khác |
| #undef | Không định nghĩa một macro tiền xử lý |
| #ifdef | Trả về giá trị true nếu macro này được định nghĩa |
| #ifndef | Trả về giá trị true nếu macro này không được định nghĩa |
| #if | Kiểm tra nếu điều kiện biên dịch là đúng |
| #else | Phần thay thế cho #if |
| #elif | #else một #if trong một lệnh |
| #endif | Kết thúc điều kiện tiền xử lý |
| #error | In thông báo lỗi trên stderr |
| #pragma | Thông báo các lệnh đặc biệt đến bộ biên dịch, sử dụng một phương thức được tiêu chuẩn hóa |

*Bảng 1.Danh sách thẻ tiền xử lý thông dụng*

### 2. Hàm main.

### 3. Biến và hàm khác.

### 4. Comments trong C.

## III. PHẦN HEADER – TIỀN XỬ LÝ

1. Lệnh tiền xử lý

2. Thư viện

## IV. HÀM MAIN - HÀM CHÍNH TRONG C

## V. BIẾN VÀ HÀM KHÁC

## VI. GIỚI THIỆU MỘT SỐ GIẢI THUẬT THÔNG DỤNG

## VII. MỘT SỐ CÁCH PHÂN TÍCH YÊU CẦU ĐỂ GIẢI QUYẾT VẤN ĐỀ

# B. PHẦN MỞ RỘNG

## I. DANH SÁCH

### 1. Danh sách

### 2. Danh sách liên kết

##### a. Danh sách liên kết đơn.

##### b. Danh sách liên kết đôi.

## II. BẢNG BĂM

### 1. Băm trong.

### 2. Băm ngoài

## III. CẤU TRÚC DỮ LIỆU

### 1. Cấu trúc thường.

### 2. Một số dạng cấu trúc khác.

## IV. GIỚI THIỆU MỘT SỐ TỪ KHÓA TRONG C

## V. COMMENTS