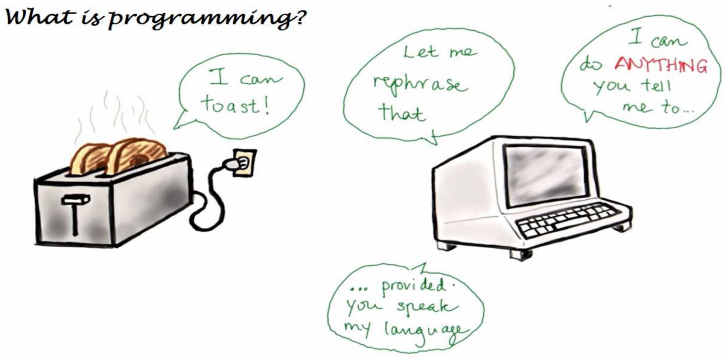


Giới thiệu

CT101 – Lập trình căn bản

Khoa CNTT&TT – ĐHCT

Mục đích

- Trong phần này, trước tiên chúng ta sẽ tìm hiểu như thế nào là chương trình (program) , ngôn ngữ lập trình (programming language) và lập trình (programming).
- Sau đó ta sẽ bước đầu làm quen với ngôn ngữ lập trình C.

Yêu cầu

- Học viên cần giải thích được tính chất của chương trình máy tính, nó được thực thi như thế nào và làm sao để tạo ra chương trình máy tính.
- Song song đó, học viên cần có thể viết chương trình đơn giản bằng ngôn ngữ C, biên dịch và thực thi nó.

Nội dung

- Trong phần giới thiệu này chúng ta sẽ xem xét một số tính năng cơ bản của máy tính, mã lệnh của máy tính, cách máy tính thực thi mã lệnh và cách chúng ta tạo ra mã lệnh cho máy tính sử dụng ngôn ngữ lập trình C.

Phương trình cơ bản về máy tính

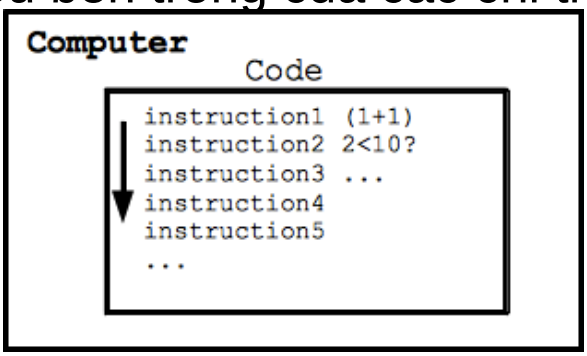
- Máy tính = Sức mạnh + Sự ngu ngốc [Nick Parlante, Stanford University]
- Sức mạnh:
  - Có thể đọc qua một lượng khổng lồ dữ liệu.
  - Có thể thực thi hàng tỉ “tác vụ” trong một giây.
- Sự ngu ngốc:
  - Các tác vụ rất đơn giản và mang tính cơ khí!
  - Thiếu “trí thông minh”.
- Chúng ta sẽ cảm nhận phương trình này trong các bài thực hành...

Phương trình cơ bản về máy tính

- Mặc dù có tính chất buồn cười này, nhưng máy tính lại cực kỳ hữu ích.
- Sự hữu ích này có được khi con người ra lệnh “thông minh” cho nó.
- Lớp học CT101 sẽ là bước tiếp cận đầu tiên để chúng ta hiểu được máy tính làm việc như thế nào, làm sao để ra lệnh cho nó và ra lệnh thế nào cho “thông minh”.

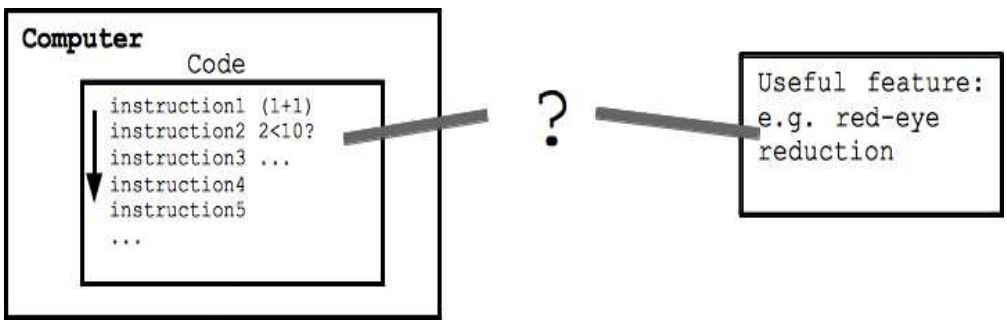
Máy tính làm việc như thế nào?

- Máy tính được điều hành bởi các "mã lệnh cho máy tính" (machine codes), gọi tắt là "mã máy".
- Mã máy được tạo thành từ những chỉ thị đơn giản, có tính chất cơ học.
- Máy tính chạy nhiều dãy chỉ thị mà không biết ý nghĩa, mục tiêu bên trong của các chỉ thị này.



Ngu ngốc nhưng cung cấp nhiều tiện ích

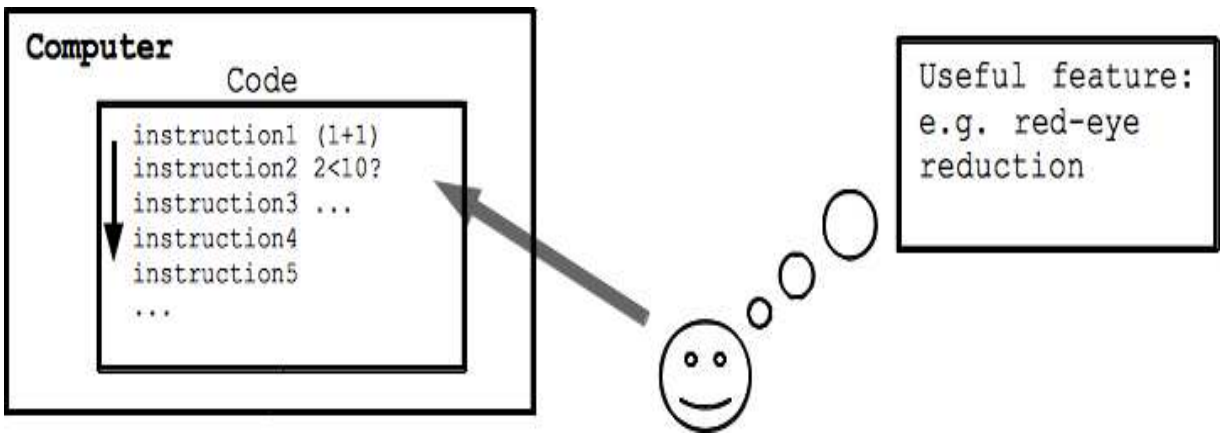
- Thử nhìn lại tất cả các tính năng hữu ích của máy tính: email, web, chơi nhạc, xử lý ảnh (xoá mắt đỏ chẳng hạn…)
- Nếu máy tính quá ngốc thì tại sao chúng lại hữu ích?
- Cái gì gắn kết 2 mặt của vấn đề này?



Lập trình cho máy tính

- Cái gì gắn kết 2 mặt của vấn đề này?
- Lập trình viên máy tính đã tạo ra sự gắn kết:
  - Lập trình viên nghĩ ra nhiều tính năng hữu ích.
  - Sau đó suy nghĩ thấu đáo các giải pháp để cài đặt.
  - Chia nhỏ giải pháp ra để viết ***mã nguồn*** (***source code***).
  - Mã nguồn có thể được viết dưới dạng thức của một ***ngôn ngữ lập trình*** (***programming language***) nào đó.
  - Biên dịch mã nguồn thành ***mã máy*** cho máy tính.
  - Đây gọi là lập trình cho máy tính.

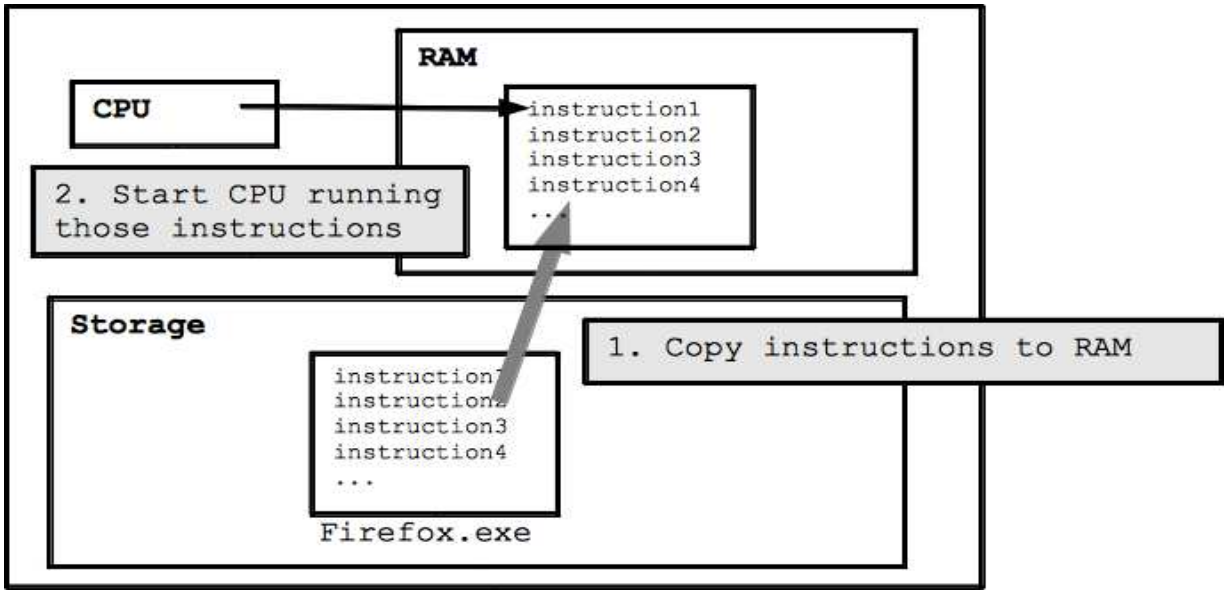
Lập trình cho máy tính



Lập trình cho máy tính

- Một mã máy định nghĩa một tập các chỉ thị đơn lẻ.
- Một chỉ thị ở dạng mã máy là rất nguyên thuỷ, ví dụ như: cộng hai số hay kiểm tra xem một số có bằng 0 hay không.
- Một "***chương trình***" (***program***), ví dụ như Firefox, được tạo thành từ một dãy hàng triệu chỉ thị
- CPU chạy một chu trình "nạp và thực thi" để thực thi một chương trình:
  - Nạp một chỉ thị trong dãy chỉ thị.
  - Thực thi chỉ thị đó (ví dụ như cộng hai số).
  - Nạp chỉ thị kế tiếp.
  - Tiếp tục đến khi tất cả các chỉ thị trong chương trình được thực thi.

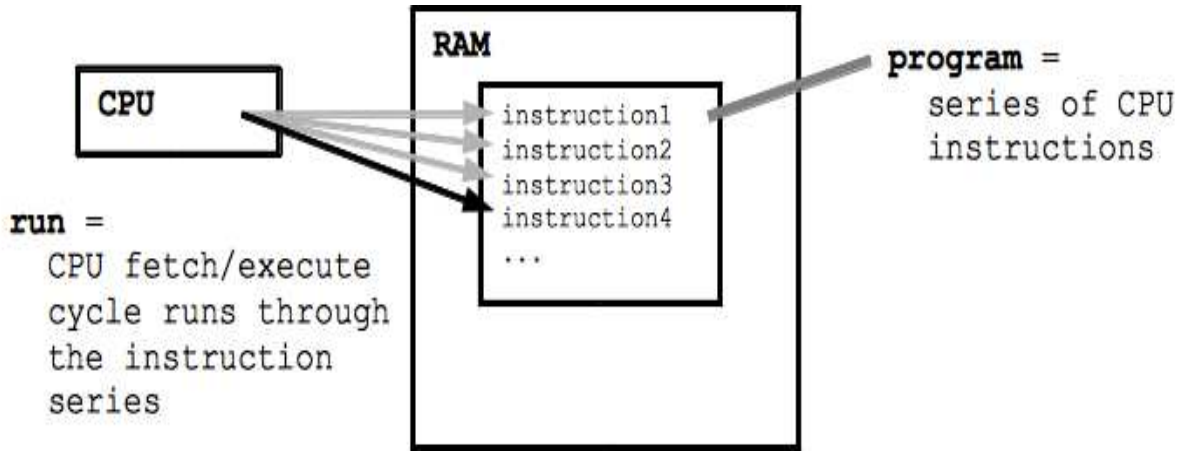
Lập trình cho máy tính



Nạp chương trình FireFox

Lập trình cho máy tính

- Thực thi chương trình:



Lập trình cho máy tính

- Ngôn ngữ lập trình:
  - Hiếm khi lập trình viên viết ra mã máy trực tiếp.
  - Họ thường viết mã lệnh bằng một ngôn ngữ cấp cao với nhiều tính năng hữu ích và mạnh mẽ hơn là các chỉ thị đơn giản như trong mã máy (ví dụ như vòng lặp, điều kiện rẽ nhánh, xử lý chuỗi, …).

Lập trình cho máy tính

- Ví dụ chương trình nguồn fact.c viết bằng ngôn ngữ C

```
#include <stdio.h>
void main(int argc, char *argv[]) {
    int i, result;
    result = 1;
    for (i = 1; i<6; i++) {
        result = result * i;
    }
    printf ("Result is:  %d.\n",result);
}
```


Lập trình cho máy tính

- Trích đoạn chương trình fact.s viết bằng hợp ngữ và tương đương với chương trình fact.c

```
.file       "fact.c"
gcc2_compiled.:
            .global .umul
.section  ".rodata"
            .align 8
.LLC0:
            .asciz     "Result is:  %d.\n"
.section  ".text"
        .align 4
        .global main
        .type            main,#function
        .proc            020
main:
        !#PROLOGUE# 0
        save             %sp, -120, %sp
```

Lập trình cho máy tính

- Trích đoạn chương trình fact.o ở dạng mã máy và tương đương với chương trình fact.c

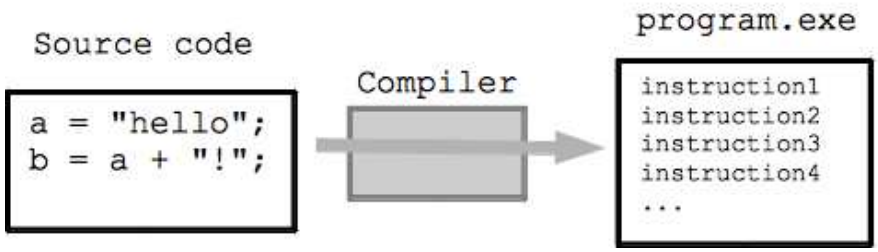
```
0000000 7f45 4c46 0102 0100 0000 0000 0000 0000
0000020 0001 0002 0000 0001 0000 0000 0000 0000
0000040 0000 0234 0000 0000 0034 0000 0000 0028
0000060 0008 0001 002e 7368 7374 7274 6162 002e
0000100 7465 7874 002e 726f 6461 7461 002e 7379
0000120 6d74 6162 002e 7374 7274 6162 002e 7265
0000140 6c61 2e74 6578 7400 2e63 6f6d 6d65 6e74
0000160 0000 0000 9de3 bf88 f027 a044 f227 a048
0000200 9010 2001 d027 bfe8 9010 2001 d027 bfec
0000220 d007 bfec 80a2 2005 0480 0004 0100 0000
0000240 1080 000c 0100 0000 d007 bfe8 d207 bfec
0000260 4000 0000 0100 0000 d027 bfe8 d007 bfec
0000300 9202 2001 d227 bfec 10bf fff2 0100 0000
0000320 1300 0000 9012 6000 d207 bfe8 4000 0000
0000340 0100 0000 81c7 e008 81e8 0000 0000 0000
```


Lập trình cho máy tính

- Ngôn ngữ lập trình:
  - Hiếm khi lập trình viên viết ra mã máy trực tiếp.
  - Họ thường viết mã lệnh bằng một ngôn ngữ cấp cao với nhiều tính năng hữu ích và mạnh mẽ hơn là các chỉ thị đơn giản như trong mã máy (ví dụ như vòng lặp, điều kiện rẽ nhánh, xử lý chuỗi, …).
- Người ta cũng cần công cụ để chuyển đổi mã lệnh từ ngôn ngữ cấp cao (lập trình viên dễ xem xét và lập trình) xuống thành mã máy (để máy tính hiểu và thực thi): Trình biên dịch lệnh (compiler) hoặc thông dịch lệnh (interpreter)

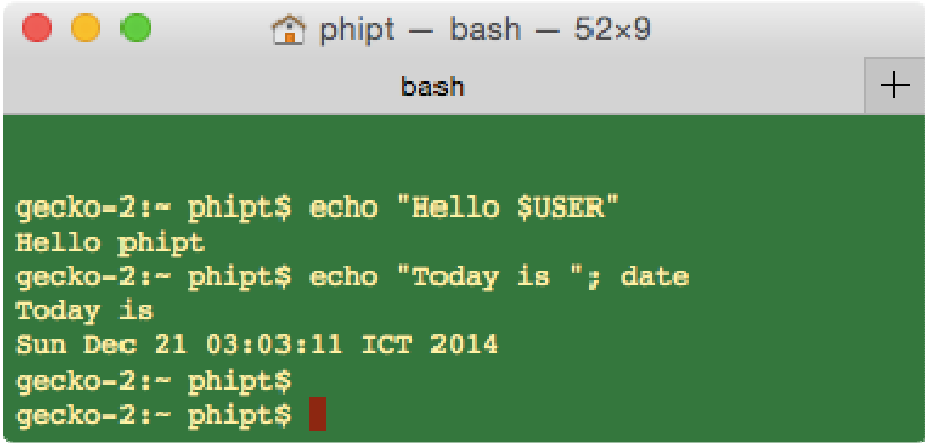
Lập trình cho máy tính

- Trình biên dịch:
  - Dịch mã nguồn thành một lượng lớn chỉ thị dạng mã máy
  - Kết quả cuối cùng là chương trình mà người dùng có thể thực thi.



Lập trình cho máy tính

- Trình thông dịch:
  - Là một chương trình mà tự nó có thể chạy các dạng mã lệnh khác.
  - Trình thông dịch nạp chương trình nguồn, phân tích và chạy tuần tự từng lệnh trong chương trình nguồn này.



Lập trình cho máy tính

- Trình biên dịch so với thông dịch:
  - Trình biên dịch sẽ dịch tất cả mã nguồn thành chương trình dạng mã máy.
  - Trình thông dịch đọc từng dòng lệnh trong chương trình nguồn, dịch và chạy từng lệnh ngay tức thì. Trình thông dịch không sinh ra chương trình dạng mã máy. Thay vào đó nó thực thi những hành động được chỉ ra trong chương trình nguồn một cách trực tiếp.

Lập trình cho máy tính

- Cần tính kiên nhẫn vì nó giống như trò chơi ghép hình Lego.
- Chúng ta có thể xây dựng được nhiều mô hình lớn và đẹp, nhưng phải bắt đầu từ những viên gạch – những mã lệnh mà chúng ta vừa thảo luận.

Lập trình bằng ngôn ngữ C

- Có nhiều ngôn ngữ lập trình hiện hành: C, Java, Python, …
- Lý do ta chọn ngôn ngữ C?
  - Mã lệnh viết bằng C chạy nhanh vì nó gần với ngôn ngữ máy nhất, nhưng lại không có máy móc và cơ học như ngôn ngữ máy.

Lập trình bằng ngôn ngữ C

- Môi trường cơ bản để viết chương trình bằng ngôn ngữ C bao gồm:
  - Công cụ soạn thảo mã nguồn: bất kỳ chương trình soạn thảo văn bản nào.
  - Công cụ biên dịch: gcc, g++
  - Công cụ gỡ rối: gdb, valgrind.
- Trong Windows, tất cả các gói phần mềm biên dịch và gỡ rối đều có sẵn trong chương trình *cygwin*.
- Trong Linux: trình biên dịch có trong gói phần mềm: *build-essential*
- Trong Mac OSX: *Xcode*.

Lập trình bằng ngôn ngữ C

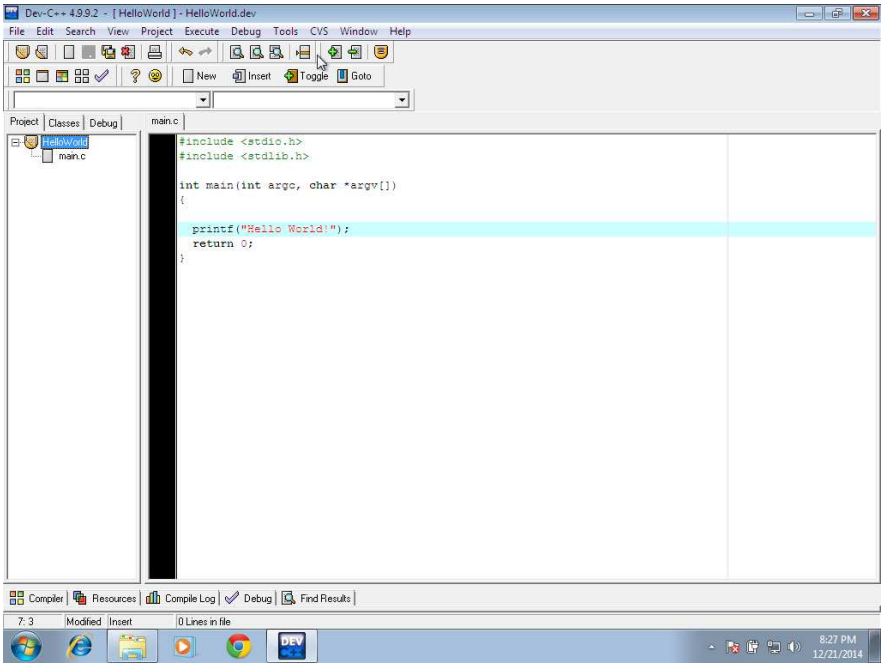
- Môi trường phát triển tích hợp (IDE):
  - Là chương trình tích hợp tất cả các công cụ soạn thảo chương trình nguồn, biên dịch, gỡ rối.
  - Một số ví dụ
    - DevC (Windows)
    - Code::Blocks (Windows, Linux, Mac OSX)
    - Eclipse (Windows, Linux, Mac OSX)
    - Visual Studio (Windows)
    - Xcode (Mac OSX)

DevC

- Cài đặt trên Windows:
  - Tải bộ công cụ DevC tại:
    http://www.bloodshed.net/dev/devcpp.html
  - Click chuột lên chương trình vừa tải về để thực hiện cài đặt.

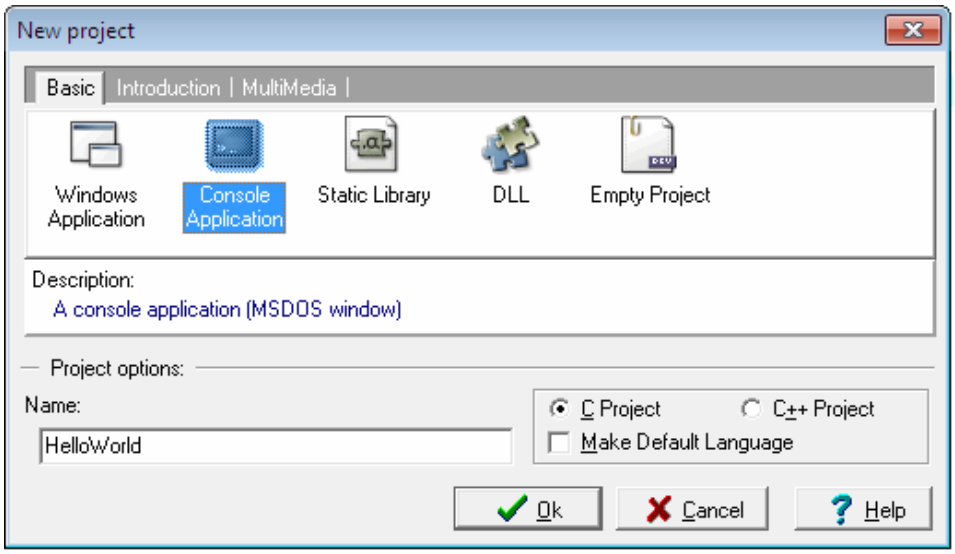
DevC

- Màn hình làm việc của DevC:



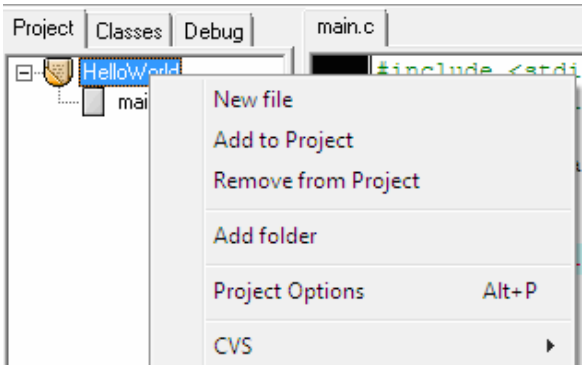
DevC

- Để lập trình trên DevC:
  - Tạo một dự án mới: **File -> New -> Project**.
  - Ở đây tên dự án là **HelloWorld**, kiểu là **Console Application**.



DevC

- Để lập trình trên DevC:
  - Tạo một tập tin mã nguồn mới: Click chuột phải trên biểu tượng dự án **HelloWorld** -> **New File**.
  - Ở đây tên tập tin là **main.c**.



Mã nguồn main.c

Ta sẽ thử in ra màn hình dòng chữ **Hello World!**
Nội dung mã nguồn như sau:

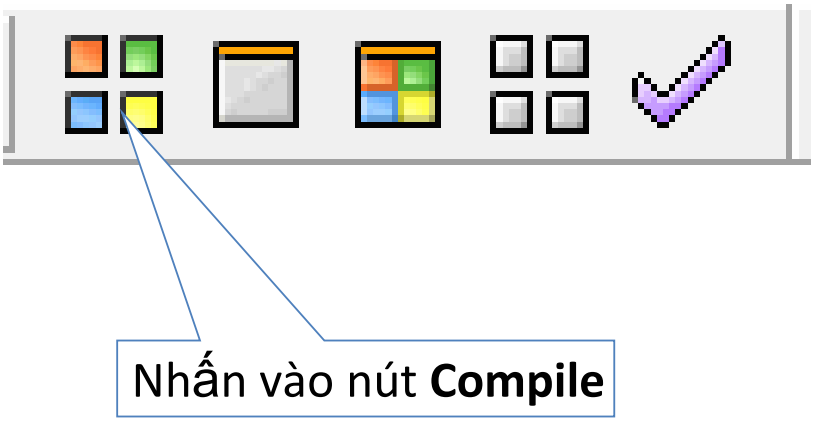
```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{

  printf("Hello World!");
  return 0;
}
```

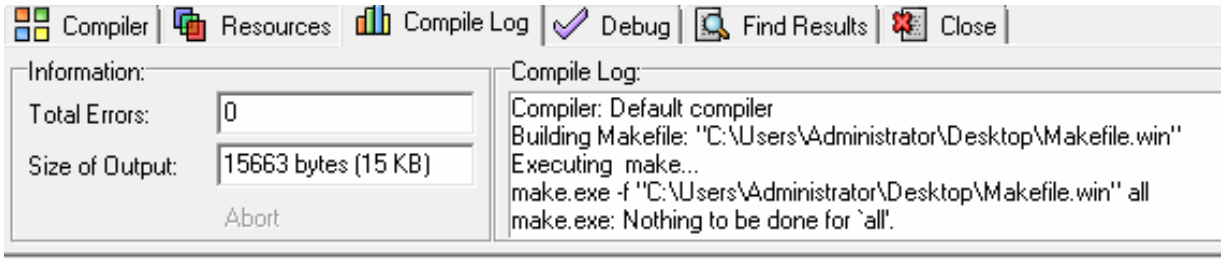
Biên dịch và chạy

- Biên dịch từ DevC:
  - Menu **Execute** -> **Compile**
  - Hoặc từ thanh công cụ:



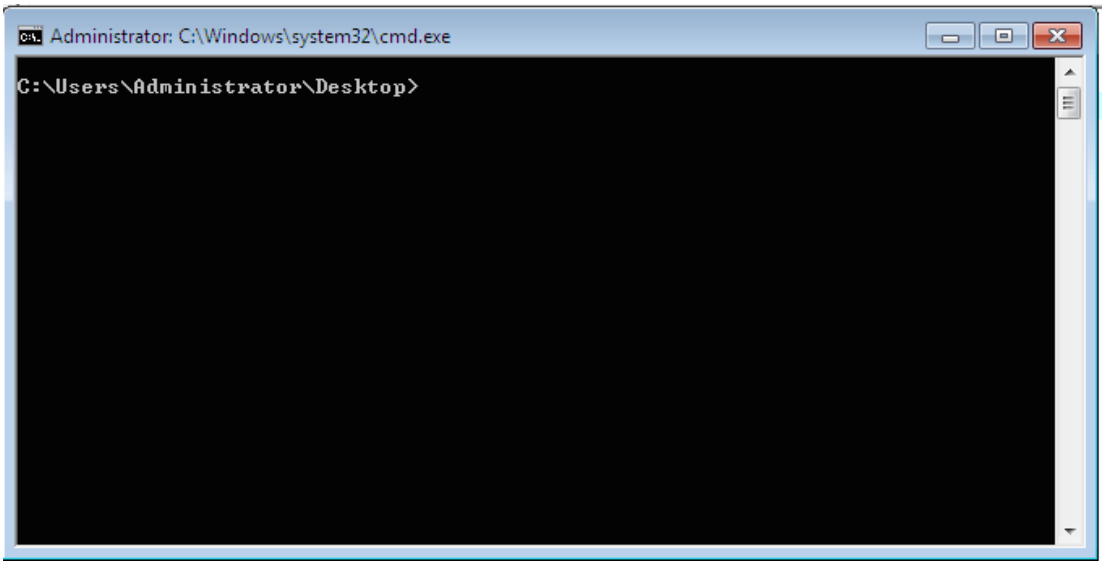
Biên dịch và chạy

- Biên dịch từ DevC:
  - Kết quả biên dịch được hiển thị trong tab **Compile Log**



Biên dịch và chạy

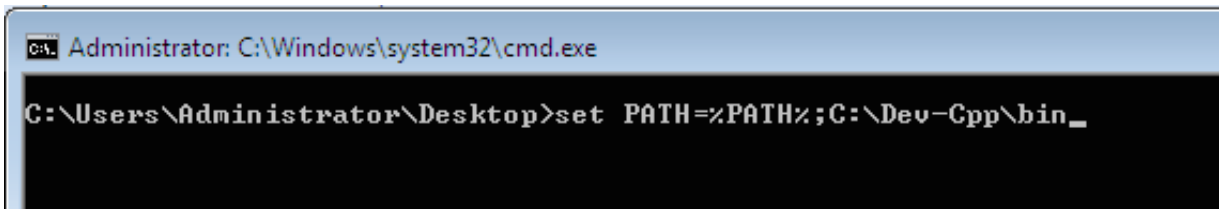
- Biên dịch từ cửa sổ lệnh của Windows:
  - Mở cửa sổ lệnh của Windows: **Start** -> **Cmd**



Biên dịch và chạy

- Biên dịch từ cửa sổ lệnh của Windows:
  - Thêm đường dẫn đến thư mục chứa công cụ **gcc** của DevC (giả sử ta cài đặt DevC ở **C:\Dev-Cpp**):

    set PATH=%PATH%;C:\Dev-Cpp\bin



Biên dịch và chạy

- Biên dịch từ cửa sổ lệnh của Windows: Mã nguồn -> Mã lệnh
  - Chuyển đến thư mục chứa dự án **HelloWorld:**
    cd C:\Users\Administrator\Desktop
  - Biên dịch

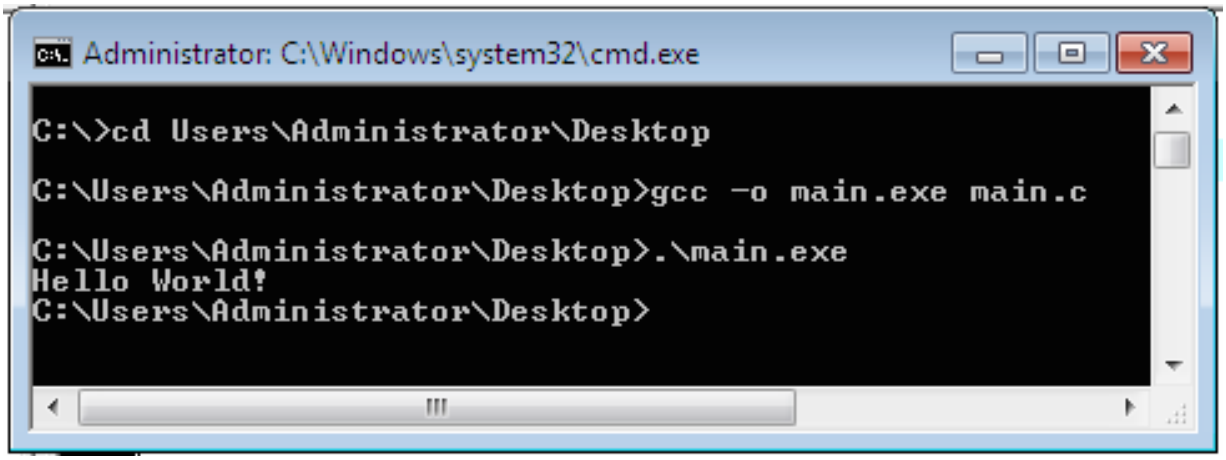
gcc –o main.exe main.c

(main.c là mã nguồn, **main.exe** là chương trình dạng mã máy)

- Chạy:

.\main.exe

Biên dịch và chạy



Tổng kết

- Chúng ta đã xem xét cách thức mà một hệ thống máy tính hoạt động dựa trên các mã lệnh mà nó được giao phó để thực thi.
- Chúng ta cũng đã trải nghiệm với việc tạo ra tập mã lệnh đơn giản cho máy tính bằng ngôn ngữ lập trình C.

Kiểm tra kiến thức

1. Bộ não của máy tính là gì?
2. Máy tính hoạt động như thế nào?
3. Cần có những gì để lập trình viên biến ý tưởng của mình thành chương trình máy tính?

Bài tập tổng kết

1. Hãy tạo dự án HelloWorld như trình bày phía trên, biên dịch và chạy chương trình.
2. Giả sử dự án HelloWorld gồm có 3 tập tin chứa mã nguồn hello.h, hello.c, main.c như sau:

1) hello.h

```
void say_hello();
```

2) hello.c

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

void say_hello() {
        printf("Hello, World!\n");
}
```


Bài tập tổng kết

Thực hành

3) main.c

```
#include "hello.h"

int main() {
        say_hello();
        return 0;
}
```

Nếu dùng dòng lệnh thì ta phải làm gì để biên dịch và chạy chương trình từ dự án trên?

CT101 – Lập trình căn bản

Khoa CNTT&TT – ĐHCT