

Ngôn ngữ lập trình C

huydq@soict.hust.edu.vn

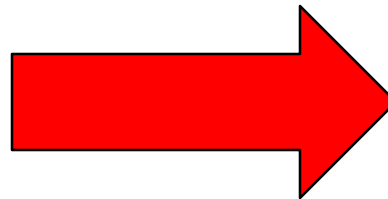
Ngôn ngữ lập trình bậc cao

```
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```

Mã nguồn



```
10100110 01110110
00100110 00000000
11111010 11111010
01001110 10100110
11100110 10010110
11001110 00101110
10100110 01001110
11111010 01100110
01001110 10000110
```

etc...

Mã máy

- **Chương trình dịch** sẽ dịch một chương trình viết ở ngôn ngữ bậc cao sang mã máy

Tại sao học ngôn ngữ C ?

- Là ngôn ngữ có tính uyển chuyển cao
 - Lập trình có cấu trúc
 - Có khả năng hỗ trợ các thao tác ở mức thấp
- Có tính khả chuyển cao
- Dịch ra chương trình mã máy có kích thước nhỏ và rất hiệu quả khi thực hiện
- được sử dụng rộng rãi trong lập trình chuyên nghiệp
- Là ngôn ngữ nền tảng của một số ngôn ngữ khác (C++, Java, Perl, awk)

Lịch sử ngôn ngữ C

- **CPL** Combined Programming Language (Barron et al., 1963)
- **BCPL** Basic CPL (Richards, 1969)
- **B** (Thompson, 1970)
- **C** K&R C (Ritchie, 1972)
- **ANSI C** American National Standards Institute C (X3J11, 1989)
- **C99** (JTC1/SC22/WG14, ISO/IEC 9899, 1999)

Chương trình C đầu tiên

Hello World — Chào thế giới

Giải thuật

output “Hello World!”

Chương trình C

```
#include <stdio.h>

int main()
{
    printf("Hello World!");

    return 0;
}
```

Cấu trúc căn bản chương trình C

Khai báo tệp tiêu đề mô tả thư viện hàm vào ra

Hàm thực hiện chính của chương trình

Bắt đầu hàm chính

Lệnh in ra màn hình

Kết thúc hàm chính

Chương trình C

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

```
int main()
```

```
{
```

```
    printf("Hello World!");
```

```
    return 0;
```

```
}
```

Cú pháp của C

- Chương trình được viết dựa trên
 - **Các từ khóa**: là các từ được định nghĩa từ trước dành riêng cho ngôn ngữ, ví dụ: main, if, do, while, ...
 - **Các bộ kí tự dấu**: sử dụng với mục đích đặc thù trong chương trình như tạo một khối lệnh { }, tạo một chuỗi kí tự “”, ...
 - **Các tên định danh**: do người sử dụng khai báo dùng đại diện cho một biến hay một chương trình con trong chương trình

Từ khoá C

- điều khiển luồng: `if, else, return, switch, case, default`
- điều khiển lặp : `for, do, while, break, continue`
- Kiểu: `int, long, short, float, double, char, void, signed, unsigned, static, const`
- Cấu trúc: `struct, typedef, union, enum, sizeof`
- Từ khoá khác: `extern, auto, register, volatile, main`
- Chú ý: Trong C phân biệt giữa chữ hoa và chữ thường

Các bộ dấu thường dùng

- {...} Tạo một khối lệnh của chương trình
- “...” Tạo một chuỗi kí tự cần hiển thị
- /* ... */ Tạo chú thích trong chương trình
- ; Dấu kết thúc một lệnh
- ... và các dấu của một biểu thức như +, -, *, /, (), ...

Tên định danh

- Khi khai báo một biến hay một chương trình con người lập trình cần phải đặt tên cho nó.
- Quy tắc đặt tên của người sử dụng
 - Chỉ dùng chữ cái, chữ số và kí tự gạch nối (_) để đặt tên
 - Tên phải bắt đầu bằng chữ cái
 - Phân biệt chữ hoa và chữ thường trong các tên
- Những tên nào sau đây là hợp lệ
 - tong, 2k, trung binh, lon_nhat, u2, %totnghiep

Ví dụ viết chương trình

In các số từ 0 đến 9

```
đặt dem = 0
while (dem nhỏ hơn 10)
do
{
    output dem
    tăng 1 vào dem
}
```

```
int main()
{
```

Tạo hàm chính cho chương trình

```
    return 0;
}
```

Ví dụ (tiếp)

In các số từ 0 đến 9

```
đặt dem = 0
while (dem nhỏ hơn 10)
do
{
    output dem
    tăng 1 vào dem
}
```

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

```
int main()
{
```



Khai báo tiêu đề

```
    return 0;
```

```
}
```

Ví dụ (tiếp)

In các số từ 0 đến 9

```
đặt dem = 0
while (dem nhỏ hơn 10)
do
{
    output dem
    tăng 1 vào dem
}
```

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

```
/* In tu 0 toi 9 */
```

```
int main()
{
```



Chú thích

```
    return 0;
```

```
}
```

Ví dụ (tiếp)

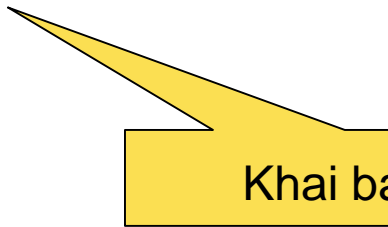
In các số từ 0 đến 9

```
đặt dem = 0
while (dem nhỏ hơn 10)
do
{
    output dem
    tăng 1 vào dem
}
```

```
#include <stdio.h>

/* In tu 0 toi 9 */
int main()
{
    int dem;

    return 0;
}
```



Khai báo biến

Ví dụ (tiếp)

In các số từ 0 đến 9


```
đặt dem = 0
while (dem nhỏ hơn 10)
do
{
    output dem
    tăng 1 vào dem
}
```

```
#include <stdio.h>

/* In tu 0 toi 9 */
int main()
{
    int count;

    dem = 0;

    return 0;
}
```



Gán giá trị cho biến

Ví dụ (tiếp)

In các số từ 0 đến 9

```
đặt dem = 0
while (dem nhỏ hơn 10)
do
{
    output dem
    tăng 1 vào dem
}
```

```
#include <stdio.h>

/* In tu 0 toi 9 */
int main()
{
    int dem;

    dem = 0;
    while ( dem < 10 )
    {

    }
    return 0;
}
```

Tạo vòng lặp

Ví dụ (tiếp)

In các số từ 0 đến 9

```
đặt dem = 0
while (dem nhỏ hơn 10)
do
{
    output dem
    tăng 1 vào dem
}
```

```
#include <stdio.h>

/* In tu 0 toi 9 */
int main()
{
    int dem;

    dem = 0;
    while ( dem < 10 )
    {
        printf ("%d\\n", dem) ;

    }
    return 0;
}
```

Ví dụ (tiếp)

In các số từ 0 đến 9

```
đặt dem = 0
while (dem nhỏ hơn 10)
do
{
    output dem
    tăng 1 vào dem
}
```

```
#include <stdio.h>

/* In tu 0 toi 9 */
int main()
{
    int dem;

    dem = 0;
    while ( dem < 10 )
    {
        printf("%d\\n", dem) ;
        dem = dem + 1;
    }
    return 0;
}
```

Chương trình sau làm gì?

```
#include <stdio.h>

int main() {
    float num;

    printf("Enter a number: ");
    scanf("%f", &num);

    if ( num < 0 ) {
        printf("%f is negative", num);
    } else {
        printf("%f is positive", num);
    }

    return 0;
}
```

Giải thuật của chương trình

/ Tìm dấu của một số */*

input num

if (num < 0) then

{

 output “số âm”

}

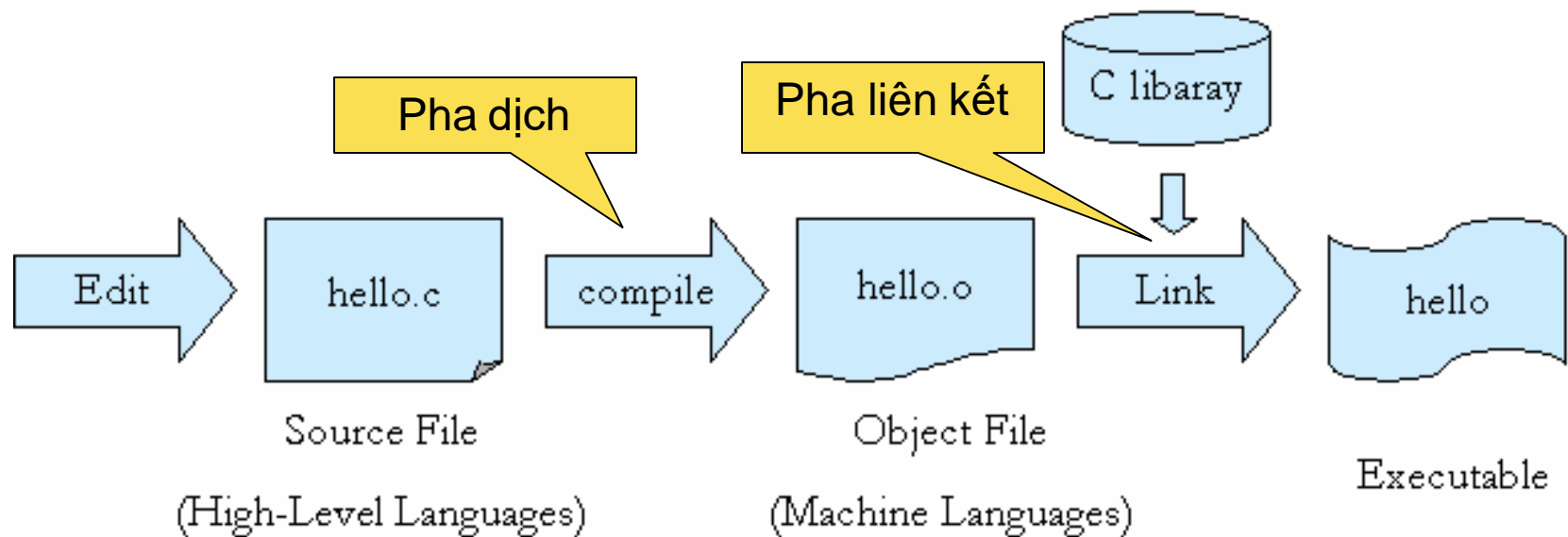
else

{

 output “số dương”

}

Biên dịch chương trình C



Lỗi có thể xảy ra tại pha dịch hay pha liên kết

Trình biên dịch

- để dịch chương trình thì cần phải có trình biên dịch, ví dụ: gcc
- Trình biên dịch C luôn hỗ trợ các tham số để thực hiện 2 pha của quá trình dịch.
- Ví dụ gcc -c để thực hiện pha dịch, và gcc -o để thực hiện pha liên kết.
- Có thể sử dụng một câu lệnh để thực hiện đồng thời cả 2 pha trên một tệp chương trình nguồn đơn.
 - \$gcc -o <tệp đích> <tệp nguồn>
 - VD: \$gcc -o hello hello.c

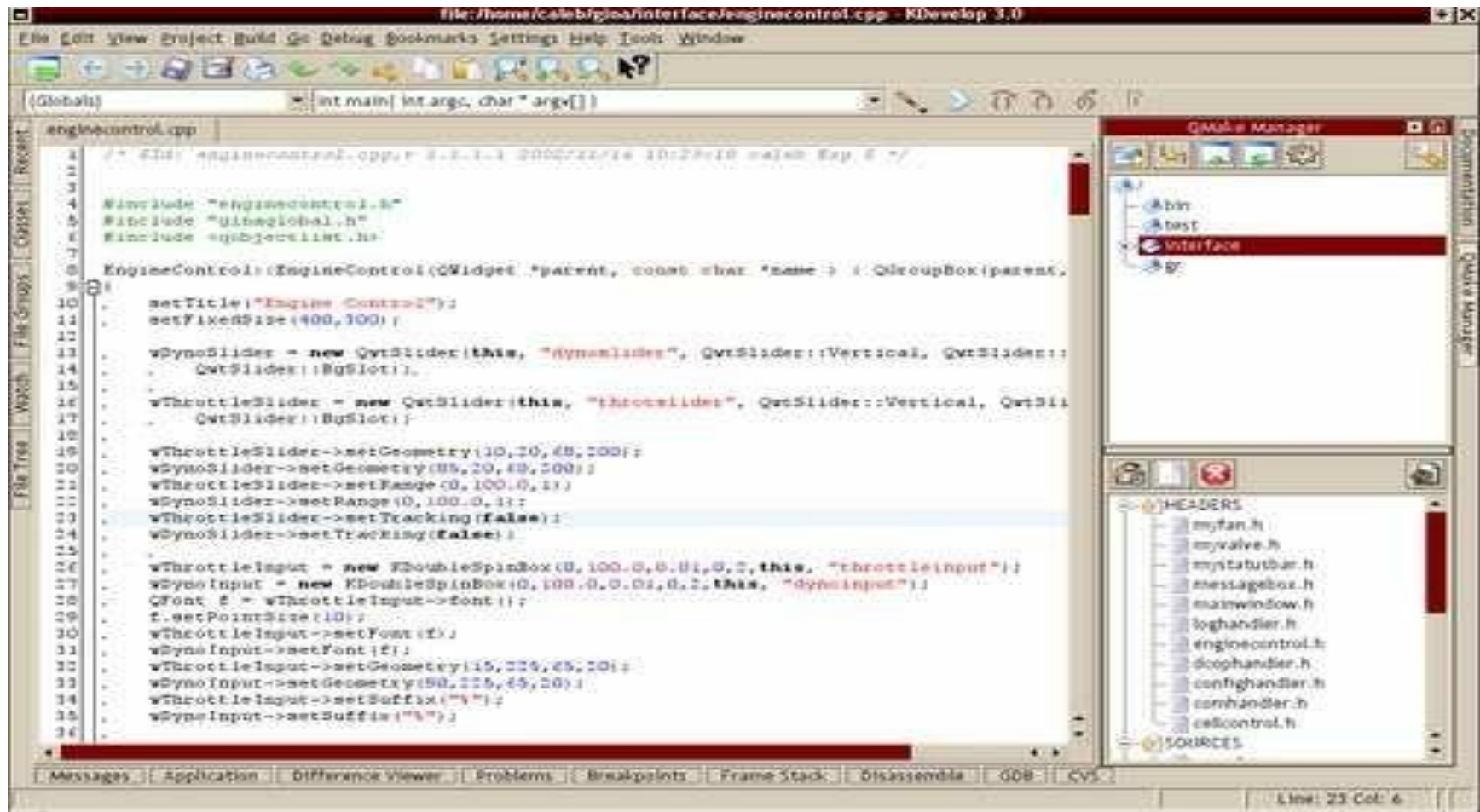
IDE: Môi trường hỗ trợ lập trình

- Lập trình là quá trình thực hiện lặp đi lặp lại các thao tác: soạn mã nguồn, thực hiện dịch, chạy tìm lỗi và sửa chương trình
- Các thao tác này hoàn toàn có thể thực hiện một cách độc lập bởi các công cụ khác nhau: ví dụ soạn thảo bằng emacs, dịch chương trình bằng gcc.
- Tuy nhiên có một cách thuận lợi hơn là tích hợp các công cụ liên quan đến lập trình vào một môi trường duy nhất để hỗ trợ việc lập trình dễ dàng hơn. Môi trường như thế này được gọi là IDE và thường bao gồm 3 bộ công cụ chính là: trình biên soạn, trình biên dịch và trình gỡ rối.

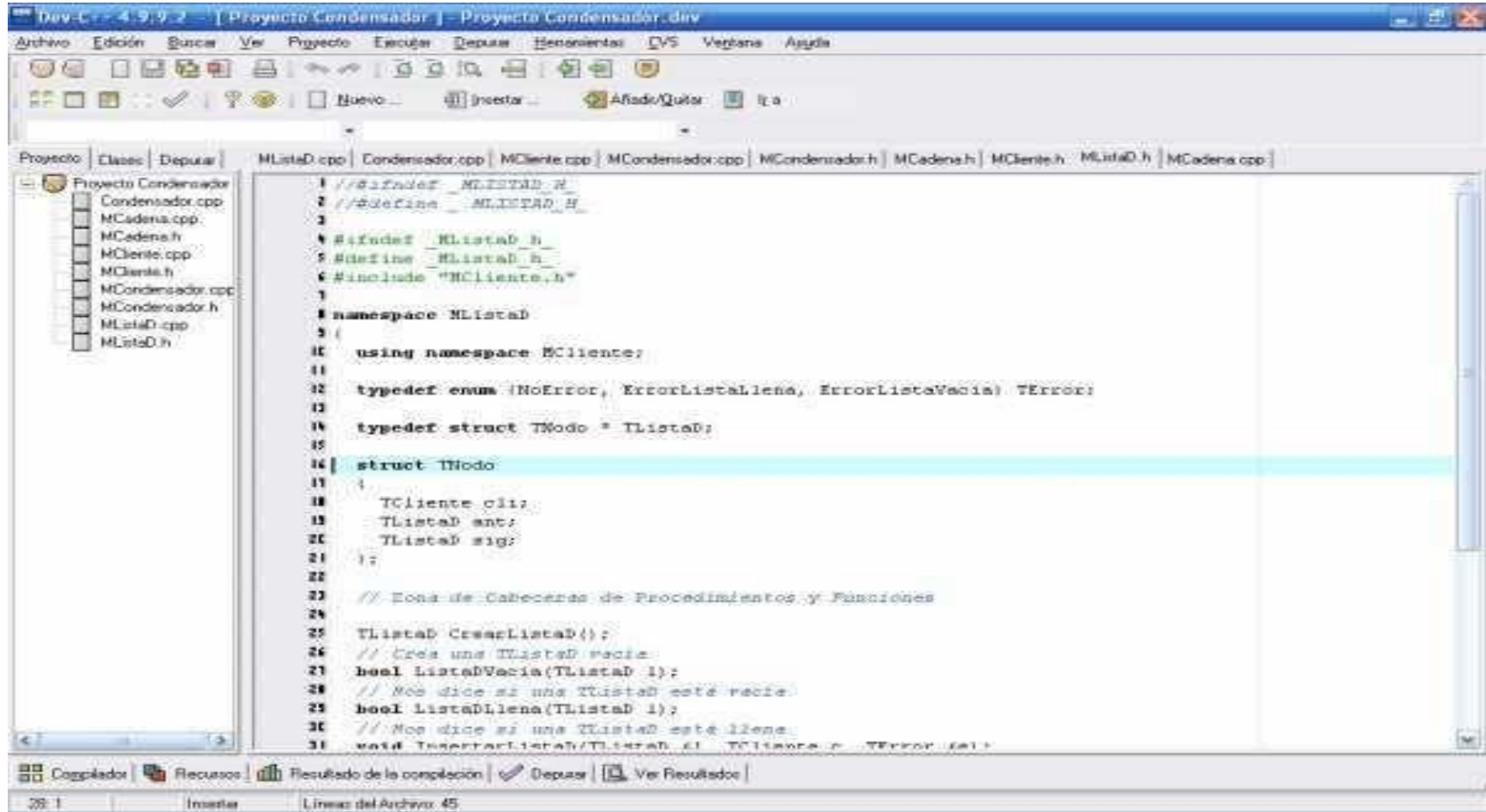
Sản phẩm IDE

- Trong Linux:
 - KDevelop
- Trong Window:
 - Dev-C++,
 - Turbo C++,
 - Visual C++,
 - etc

KDevelop



Dev-C++



Visual C++

