

# Tìm hiểu tuần 1

## ▼ Lập trình C cơ bản

### 1. Ngôn ngữ C là gì, ưu nhược điểm, phạm vi ứng dụng ?

- Ngôn ngữ C là ngôn ngữ lập trình rất phổ biến, là ngôn ngữ cơ bản trong lĩnh vực khoa học máy tính. C là một ngôn ngữ lập trình mạnh mẽ, hiệu suất cao, C được liên kết chặt chẽ với UNIX, vì nó được phát triển để viết hệ điều hành UNIX
- Ưu điểm:
  - Là 1 ngôn ngữ mạnh mẽ và hiệu quả: nó chứa nhiều kiểu dữ liệu và toán tử. Có khả năng tương tác gần gũi với cấu trúc máy tính.
  - Tạo nền tảng cho nhiều ngôn ngữ lập trình khác: C++, C#
  - C rất linh hoạt, độc lập với máy
  - Là ngôn ngữ lập trình có cấu trúc: giúp kiểm tra và bảo trì dễ dàng, đơn giản hơn
  - Có thể tự do cấp phát bộ nhớ động tại thời điểm chạy.
  - Dùng trong lập trình hệ thống
- Nhược điểm:
  - Không hỗ trợ khái niệm về OOP (kế thừa, đa đỉnh, đóng gói, trừu tượng, ẩn dữ liệu)
  - Không hiển thị lỗi sau mỗi dòng mã, trình biên dịch sẽ hiển thị tất cả các lỗi sau khi viết chương trình. Khiến cho việc kiểm tra mã rất phức tạp trong các chương trình lớn.
  - Thiếu xử lý ngoại lệ: C không cho phép bắt lỗi và thực hiện các phản hồi thích hợp
  - C không có bất kỳ hàm tạo hoặc hàm hủy nào
- Phạm vi ứng dụng:
  - Phát triển hệ điều hành, hầu hết những hệ điều hành phổ biến hiện nay đều được viết bằng C: Linux, Windows, MacOS, UNIX.
  - Phát triển phần mềm hệ thống: chẳng hạn như trình quản lý cơ sở dữ liệu (MySQL, SQLite), trình quản lý tập tin và các thành phần quan trọng của hệ thống máy tính

- Phát triển phần mềm nhúng
- Được sử dụng để viết thư viện và framework
- Thường được sử dụng trong các ứng dụng liên quan đến tính toán khoa học và phân tích số liệu: ví dụ như các thư viện OpenCV, TensorFlow cung cấp API cho C/C++ để phát triển các ứng dụng trong lĩnh vực khoa học dữ liệu và máy học

## 2. Biến, hằng, toán tử, macro, kiểu dữ liệu

- Biến
  - Biến trong C là 1 định danh được sử dụng để lưu trữ giá trị dữ liệu. Mỗi biến có 1 kiểu dữ liệu và tên riêng biệt để xác định nó
  - Khai báo biến: bao gồm kiểu dữ liệu và tên biến
  - Khởi tạo biến
  - Tên biến: là 1 định danh duy nhất được sử dụng để xác định biến, tên biến phải tuân theo các quy tắc về cú pháp và không trùng với từ khoá
  - Phạm vi biến: xác định nơi mà biến có thể được truy cập
  - + Biến cục bộ: truy cập trong phạm vi của hàm hoặc khối mã trong đó chúng được khai báo
  - + Biến toàn cục: có thể truy cập từ mọi nơi trong chương trình
- Hằng
  - Nếu muốn khởi tạo 1 biến có giá trị không thể thay đổi, có 2 cách:
    - + sử dụng `#define`
    - + sử dụng `const`
- Toán tử: được sử dụng để thực hiện các thao tác trên các biến và giá trị
  - C chia các toán tử thành các nhóm:
    - + toán tử số học
    - + toán tử gán
    - + toán tử so sánh
    - + toán tử logic
    - + toán tử bitwise
- Macro: là 1 tên bất kì (do người viết mã đặt tên) trỏ tới 1 khối lệnh thực hiện 1 chức năng nào đó. Trong quá trình tiền xử lý, các macro được sử dụng trong chương trình được thay thế bởi các khối câu lệnh tương ứng

- Kiểu dữ liệu
  - C có 1 loạt kiểu dữ liệu có thể sử dụng để định nghĩa và lưu trữ dữ liệu.
  - VD: int, float, double, char ,...

### 3. Các cấu trúc điều kiện, vòng lặp

- if...else
 

```
if (đk) {
    //thực hiện nếu đk đúng
} else {
    // thực hiện nếu đk là sai
}
```
- if...else if ... else
 

```
if (đk1) {
    //thực hiện nếu đk1 đúng
} else if (đk2) {
    //thực hiện nếu đk2 đúng
} else {
    // thực hiện nếu cả 1 và 2 không đúng
}
```
- switch: cho phép kiểm tra 1 biểu thức hoặc giá trị và thực hiện các tác vụ tương ứng với từng giá trị
 

```
switch (bieuthuc) {
    case gt1:
        //thực hiện nếu bieuthuc bằng gt1
        break;
    case gt2:
        //thực hiện nếu bieuthuc bằng gt2
        break;
    default:
        // thực hiện nếu không có trường hợp nào
}
```
- vòng lặp for: sử dụng khi số lần lặp xác định
 

```
for (khoi_tao; đk; cap_nhat) {
    //thực hiện
}
```

- vòng lặp while: sử dụng khi không biết trước số lần lặp và muốn thực hiện lặp dựa trên điều kiện nào đó  

```
while (dk) {
//khối mã thực hiện trong vòng lặp
}
```
- do .... while: sử dụng khi muốn khối mã được thực hiện ít nhất 1 lần, sau đó mới kiểm tra điều kiện xem có tiếp tục hay không  

```
do {
//khối mã thực hiện ít nhất 1 lần
} while (dk);
```

#### 4. struct, union

- Struct trong C là một cấu trúc dữ liệu cho phép tổng hợp nhiều biến có kiểu dữ liệu khác nhau vào 1 đối tượng duy nhất  
VD: struct SV {  
         string hoten;  
         int mssv;  
 }  
 - Truy xuất các thuộc tính  
 + sử dụng . để truy xuất tới thuộc tính khi khai báo biến bình thường  
 + sử dụng → để truy xuất tới thuộc tính khi biến là con trỏ
- union

#### 5. Con trỏ, mảng

- Con trỏ: là 1 biến đặc biệt được sử dụng để lưu địa chỉ bộ nhớ của biến khác. Cho phép truy cập và thao tác trực tiếp với bộ nhớ và dữ liệu  
 - khai báo: dùng \* sau kiểu dữ liệu  
 - sử dụng toán tử & để lấy địa chỉ của biến  
 - truy cập giá trị mà con trỏ đang trỏ đến: dùng toán tử \*  
 VD: int x = \*ptr;  
 - sử dụng con trỏ để thay đổi giá trị biến  
 VD: int x = 0;  
     int \*ptr = &x;  
     \*ptr = 20; //thay đổi giá trị biến x thành 20

- Mảng 1 chiều
  - Là tập hợp các phần tử có cùng kiểu dữ liệu được sắp xếp theo thứ tự. Cho phép lưu trữ nhiều giá trị cùng loại trong 1 biến duy nhất
  - khai báo: `kiểu_du_lieu ten_mang[kich_thuoc];`
  - truy cập phần tử trong mảng: bằng cách sử dụng index của phần tử trong dấu ngoặc vuông []
- Mảng 2 chiều: là một cấu trúc dữ liệu cho phép lưu trữ dữ liệu dưới dạng ma trận

## 6. Hàm, tham chiếu, tham trị

- Hàm: là một khối mã chỉ chạy được khi gọi. Có thể truyền dữ liệu (truyền tham số) vào một hàm. Các hàm được sử dụng để thực hiện 1 số hành động nhất định, chúng rất quan trọng để sử dụng lại
- Các hàm đã được xác định trước: ví dụ `main()`, `printf()`
- tham số và đối số

```
void myName (char[]) {
    printf("Hello %s\n", name);
}
```

```
int main() {
    myName("A");
    myName("B");
    return 0;
}
```

trong ví dụ trên thì `name` là tham số còn `A B` là đối số

- Tham trị:
  - Truyền tham trị là truyền cho đối số 1 bản sao
  - Ví dụ `x` xuất hiện ở hàm bất kỳ và hàm `main` thì `x` ở 2 hàm này độc lập với nhau
- Tham chiếu:
  - Truyền tham chiếu: truyền cho đối số 1 bản gốc thông qua toán tử `&`
  - Ví dụ `x` xuất hiện ở hàm bất kỳ và hàm `main` thì 2 biến `x` này là một.

## ▼ Embedded C

1. Sự giống và khác nhau giữa C thông thường và Embedded C
2. Lưu ý khi lập trình C embedded
  - ROM và RAM hạn chế
  - Lập trình phụ thuộc phần cứng
  - Cần đáp ứng chính xác về thời gian
  - Nhiều kiểu Pointer
3. C memory management
4. Compilation process, toolchain, compiler, linker
  - toolchain: là tập hợp các tool dùng để biên dịch Source code thành các file có thể chạy được trên thiết bị thật. Toolchain bao gồm: compiler, linker, ....
  - compiler: chuyển ngôn ngữ bậc cao (C) về ngôn ngữ mà máy có thể đọc được
  - linker: là phần mềm để xây hệ thống bằng cách kết nối các thành phần của software lại nhau
5. Make, build system
6. 8bit/ 16bit/ 32bit machine
  - 8 bit
7. Tổ chức firmware, kiến trúc phần mềm
  - firmware
    - firmware là một phần mềm được thiết kế riêng và nhúng lên trên một và một số loại phần cứng cố định
    - Chức năng chính: kiểm soát, giám sát, thao tác dữ liệu, giao tiếp với phần cứng
    - Firmware có thể giao tiếp trực tiếp với phần cứng có thông qua/ không thông qua hệ điều hành
  - Kiến trúc phần mềm
    - Phần mềm hệ thống nhúng là các chương trình được viết, thiết kế, lập trình chuyên biệt cho các thiết bị số và hoạt động song song với các thiết bị đó. Hiện nay có 6 loại kiến trúc phần mềm hệ thống nhúng phổ biến là:

- + vòng lặp kiểm soát đơn giản
- + hệ thống ngắt điều khiển
- + đa nhiệm tương tác
- + đa nhiệm ưu tiên
- + vi nhân và nhân ngoại
- + nhân khối

▼ Embedded C for 8051

em bổ sung sau ạ ;<

▼ Layout PCB

1.n

2. Thực hiện vẽ Kit 8051 với các ngoại vi như: GPIO, LED, BUTTON, UART, khối nạp