

TIN HỌC ĐẠI CƯƠNG

Bài 7. Tổng quan về ngôn ngữ C

Bùi Trọng Tùng, SoICT, HUST

Nội dung

1. Lịch sử phát triển
2. Các phần tử cơ bản của ngôn ngữ C
3. Cấu trúc cơ bản của chương trình C
4. Biên dịch chương trình C

Nội dung

- ⇒ 1. Lịch sử phát triển
- 2. Các phần tử cơ bản của ngôn ngữ C
- 3. Cấu trúc cơ bản của chương trình C
- 4. Biên dịch chương trình C

3

1. Lịch sử phát triển

- Ra đời tại phòng thí nghiệm BELL của tập đoàn AT&T (Hoa Kỳ)
- Do Brian W. Kernighan và Dennis M. Ritchie phát triển vào đầu 1970, hoàn thành 1972
- C dựa trên nền các ngôn ngữ BCPL (*Basic Combined Programming Language*) và ngôn ngữ B.
- Tên là ngôn ngữ C như là sự tiếp nối ngôn ngữ B.

4

Lịch sử phát triển

- Đặc điểm của ngôn ngữ lập trình C
 - Ngôn ngữ lập trình hệ thống
 - Tính khả chuyển, linh hoạt cao
 - Có thể mạnh trong xử lý dữ liệu số, văn bản, cơ sở dữ liệu
- C thường được sử dụng để viết các chương trình hệ thống
 - Hệ điều hành Unix có 90% mã C, 10% hợp ngữ
 - Các trình điều khiển thiết bị (device driver)
 - Xử lý ảnh...

5 5

Lịch sử phát triển

- 1978: C được giới thiệu trong phiên bản đầu của cuốn sách "*The C programming language*"
- Sau đó, C được bổ sung thêm những tính năng và khả năng mới → Đồng thời tồn tại nhiều phiên bản nhưng không tương thích nhau.
- Năm 1989, Viện tiêu chuẩn quốc gia của Hoa Kỳ (*American National Standards Institute - ANSI*) đã công bố phiên bản chuẩn hóa của ngôn ngữ C: *ANSI C* hay *C chuẩn* hay C89

6

Lịch sử phát triển

- Các phiên bản ngôn ngữ C
 - ANSI C: C chuẩn (1989)
 - Các phiên bản khác thường bổ sung thêm thư viện của ANSI C
- Hiện nay cũng có nhiều phiên bản của ngôn ngữ C khác nhau, gắn liền với một bộ chương trình dịch cụ thể của ngôn ngữ C
 - Turbo C++ và Borland C++ của Borland Inc.
 - MSC và VC của Microsoft Corp.
 - GCC của GNU project...

7 7

Nội dung

1. Lịch sử phát triển
- ⇒ 2. Các phần tử cơ bản của ngôn ngữ C
3. Cấu trúc cơ bản của chương trình C
4. Biên dịch chương trình C

8

Một chương trình đơn giản

```
#include<stdio.h>
#include<conio.h>
int main(){
    printf("Hello World\n");
    getch();
    return 0;
}
```

9

2.1. Tập ký tự

- Tập ký tự là tập các phần tử cơ bản tạo nên chương trình
 - Tổ hợp các ký tự → từ
 - Liên kết các từ theo cú pháp → câu lệnh
 - Tổ chức các câu lệnh → chương trình
- Ví dụ:
 - include, void, main...
 - printf("..."), getch();

10

Tập ký tự

- Tập ký tự trong C
 - 26 chữ cái hoa: A B C ... X Y Z
 - 26 chữ cái thường: a b c ... x y z.
 - 10 chữ số: 0 1 2 3 4 5 6 7 8 9.
 - Các kí hiệu toán học: + - * / = < >
 - Các dấu ngăn cách: . ; , : space tab
 - Các dấu ngoặc: () [] { }
 - Các kí hiệu đặc biệt: _ ? \$ & # ^ \ ! ' " ~ .v.v.

11

2.2. Từ khóa

- Từ khóa (keyword)
 - Có sẵn trong mỗi ngôn ngữ lập trình
 - Dành riêng cho các mục đích xác định
 - Đặt tên cho kiểu dữ liệu: **char, int, float, double...**
 - Mô tả các lệnh, các cấu trúc lập trình: **if, else, switch, case, while, do...**
- Chú ý:
 - *Tất cả từ khóa trong C đều viết bằng chữ cái thường*

12

Từ khóa

- Từ khóa hay dùng trong C

break	case	char	const	continue	default
do	double	else	enum	float	for
goto	if	int	interrupt	long	return
short	signed	sizeof	static	struct	switch
typedef	union	unsigned	void	while	

13

2.3. Định danh

- Định danh (*Identifier* – hoặc còn gọi là *Tên*) là một dãy các kí tự dùng để gọi tên các đối tượng trong chương trình.
- Các đối tượng trong chương trình
 - Biến
 - Hằng số
 - Hàm
 - Kiểu dữ liệu... (sẽ làm quen ở các mục sau)
- Định danh có thể được đặt bởi
 - Ngôn ngữ lập trình → các từ khóa
 - Người lập trình

14

Định danh

- Quy tắc đặt tên định danh trong C
 - Các kí tự được sử dụng: chữ cái, chữ số và dấu gạch dưới “_” (*underscore*)
 - Không dùng dấu cách (space)
 - Bắt đầu của định danh phải là chữ cái hoặc dấu gạch dưới “_”, không được bắt đầu định danh bằng chữ số.
 - Định danh do người lập trình đặt không được trùng với các từ khóa của C
- Chú ý: *C là ngôn ngữ có phân biệt chữ hoa và chữ thường*

15

Định danh

- Ví dụ
 - Định danh hợp lệ:
i, x, y, a, b, _function, _MY_CONSTANT, PI, gia_tri_1
 - Định danh không hợp lệ
 - 1_a, 3d, 55x (bắt đầu bằng chữ số)
 - so luong, ti le (có dấu cách - kí tự không hợp lệ)
 - int, char (trùng với từ khóa của ngôn ngữ C)

16

Định danh

- Quy tắc con lặc đà:
 - Định danh có tính gọi nhớ
 - Tên hằng số: viết hoa toàn bộ, sử dụng dấu gạch dưới để phân cách các từ
 - Tên biến, tên hàm: viết thường từ đầu tiên, viết hoa chữ cái đầu các từ tiếp theo
 - Tên kiểu dữ liệu mới: viết hoa chữ cái đầu tiên tất cả các từ

Định danh	Loại đối tượng
HANG_SO_1, CONSTANT_2	Hằng số
a, b, i, j, boDem, danhSach	Biến
nhapDuLieu, hienThi	Hàm
SinhVien, MatHang	Kiểu dữ liệu mới

17

Các kiểu dữ liệu

- Định nghĩa:
 - Kiểu dữ liệu cho biết cách lưu trữ dữ liệu trong bộ nhớ, dải giá trị và các phép toán có thể thực hiện trên dữ liệu.
- Ví dụ:
 - Kiểu dữ liệu **int** (số nguyên) trong C
 - Một dữ liệu thuộc kiểu dữ liệu int
 - Là một số nguyên có dấu
 - 2 byte
 - Nhận giá trị từ $-32,768$ (-2^{15}) đến $32,767$ ($2^{15} - 1$)

18

Các kiểu dữ liệu

- Ví dụ (tiếp)
 - Một số phép toán được định nghĩa trên kiểu dữ liệu int của C

Tên phép toán	Ký hiệu
Đảo dấu	-
Cộng	+
Trừ	-
Nhân	*
Chia lấy phần nguyên	/
Chia lấy phần dư	%
So sánh	>, <, >=, <=, ==, !=

19

Ví dụ

```
#include<stdio.h>
#include<conio.h>
#define PI 3.14
int main(){
    float r, s;
    printf("Nhap ban kinh hinh tron:");
    scanf("%f",&r);
    s = PI * r * r;
    printf("Dien tich hinh tron %f\n",s);
    getch();
    return 0;
}
```

20

2.5. Hằng số

- Định nghĩa:
 - hằng (*constant*) là đại lượng có giá trị không đổi trong chương trình.
- Biểu diễn hằng số nguyên: trong C, một hằng số nguyên có thể biểu diễn dưới 3 dạng
 - Dạng thập phân
 - Dạng thập lục phân
 - Dạng bát phân

Giá trị thập phân	Giá trị thập lục phân	Giá trị bát phân
2007	0x7D7	03727
396	0x18C	0614

21

Hằng số

- Biểu diễn hằng số thực: trong C, một hằng số thực có thể biểu diễn dưới 2 dạng
 - Dạng số thực dấu phẩy tĩnh
 - Dạng số thực dấu phẩy động
- Ví dụ

Số thực dấu phẩy tĩnh	Số thực dấu phẩy động
3.14159	31.4159 E-1
123.456	12.3456 E+1 hoặc 1.23456 E+2

22

Hằng số

- Biểu diễn hằng ký tự: trong C, một hằng ký tự có thể biểu diễn theo hai cách
 - Bằng ký hiệu của ký tự đặt giữa hai dấu nháy đơn
 - Bằng số thứ tự của ký tự đó trong bảng mã ASCII (số nguyên -> tuân thủ quy tắc biểu diễn hằng số nguyên)
- Ví dụ

Ký tự cần biểu diễn	Cách 1	Cách 2
Chữ cái A	'A'	65, 0x41, 0101
Dấu nháy đơn	'\''	39, 0x27, 047
Ký tự tab	'\t'	0, 0x09, 011

23

Hằng số

- Biểu diễn hằng chuỗi ký tự:
 - Hằng chuỗi ký tự được biểu diễn bởi dãy các ký tự thành phần có trong chuỗi đó và được đặt trong cặp dấu nháy kép.
- Ví dụ:
 - “ngon ngu lap trinh C”
 - “Tin hoc dai cuong”
 - “Dai hoc Bach Khoa Ha Noi”

24

Ví dụ

```
#include<stdio.h>
#include<conio.h>
int main(){
    float a, b, x;
    printf("Nhap he so a=");scanf("%f",&a);
    printf("Nhap he so b khac 0, b=");
    scanf("%f",&b);
    x = -b/a;
    printf("Nghiem don cua phuong trinh %f",x);
    getch();
    return 0;
}
```

25

2.6. Biến

- Định nghĩa:
 - Biến (*variable*) là đại lượng mà giá trị có thể thay đổi trong chương trình.
- Chú ý:
 - Hằng số và biến được sử dụng để lưu trữ dữ liệu trong chương trình
 - Hằng số và biến phải thuộc một kiểu dữ liệu nào đó
 - Hằng số và biến đều phải đặt tên theo quy tắc

26

2.7. Hàm

- Mô tả:

- Hàm (function) là một chương trình con có chức năng nhận dữ liệu đầu vào (các tham số đầu vào), thực hiện một chức năng nào đó và đưa ra các kết quả

Hàm	Ý nghĩa	Ký hiệu toán học	Ví dụ
pow(x,y)	x mũ y	x^y	pow(2,3)=8
sin(x)	sin của x	sinx	sin(0)=0
cos(x)	cos của x	cosx	cos(0)=1

27

2.8. Biểu thức

- Định nghĩa:

- Biểu thức là sự ghép nối các toán tử (*operator*) và các toán hạng (*operand*) theo một quy tắc xác định.
- Các toán hạng có thể là biến, hằng
- Các toán tử rất đa dạng: cộng, trừ, nhân, chia..

- Ví dụ: biểu thức tính thể tích hình hộp chữ nhật

chieuDai * chieuRong * chieuCao

- chieuDai, chieuRong, chieuCao là các hằng hoặc biến số đóng vai trò toán hạng
- Phép * đóng vai trò toán tử

28

2.9. Câu lệnh

- Câu lệnh (*statement*) diễn tả một hoặc một nhóm các thao tác trong giải thuật.
- Chương trình được tạo thành từ dãy các câu lệnh.
- Cuối mỗi câu lệnh bắt buộc có dấu chấm phẩy ';' để đánh dấu kết thúc câu lệnh

29

Câu lệnh

- Phân nhóm:
 - Nhóm các câu lệnh đơn: những câu lệnh không chứa câu lệnh khác. Ví dụ: phép gán, phép cộng, phép trừ...
 - Nhóm các câu lệnh phức: những câu lệnh chứa câu lệnh khác.
 - Ví dụ: lệnh khối đặt trong cặp ngoặc nhọn { }

30

2.10. Chú thích

- Chú thích (comment):
 - Lời mô tả, giải thích vấn đề cho một câu lệnh, một đoạn chương trình hoặc cả chương trình
 - Giúp việc đọc và hiểu chương trình dễ dàng hơn
 - Chú thích không phải là câu lệnh -> không ảnh hưởng tới chương trình
- Cách viết chú thích: trong C có hai cách
 - Chú thích một dòng: sử dụng //
 - Chú thích nhiều dòng: sử dụng /* và */

31

Nội dung

1. Lịch sử phát triển
2. Các phần tử cơ bản của ngôn ngữ C
- ⇒ 3. Cấu trúc cơ bản của chương trình C
4. Biên dịch chương trình C

32

Ví dụ

```
#include <stdio.h> //Khai bao tep tieu de
int main(){
    // Khai bao cac bien
    int a, b;
    int tong, hieu, tich;

    // Nhap vao tu ban phim 2 so nguyen
    printf("\nNhap vao so nguyen thu nhat: ");
    scanf("%d",&a);
    printf("\n Nhap vao so nguyen thu hai: ");
    scanf("%d",&b);
```

33

Ví dụ

```
// Tinh tong, hieu, tich cua 2 so do
tong = a+b; hieu = a - b;tich = a*b;

// Hien thi cac gia tri ra man hinh
printf("\n Tong cua 2 so vua nhap la %d", tong);
printf("\n Hieu cua 2 so vua nhap la %d", hieu);
printf("\n Tich cua 2 so vua nhap la %d", tich);

// Yeu cau nguoi dung an phim bat ki
getch();
return 0;
}
```

34

3. Cấu trúc cơ bản của chương trình C

- **Gồm 6 phần có thứ tự như sau:**

Phần 1: Khai báo tệp tiêu đề: #include
Phần 2: Định nghĩa kiểu dữ liệu mới: typedef ...
Phần 3: Khai báo các hàm nguyên mẫu
Phần 4: Khai báo các biến toàn cục
Phần 5: Hàm main()
Phần 6: Nội dung các hàm đã khai báo

35

Cấu trúc cơ bản của chương trình C

- Phần 1: Khai báo tệp tiêu đề:
 - Thông báo cho chương trình dịch biết là chương trình có sử dụng những thư viện nào.
 - VD: **#include** <stdio.h>
#include <conio.h>

Thư viện	Chức năng
stdio.h	Các hàm vào/ra cơ bản
conio.h	Các hàm giao tiếp người dùng sử dụng MS-DOS API
math.h	Các hàm toán học cơ bản
ctype.h	Các hàm xử lý ký tự
string.h	Các hàm xử lý chuỗi
stdlib.h	Các hàm tiện ích
time.h	Các hàm điều khiển thời gian

5

Cấu trúc cơ bản của chương trình C

- Phần 2: Định nghĩa các kiểu dữ liệu mới
 - Định nghĩa các kiểu dữ liệu mới (nếu cần) dùng cho cả chương trình.
- Phần 3: Khai báo các hàm nguyên mẫu:
 - Giúp cho chương trình dịch biết được những thông tin cơ bản của các hàm sử dụng trong chương trình.
- Phần 4: Khai báo các biến toàn cục
 - Ví dụ:

```
int a, b;  
int tong, hieu, tich;
```

37

Cấu trúc cơ bản của chương trình C

- Phần 5: Hàm **main()**
 - Khi thực hiện, chương trình sẽ bắt đầu bằng việc thực hiện các lệnh trong hàm **main()**.
 - Trong hàm **main()** có thể có lệnh gọi tới các hàm khác.
- Phần 6: Nội dung của các hàm đã khai báo
 - Cài đặt (viết mã) cho các hàm đã khai báo nguyên mẫu ở phần 3.

38

Viết một chương trình “dễ đọc”

- Chú thích mô tả hàm/chương trình:
 - Chức năng
 - Đầu vào
 - Đầu ra
- Chú thích các biến quan trọng
- Chú thích các bước xử lý dữ liệu theo thuật toán
- Lùi dòng cho các lệnh lồng nhau
- Định danh đối tượng rõ ràng

39

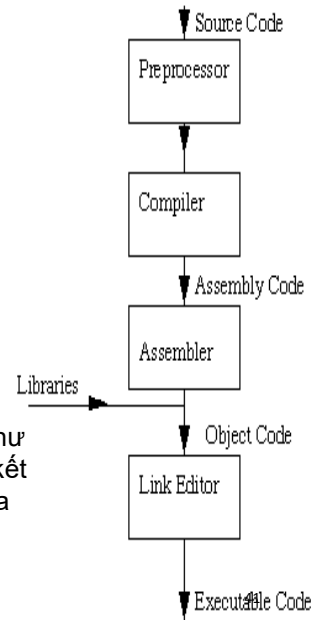
Nội dung

1. Lịch sử phát triển
2. Các phần tử cơ bản của ngôn ngữ C
3. Cấu trúc cơ bản của chương trình C
- ⇒ 4. Biên dịch chương trình C

40

4. Biên dịch chương trình C

- **Preprocessor**
 - Loại bỏ các chú thích
 - Dịch các chỉ thị tiền xử lý bắt đầu là #
- **C Compiler**
 - Biên dịch mã nguồn thành mã assembly.
- **Assembler**
 - Tạo ra mã object.
 - Trên UNIX → file .o
 - Trên MS-DOS → file.OBJ
- **Link Editor**
 - Nếu tệp nguồn tham chiếu đến các hàm thư viện/hàm được định nghĩa thì *Link editor* kết hợp các hàm này với hàm main() để tạo ra tệp có thể thực thi được
 - Trong MS-DOS là file .exe



Thảo luận

