



Chương 1: Tổng quan về ngôn ngữ C



Nội dung chương này

- 1.1. Lịch sử phát triển ngôn ngữ lập trình C
- 1.2. Các phần tử cơ bản của ngôn ngữ C
- 1.3. Cấu trúc cơ bản của một chương trình C
- 1.4. Biên dịch chương trình viết bằng C
- 1.5. Bài tập

2

1.1. Lịch sử phát triển ngôn ngữ lập trình C

- Ngôn ngữ lập trình C (NNLT C) ra đời tại phòng thí nghiệm BELL của tập đoàn AT&T (Hoa Kỳ)
- Do Brian W. Kernighan và Dennis 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.

3

1.1. Lịch sử phát triển ngôn ngữ lập trình C

- Đặc điểm của NNLT C:
 - Là một ngôn ngữ lập trình hệ thống mạnh, khả chuyển, có tính linh hoạt cao.
 - Có thể mạnh trong xử lý các dạng dữ liệu số, văn bản, cơ sở dữ liệu.
 - Thường được sử dụng để viết:
 - Các chương trình hệ thống như hệ điều hành (VD Unix: 90% viết bằng C, 10% viết bằng hợp ngữ).
 - Các chương trình ứng dụng chuyên nghiệp có can thiệp tới dữ liệu ở mức thấp như xử lý văn bản, xử lý ảnh...

4

1.1. Lịch sử phát triển ngôn ngữ lập trình C

- 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 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

5

1.1. Lịch sử phát triển ngôn ngữ lập trình C

- Tất cả các phiên bản của ngôn ngữ C hiện nay đều tuân theo các mô tả đã được nêu ra trong ANSI C, sự khác biệt nếu có thì chủ yếu ở các thư viện bổ sung.
- 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.

6

1.2. Các phần tử cơ bản của ngôn ngữ C

1.2.1. Tập ký tự

- Chương trình C được tạo ra từ các phần tử cơ bản là tập kí tự .
- Các kí tự tổ hợp với nhau tạo thành các từ
- Các từ liên kết với nhau theo một quy tắc xác định để tạo thành các câu lệnh
- Từ các câu lệnh → tổ chức thành chương trình.

7

8

1.2.1. Tập ký tự (tiếp)

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	+ - * / = < >
Các dấu ngăn cách:	. ; , : space tab
Các dấu ngoặc:	() [] { }
Các kí hiệu đặc biệt:	_ ? \$ & # ^ \ ! ' " ~

9

1.2.2. Từ khóa (keyword)

- Là những từ có sẵn của ngôn ngữ và được sử dụng dành riêng cho những mục đích xác định.
- Các từ khóa trong C được sử dụng để
 - Đặt tên cho các kiểu dữ liệu: **int, float, double, char, struct, union...**
 - Mô tả các lệnh, các cấu trúc điều khiển: **for, do, while, switch, case, if, else, break, continue...**

10

1.2.2. Từ khóa (keyword) (tiếp)

Một số từ khóa hay dùng trong Turbo 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	

11

1.2.3. Định danh / tên (identifier)

- Là một dãy các kí tự dùng để gọi tên các đối tượng trong chương trình.
 - Các đối tượng trong chương trình gồm có biến, hằng, hàm, kiểu dữ liệu... ta sẽ làm quen ở những mục tiếp theo.
- Có thể được đặt tên:
 - Bởi ngôn ngữ lập trình (đó chính là các từ khóa)
 - Hoặc do người lập trình đặt.

12

1.2.3. Định danh / tên (identifier) (tiếp)

■ Quy tắc đặt tên:

- Chỉ được gồm có: **chữ cái**, **chữ số** và dấu gạch dưới “_” (*underscore*).
- 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 từ khóa.

13

1.2.3. Định danh / tên (identifier) (tiếp)

■ Ví dụ định danh/tên hợp lệ:

i, x, y, a, b, _function, _MY_CONSTANT, PI, gia_tri_1

■ Ví dụ về định danh/tên không hợp lệ:

1_a, 3d, 55x	bắt đầu bằng chữ số
so luong, ti le	có kí tự không hợp lệ (dấu cách – <i>space</i>) trong tên
int, char	trùng với từ khóa của ngôn ngữ C

14

1.2.3. Định danh / tên (identifier) (tiếp)

■ Cách thức đặt định danh/tên:

- Hằng số: chữ hoa
- Các biến, hàm hay cấu trúc: Bằng chữ thường.
- Nếu tên gồm nhiều từ thì ta nên phân cách các từ bằng dấu gạch dưới.

■ Ví dụ:

Định danh	Loại đối tượng
HANG_SO_1, _CONSTANT_2	hằng
a, b, i, j, count	biến
nhap_du_lieu, tim_kiem, xu_li	hàm
sinh_vien, mat_hang	cấu trúc

15

1.2.4. Các kiểu dữ liệu

- Là một tập hợp các giá trị mà một dữ liệu thuộc kiểu dữ liệu đó có thể nhận được.
- Trên một kiểu dữ liệu ta xác định một số phép toán đối với các dữ liệu thuộc kiểu dữ liệu đó.
- Ví dụ: Trong ngôn ngữ C có kiểu dữ liệu **int**. Một dữ liệu thuộc kiểu dữ liệu **int** thì:
 - Là một số nguyên (*integer*)
 - Có thể nhận giá trị từ - 32768 (- 2¹⁵) đến 32767 (2¹⁵ - 1).

16

1.2.4. Các kiểu dữ liệu (tiếp)

- Trên kiểu dữ liệu **int** ngôn ngữ C định nghĩa các phép toán số học đối với số nguyên như sau:
 - Đảo dấu: **-**
 - Cộng: **+**
 - Trừ: **-**
 - Nhân: *****
 - Chia lấy phần nguyên: **/**
 - Chia lấy phần dư: **%**
 - So sánh bằng: **= =**
 - So sánh lớn hơn: **>**
 - So sánh nhỏ hơn: **<**

17

1.2.5. Hằng số (constant)

- Là đại lượng có giá trị không đổi trong chương trình.
- Để giúp chương trình dịch nhận biết hằng ta cần nắm được cách biểu diễn hằng trong một chương trình C.

18

Biểu diễn hằng số nguyên

- Dạng thập phân:
 - Giá trị số dưới hệ đếm cơ số 10 thông thường
 - Ví dụ: 2007, 396
- Dạng thập lục phân:
 - Giá trị số dưới dạng hệ đếm cơ số 16 và thêm tiền tố **0x**
 - Ví dụ: 0x7D7, 0x18C.
- Dạng bát phân:
 - Giá trị số dưới dạng hệ đếm cơ số 8 và thêm tiền tố **0**
 - Ví dụ: 03727, 0614.

19

Biểu diễn hằng số thực

- Dưới dạng số thực dấu phẩy tĩnh:
Ví dụ: 3.14159 , 123.456
- Dưới dạng số thực dấu phẩy động:
Ví dụ: 31.4159 E -1
12.3456 E +1
1.23456 E +2

20

Biểu diễn hằng ký tự

- Bảng ký hiệu của ký tự đó đặt giữa 2 dấu nháy đơn ('').
- Bảng số thứ tự của ký tự đó trong bảng mã ASCII (và lưu ý số thứ tự của một ký tự trong bảng mã ASCII là một số nguyên nên có một số cách biểu diễn).

21

Biểu diễn hằng ký tự - Ví dụ:

Kí tự cần biểu diễn	Cách 1	Cách 2
Chữ cái A	'A'	65 hoặc 0101 hoặc 0x41
Dấu nháy đơn '	'\''	39 hoặc 047 hoặc 0x27
Dấu nháy kép ''	'\"'	34 hoặc 042 hoặc 0x22
Dấu gạch chéo ngược \	'\\'	92 hoặc 0134 hoặc 0x5c
Kí tự xuống dòng	'\n'	
Kí tự NUL	'\0'	0 hoặc 00 hoặc 0x0

22

Biểu diễn hằng xâu ký tự

- Một hằng là xâu kí tự được biểu diễn bởi dãy các kí tự thành phần có trong xâu đó và được đặt trong cặp dấu nháy kép ("").
- Ví dụ: "Đại học Bách Khoa", "Tin học đại cương", "Nguyễn Hồng Phương",...

23

1.2.6. Biến (variable)

- Là đại lượng mà giá trị có thể thay đổi trong chương trình.
- Hằng và biến được sử dụng để lưu trữ dữ liệu, và phải thuộc một kiểu dữ liệu nào đó.
- Tên biến và hằng được đặt theo quy tắc đặt tên cho định danh.

24

1.2.7. Hàm (function)

- Còn được gọi là chương trình con
- Những đoạn chương trình lặp đi lặp lại nhiều lần ở những chỗ khác nhau → Viết thành hàm để khi cần chỉ cần gọi ra chứ không phải viết lại toàn bộ.
- Giải quyết một bài toán lớn thì chương trình của ta có thể rất lớn và dài → Chia thành các công việc nhỏ hơn được viết thành các hàm.

25

Một số hàm toán học hay dùng trong C

Một số hàm toán học hay được sử dụng trong C

Hàm	Ý nghĩa	Kí hiệu toán học	Ví dụ
sqrt(x)	Căn bậc 2 của x	\sqrt{x}	sqrt(16.0) bằng 4.0
pow(x, y)	x mũ y	x^y	pow(2, 3) bằng 8
exp(x)	e mũ x	e^x	exp(1.0) bằng 2.718282
log(x)	logarithm tự nhiên (cơ số e) của x	$\ln x$	log(2.718282) bằng 1.0

26

Một số hàm toán học hay dùng trong C (tiếp)

log10(x)	logarithm cơ số 10 của x	$\log x$	log10(100) bằng 2
sin(x)	sin của x	$\sin x$	sin(0.0) bằng 0.0
cos(x)	cosin của x	$\cos x$	cos(0.0) bằng 1.0
tan(x)	tang của x	$\tan x$	tan(0.0) bằng 0.0
ceil(x)	phần nguyên giả của x, tức là số nguyên nhỏ nhất không nhỏ hơn x	$\lceil x \rceil$	ceil(2.5) bằng 3 ceil(-2.5) bằng -2
floor(x)	phần nguyên non của x, tức là số nguyên lớn nhất không lớn hơn x	$\lfloor x \rfloor$	floor(2.5) bằng 2 floor(-2.5) bằng -3

27

1.2.8. 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 đều có dấu chấm phẩy (;) để đánh dấu kết thúc câu lệnh.

28

1.2.8. Câu lệnh (tiếp)

- Câu lệnh được chia thành 2 nhóm chính:
 - Nhóm các câu lệnh đơn:
 - Không chứa câu lệnh khác.
 - Ví dụ: phép gán, phép cộng, phép trừ...
 - Nhóm các câu lệnh phức:
 - Chứa câu lệnh khác trong nó.
 - Ví dụ: lệnh khối, các cấu trúc lệnh rẽ nhánh, cấu trúc lệnh lặp...
 - Lệnh khối là một số các lệnh đơn được nhóm lại với nhau và đặt trong cặp dấu ngoặc nhọn { }

29

1.2.9. Chú thích (Comment)

- Lời mô tả, giải thích văn tắt cho một câu lệnh, một đoạn chương trình hoặc cả chương trình
- Chỉ có tác dụng giúp chương trình viết ra dễ đọc và dễ hiểu hơn
- Trình biên dịch sẽ tự động bỏ qua không dịch phần nội dung nằm trong phạm vi của vùng chú thích đó.
- 2 cách chú thích
 - Trên 1 dòng: //
 - Trên nhiều dòng: /* */

30

1.2.9. Chú thích (tiếp)

- Cách 1:
 - Vùng bắt đầu từ // đến cuối dòng là vùng chú thích.
 - Ví dụ:
`a = 5; b = 3; // Khai tạo giá trị cho các biến này`
- Cách 2:
 - Toàn bộ vùng bắt đầu nằm trong cặp kí hiệu /* */ là vùng chú thích.
 - Ví dụ:
`/* Đoạn chương trình sau khai báo 2 biến nguyên và khai tạo giá trị cho 2 biến nguyên này */
int a, b;
a = 5; b = 3;`

31

1.3. Cấu trúc cơ bản của một 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

32

1.3. Cấu trúc cơ bản... (tiếp)

- Phần 1: Khai báo tệp tiêu đề:
 - Thông báo cho chương trình dịch biết là chương trình có sử dụng những thư viện nào.
 - Ví dụ:

```
#include <stdio.h> // thao tác vào ra
#include <conio.h> // hàm của DOS
```
- Phần 2: Định nghĩa các kiểu dữ liệu mới
 - Định nghĩa các kiểu dữ liệu mới (nếu cần) dùng cho cả chương trình.

33

1.3. Cấu trúc cơ bản... (tiếp)

- 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;
```

34

1.3. Cấu trúc cơ bản... (tiếp)

- 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.

35

Ví dụ một chương trình C đơn giản

```
/* Chương trình sau sẽ nhập vào từ bàn phím
2 số nguyên và hiển thị ra màn hình tổng,
hieu tích của 2 số nguyên vừa nhập vào */
#include <stdio.h>
#include <conio.h>
void main()
{
    // Khai báo các biến trong chương trình
    int a, b;
    int tong, hieu, tich;
```

36

Ví dụ một chương trình C đơn giản (tiếp)

```
// Nhập vào từ bàn phím 2 số nguyên
printf("\nNhập vào số nguyên thứ nhất: ");
scanf("%d", &a);
printf("\nNhập vào số nguyên thứ hai: ");
scanf("%d", &b);
// Tính tổng, hiệu, tích của 2 số vừa nhập
tong = a + b;
hieu = a - b;
tich = a*b;
```

37

Ví dụ một chương trình C đơn giản (tiếp)

```
// Hiện thị các giá trị ra màn hình
printf("\n Tổng của 2 số vừa nhập là
%d", tong);
printf("\n Hiệu của 2 số vừa nhập là
%d", hieu);
printf("\n Tích của 2 số vừa nhập là
%d", tich);
// Đợi người dùng ấn phím bất kỳ để kết thúc
getch();
}
```

38

1.4. Biên dịch chương trình viết bằng ngôn ngữ C

- Biên dịch chương trình viết bằng ngôn ngữ C
 - Dùng trình biên dịch Turbo C++ 3.0
 - Cài đặt Turbo C++ 3.0
 - Viết chương trình
 - Sửa đường dẫn tới các thư viện (nếu cần)
 - Biên dịch
 - Chạy chương trình

39

1.5. Bài tập

- Bài tập 1:
Trong các định danh sau, định danh nào là không hợp lệ:
 - MAX_SINH_VIEN
 - CHIEU_CAO
 - ho va ten
 - 1_bien_nao_do
 - so_thuc_1

40



1.5. Bài tập (tiếp)

- Bài tập 2:

Hãy cho biết giá trị của các hằng nguyên sau trong chương trình:

0345, 0x168, 06356, 0xAF04

41



1.5. Bài tập (tiếp)

- Bài tập 3:

Cho biết biểu diễn dưới dạng số thực dấu phẩy tinh của các hằng số thực sau:

- 535.235 E+3
- 256.89 E-1
- 10.103 E-5

42



1.5. Bài tập (tiếp)

- Bài tập 4: Chạy thử hai chương trình sau xem có chương trình nào có lỗi không? Nếu có lỗi thì hãy xem trình biên dịch báo là lỗi gì?

- Chương trình 1:

```
void main() { }
```

- Chương trình 2:

```
#include <stdio.h>
#include <conio.h>
void fct() { }
```

43



Discussion

44