

Tin học đại cương

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội



Thông tin chung

- Giảng viên:
 - Nguyễn Hồng Phương
 - Bô môn Hê thống thông tin, Viên CNTT&TT, phòng 603, 702-B1.
 - Email: phuongnh-fit@mail.hut.edu.vn phuongnh@gmail.com
- Giờ tiếp sinh viên tại Bộ môn:
 - Sáng thứ hai hàng tuần
 - Ngoài ra, xin liên hê trước.



Tổng quan về môn học

- Mục tiêu: Sau khi học xong môn học này, sinh viên năm thứ nhất:
 - có kiến thức về tin học căn bản
 - có kiến thức về giải quyết bài toán
 - có thể lập trình bằng ngôn ngữ C
- Khối lương:
 - 4 tiết/ tuần, học trong 15 tuần trên giảng đường
 - Bài tập và Thực hành (theo quy định chung).

Đánh giá môn học

Dự lớp đầy đủ, tích cực xây dựng bài



- Kiểm tra giữa kỳ
- Thực hành
- Kiểm tra cuối kỳ (theo lịch chung)



Tài liệu học tập



- Các tác giả thuộc Viên CNTT&TT-ĐHBK Hà Nội, Giáo trình Tin học đại cương, NXB Bách Khoa -Hà Nội, 2011.
- Bài giảng trên lớp.
- TS. Quách Tuấn Ngọc, Giáo trình Tin học căn bản, NXB Thống kê, 2001.
- GS. Hoàng Kiếm, Tin học đại cương nâng cao, NXB Giáo dục, 1998.
- GS. Phạm Văn Ất, Kỹ thuật lập trình C: cơ sở và nâng cao, NXB Khoa học và kỹ thuật, 1999.



Nội dung môn học



- Phần I: TIN HỌC CĂN BẢN
 - Chương 1: Mở đầu
 - Thông tin và xử lý thông tin
 Máy tính và phân loại
 - Tin học
 - Chương 2: Biểu diễn dữ liệu trong máy tính Hê đểm
 - Mã hóa dữ liệu trong máy tính và đơn vị thông tin
 - Biểu diễn số nguyên
 - Thực hiện phép toán số học với số nguyên
 - Các phép toán logic với số nhị phân
 - Biểu diễn ký tự
 - Biểu diễn số thực



Nội dung môn học (2)



- Chương 3: Hệ thống máy tính
 - Tổ chức bên trong máy tính
 - Phần mềm máy tính
- Chương 4: Mang máy tính
 - Lịch sử, khái niệm
 - Phân loại
 - Các thành phần cơ bản
 - Mang Internet





Nội dung môn học (3)



- Chương 5: Hệ điều hành
 - Giới thiệu chung
 - Các hệ điều hành: Window, Linux,...
- Chương 6: Các hê thống ứng dung
 - Hệ thống thông tin quản lý
 - Hệ thông tin bảng tính
 - Hệ quản trị cơ sở dữ liệu
 - Các hệ thống thông minh



Nội dung môn học (4)



- Phần II: GIẢI QUYẾT BÀI TOÁN
- Chương 1: Giải quyết bài toán
 - Khái niêm
 - Các bước giải quyết bài toán bằng máy tính
 - Các phương pháp giải quyết bài toán bằng máy tính

 - Chương 2: Thuật toán
 - Định nghĩa
 - Biểu diễn
 - Một số thuật toán thông dụng

 - Thuật toán đệ quyThuật giải heuristic



Nội dung môn học (5)



- Phần III: LẬP TRÌNH BẰNG NGÔN NGỮ C
 - Chương 1: Tổng quan về ngôn ngữ C
 - Lịch sử
 - Các phần tử cơ bản
 - Cấu trúc cơ bản
 - Chương trình biên dịch
 - Chương 2: Kiểu dữ liệu, biểu thức và cấu trúc lập trình trong C
 - Các kiểu dữ liệu chuẩn
 - Biểu thức
 - Các cấu trúc lập trình

10



Nội dung môn học (6)



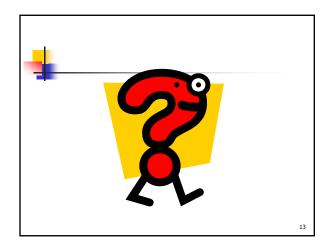
- Chương 3: Con trỏ, mảng và xâu trong C
 - · Con trỏ và địa chỉ
 - Mång
- Xâu ký tự
- Chương 4: Hàm
 - Khái niệm hàm
 - Khai báo và sử dụng hàm



Nội dung môn học (7)



- Chương 5: Cấu trúc
 - Khái niệm cấu trúc
 - Khai báo và sử dụng cấu trúc
 - Mảng cấu trúc
- Chương 6: Tệp dữ liệu (bỏ)
 - Khái niệm và phân loại tệp
 - Tệp văn bản
 - Tệp binary





Lời hay ý đẹp

"Trời có bốn mùa: Xuân, Hạ, Thu, Đông; Đất có bốn phương: Đông, Tây, Nam, Bắc; Người có bốn đức: Cần, Kiệm, Liêm, Chính; Thiếu một mùa không thể thành Trời; Thiếu một phương không thể thành Đất; Thiếu một đức không thể thành Người."

Hồ Chí Minh



PHẦN I: TIN HỌC CẮN BẢN Chương 1: Mở đầu

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viên Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội



Nội dung chương 1

- 1.1. Thông tin và xử lý thông tin
- 1.2. Máy tính và phân loại máy tính
- 1.3. Tin học



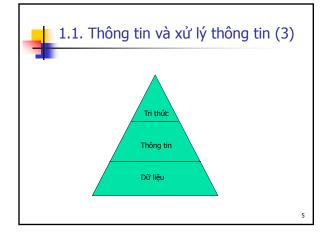
1.1. Thông tin và xử lý thông tin

- Thông tin, dữ liệu, tri thức
 - Thông tin (Information): mang lai cho con người sự hiểu biết, nhận thức tốt hơn về những đối tượng trong tự nhiên - xã hội
 - Dữ liệu (Data): biểu diễn của thông tin được thể hiện bằng các tín hiệu vật lý. Dữ liệu trong thực tế
 - các số liệu trong các biểu đồ
 - các ký hiệu quy ước như chữ viết
 - . các tín hiệu vật lý: ánh sáng, âm thanh, nhiệt độ, áp suất,...



1.1. Thông tin và xử lý thông tin (2)

- Tri thức (Knowledge):
 - thông tin ở mức trừu tượng hơn
 - khá đa dang
 - sự hiểu biết chung hay về một lĩnh vực cụ thể nào đó.
- Hệ thống thông tin (information system)
 - \bullet Dữ liệu \to Thông tin \to Tri thức Xử lý Xử lý





1.1. Thông tin và xử lý thông tin (4)

- Quy trình xử lý thông tin:
 - Mọi quá trình xử lý thông tin bằng máy tính hay bởi con người đều được thực hiện theo một qui trình sau:

LƯU TRỮ (STORAGE)



1.2. Máy tính và phân loại máy tính điện tử

- Lịch sử phát triển của máy tính điện tử
 - Công cụ tính toán ngày xưa: bàn tính bằng tay
 - Máy cộng cơ học của nhà toán học Pháp Blaise Pascal (1623-1662)
 - Máy tính cơ học cộng trừ nhân chia của nhà toán học Đức Leibnit (1646-1716)
 - Máy tính điện tử thực sự bắt đầu vào những năm 1950, đến nay đã trải qua 5 thế hệ dựa vào sự tiến bộ về công nghệ điện tử và vi điện tử.

-

Lịch sử phát triển máy tính(2)

- Thế hệ 1 (1950-1958): Von Neumann Machine
 - Sử dụng các bóng đèn điện tử chân không
 - Mạch riêng rẽ, vào số liệu bằng phiếu đục lỗ
 - Điều khiển bằng tay, kích thước rất lớn
 - Tiêu thụ năng lượng nhiều, tốc độ tính chậm khoảng 300 - 3.000 phép tính/s.

8





Lịch sử phát triển máy tính(3)

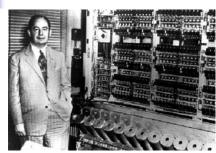


Bóng đèn chân không Máy tính đầu tiên:

ENIAC (Electronic Numerical Integrator And Computer)

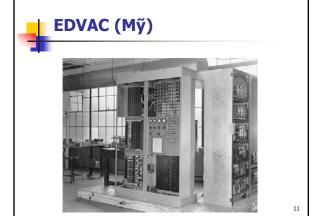


Lịch sử phát triển máy tính(4)



Von Neumann với máy tính Institute đầu tiên năm 1952

...

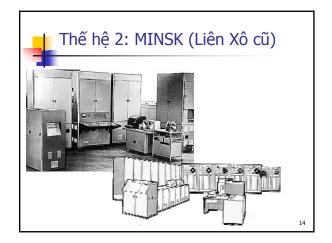




Lịch sử phát triển máy tính (tiếp)

- Thế hệ 2 (1958 1964): Transistors
 - Sử dụng bộ xử lý bằng đèn bán dẫn, mạch in
 - Đã có chương trình dịch như Cobol, Fortran và hệ điều hành đơn giản.
 - Kích thước máy còn lớn
 - Tốc độ tính khoảng 10.000 100.000 phép tính/giẩy
 - Điển hình:
 - IBM 7000 series (Mỹ)
 - MINSK (Liên Xô cũ)







- Thế hệ 3 (1965 1974): Integrated Circuits
 - Các bộ vi xử lý được gắn vi mạch điện tử cỡ nhỏ
 - Tốc độ tính khoảng 100.000 1 triệu phép tính/giây.
 - Có các hệ điều hành đa chương trình, nhiều người đồng thời hoặc theo kiểu phân chia thời gian.
 - Kết quả từ máy tính có thể in trực tiếp từ máy in.
 - Điển hình:
 - IBM-360 (Mỹ)
 - DEC PDP-8

Thế hệ 3: IBM 360 (Mỹ)

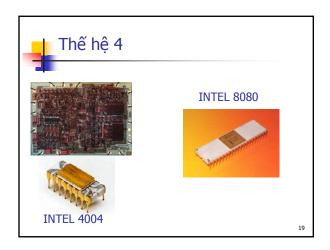
15

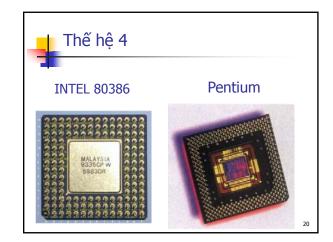




Lịch sử phát triển máy tính (tiếp)

- Thế hệ 4 (1974 1990): LSI (Large Scale Integration), Multiprocessors
- Tốc độ tính hàng chục triệu đến hàng tỷ phép tính/giầy.
- 2 loại máy tính chính:
 - Máy tính cá nhân để bàn (Personal Computer PC) hoặc xách tay (Laptop hoặc Notebook computer)
 - Các loại máy tính chuyên nghiệp thực hiện đa chương trình, đa xử lý,...
 - Hình thành các hệ thống mạng máy tính (Computer Networks).
 - Các ứng dụng phong phú đa phương tiện









- Thế hệ 5 (1990 nay): VLSI (Very Large Scale Integration), ULSI (Ultra), Artificial Intelligence (AI)
 - Công nghệ vi điện tử với tốc độ tính toán cao và khả năng xử lý song song.
 Mô phỏng các hoạt động của não bộ và hành vi con người

 - Có trí khôn nhân tạo với khả năng tự suy diễn phát triển các tình huống nhận được
 - Hệ quản lý kiến thức cơ bản để giải quyết các bài toán đa dang.

22



- Máy Vi tính (Microcomputer)
 - Được thiết kế cho một người dùng
 - Giá thành rẻ.
 - Được sử dụng phổ biến: máy để bàn (Desktop), máy trạm (Workstation), máy xách tay (Notebook),...
- Máy tính tầm trung (Mini Computer)
 - Tốc độ và hiệu năng tính toán mạnh hơn
 - Được thiết kế cho các ứng dụng phức tạp.
 - Giá ~ hàng vài chục nghìn USD
- Máy tính lớn (Mainframe Computer) và Siêu máy tính (Super Computer).



23

Phân loại máy tính (tiếp)

- Máy tính lớn và siêu máy tính (tiếp)
 - Phức tạp, có tốc độ siêu nhanh
 - Hiệu năng tính toán cao, cỡ hàng tỷ phép tính/giây
 - Nhiều người dùng đồng thời
 - Được sử dụng tại các Trung tâm tính toán/ Viện nghiên cứu để giải quyết các bài toán cực kỳ phức tạp, yêu cầu cao về tốc độ.
 - Giá thành rất đắt ~ hàng trăm ngàn, thậm chí hàng triệu USD



Phân loại máy tính hiện đại

- Máy tính để bàn (Desktop Computers)
- Máy chủ (Server)
- Máy tính nhúng (Embedded Computers)

25



Máy tính để bàn (Desktop)

- Là loại máy tính phổ biến nhất
- Các loại máy tính để bàn:
 - Máy tính cá nhân (Personal Computers PC)
 - Máy tính trạm làm việc (Workstations)
- 1981: IBM giới thiệu máy tính IBM-PC sử dụng bô xử lý Intel 8088
- 1984: Apple đưa ra Macintosh sử dụng bộ xử lý Motorola 68000
- Giá thành: 500 USD đến 10,000 USD

26



Máy chủ (Server)

- Thực chất là máy phục vụ
- Dùng trong mạng theo mô hình Client/Server
- Tốc độ và hiệu năng tính toán cao
- Dung lương bô nhớ lớn
- Đô tin cây cao
- Giá thành: hàng chục nghìn đến hàng chuc triêu USD

27



Máy tính nhúng (Embedded Computers)

- Được đặt trong thiết bị khác để điều khiển thiết bi đó làm việc
- Được thiết kế chuyên dụng
- Ví du:
 - Điện thoại di động
 - Máy ảnh số
- Bộ điều khiển trong máy giặt, điều hòa
- Router bộ định tuyến trên mạng
- Giá thành: vài USD đến hàng trăm nghìn USD

20



1.3. Tin học

- Tin hoc (Informatics)
 - Ngành khoa học nghiên cứu các phương pháp, công nghệ và kỹ thuật xử lý thông tin một cách tự động.
 - Công cụ: Máy tính điện tử và các thiết bị truyền tin.
 - Nội dung nghiên cứu:
 - Kỹ thuật phần cứng (Hardware engineering)
 - Thiết bị, linh kiện điện tử, công nghệ vật liệu mới... hỗ trợ cho máy tính và mạng máy tính, đẩy mạnh khả năng xử lý toán học và truyền thông thông tin.
 - Kỹ thuật phần mềm (Software engineering)
 - Các hệ điều hành, ngôn ngữ lập trình cho các bài toán khoa học kỹ thuật, mô phong, điều khiển tự động, tổ chức dữ liệu và quản lý hệ thống thông tin

20



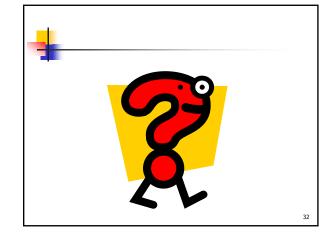
1.3. Tin học (tiếp)

- Công nghệ thông tin (Information Technology IT)
 - Ngành nghiên cứu các hệ thống thông tin dựa vào máy tính, đặc biệt là các phần mềm ứng dụng và phần cứng máy tính.
 - IT xử lý với các máy tính điện tử và các phần mềm máy tính nhằm chuyển đổi, lưu trữ, bảo vệ, truyền tin và trích rút thông tin một cách an toàn.
- Các ứng dụng ngày nay của IT
 - Các bài toán khoa học kỹ thuật
 - Các bài toán quản lý
 - Tự động hóa
 - Công tác văn phòng
 - Tin học và giáo dục
 - Thương mại điện tử
 Công nghệ thông tin với cuộc sống đời thường



1.3. Tin học (tiếp)

- Công nghệ thông tin và truyền thông: Information and Communication Technology (ICT).
 - Kết nối một số lượng máy tính với nhau
 - Internet Mạng máy tính toàn cầu



31



Lời hay ý đẹp

Trong 10 lần thành công thì có tới 9 lần thành công nhờ sự hăng hái và niềm tin trong công việc

Teewilson



Chương 2: Biểu diễn dữ liệu trong máy tính

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn
Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viên Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Nội dung chương này

- 2.1. Biểu diễn số trong các hệ đếm
 - 2.1.1. Hệ đếm cơ số b
 - 2.1.2. Hệ đếm thập phân
 - 2.1.3. Hệ đếm nhị phân
 - 2.1.4. Hệ đếm bát phân
 - 2.1.5. Hệ đếm thập lục phân
 - 2.1.6. Chuyển đổi một số từ hệ thập phân sang hệ đếm cơ số b
 - 2.1.7. Mệnh đề logic

2



Nội dung chương này (tiếp)

- 2.2. Biểu diễn dữ liệu trong máy tính và đơn vị thông tin
 - 2.2.1. Nguyên tắc chung
- 2.2.2. Đơn vị thông tin
- 2.3. Biểu diễn số nguyên
 - 2.3.1. Số nguyên không dấu
 - 2.3.2. Số nguyên có dấu
- 2.4. Phép toán số hoc với số nguyên
 - Công/trừ
 - Nhân/chia

+

Nội dung chương này (tiếp)

- 2.5. Tính toán logic với số nhị phân
- 2.6. Biểu diễn ký tự
 - 2.6.1. Nguyên tắc chung
 - 2.6.2. Bô mã ASCII
 - 2.6.3. Bộ mã Unicode
- 2.7. Biểu diễn số thực
 - 2.7.1. Nguyên tắc chung
 - 2.7.2. Chuẩn IEEE754/85



2.1. Biểu diễn số trong các hệ đếm

Hệ đếm là tập hợp các ký hiệu và quy tắc sử dụng tập ký hiệu đó để biểu diễn và xác định giá trị các số. Mỗi hệ đếm có một số chữ số/ký số (digits) hữu hạn. Tổng số chữ số của mỗi hệ đếm được gọi là **cơ số** (base hay radix), ký hiệu là b.



Các hệ đếm cơ bản

- Hệ thập phân (Decimal System) → con người sử dụng
- Hệ nhị phân (Binary System) → máy tính sử dụng
- Hệ mười sáu (Hexadecimal System) → dùng để viết gọn cho số nhị phân
- Hê bát phân (Octal System)



2.1.1. Hệ đếm cơ số b

- Hệ đếm cơ số b (b≥2 và nguyên dương) mang tính chất sau:
 - có b chữ số (ký số) để thể hiện giá trị số. Ký số nhỏ nhất là 0 và lớn nhất là b-1
 - giá trị (trọng số) vị trí thứ n trong một số của hệ đếm bằng cơ số b lũy thừa n: bⁿ
 - Số N_(b) trong hệ đếm cơ số b được biểu diễn bởi:

$$N_{(h)} = a_n a_{n-1} a_{n-2} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-m}$$



2.1.1. Hê đếm cơ số b

trong đó, số N_(b) có n+1 chữ số biểu diễn cho phần nguyên và m chữ số lẻ biểu diễn cho phần b_phân, và có giá trị là:

$$\begin{split} \mathbf{N}_{(b)} = \mathbf{a}_n.\mathbf{b}^n + \mathbf{a}_{n-1}.\mathbf{b}^{n-1} + \mathbf{a}_{n-1}.\mathbf{b}^{n-2} + ... + \mathbf{a}_1.\mathbf{b}^1 + \mathbf{a}_0.\mathbf{b}^0 + \mathbf{a}_{-1}.\mathbf{b}^{-1} + \mathbf{a}_{-1}.\mathbf{b}^{-2} + ... + \mathbf{a}_{-n}.\mathbf{b}^{-m} \\ \text{hay là:} \\ N_{(b)} = \sum_{i=1}^n a_i.\mathbf{b}^i \end{split}$$



2.1.2. Hệ đếm thập phân (Decimal System, b=10)

- Hệ đếm thập phân hay hệ đếm cơ số 10 là một trong các phát minh của người Ả rập cổ, bao gồm 10 chữ số theo ký hiệu sau: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Quy tắc tính giá trị của hệ đếm này là mỗi đơn vị ở một hàng bất kỳ có giá trị bằng 10 đơn vị của hàng kế cận bên phải. Ở đây b=10



2.1.2. Hệ đếm thập phân (Decimal System, b=10)

Bất kỳ số nguyên dương trong hệ thập phân có thể biểu diễn như là một tổng các số hạng, mỗi số hạng là tích của một số với 10 lũy thừa, trong đó số mũ lũy thừa được tăng thêm 1 đơn vị kể từ số mũ lũy thừa phía bên phải nó. Số mũ lũy thừa của hàng đơn vị trong hệ thập phân là 0

10



2.1.2. Hệ đếm thập phân (Decimal System, b=10)

 Ví dụ: Số 5246 có thể được biểu diễn như sau:

$$5246 = 5x10^3 + 2x10^2 + 4x10^1 + 6x10^0$$

= $5 \times 1000 + 2 \times 100 + 4 \times 10 + 6 \times 1$

 Thể hiện như trên gọi là ký hiệu mở rộng của số nguyên vì

$$5246 = 5000 + 200 + 40 + 6$$

11



2.1.2. Hệ đếm thập phân (Decimal System, b=10)

- Như vậy, trong số 5246: chữ số 6 trong số nguyên đại diện cho giá trị 6 đơn vị, chữ số 4 đại diện cho giá trị 4 chục (hàng chục), chữ số 2 đại diện cho giá trị 2 trăm (hàng trăm) và chữ số 5 đại diện cho giá trị 5 nghìn (hàng nghìn)
- Phần thập phân:
 - 254.68 = $2x10^2 + 5x10^1 + 4x10^0 + 6x10^{-1} + 8x10^{-2}$ = $200 + 50 + 4 + \frac{6}{10} + \frac{8}{100}$



2.1.3. Hệ đếm nhị phân (Binary System, b=2)

Với cơ số b=2, chúng ta có hệ đếm nhị phân. Đây là hệ đếm đơn giản nhất với 2 chữ số là 0 và 1. Mỗi chữ số nhị phân gọi là BIT (viết tắt từ chữ BInary digiT). Vì hệ nhị phân chỉ có 2 chữ số là 0 và 1, nên khi muốn diễn tả một số lớn hơn cần kết hợp nhiều bit với nhau. Ta có thể chuyển đổi số trong hệ nhị phân sang số trong hệ thập phân quen thuộc.

13



2.1.3. Hệ đếm nhị phân (Binary System, b=2)

 Ví dụ: Số 11101.11₍₂₎ sẽ tương đương với giá trị thập phân là :

						vị trí đấu chấm cách	
Số nhị phân :	1	1	1	0	1 .	1	1
Số vị trí :			2			-1	-
Trị vị trí :	2^4	2 ³	2^2	2^1	2°	2-1	2-2
Hệ 10 là :	16	8	4	2	1	0.5	0.25
như vậy:							
11101	.11(2)	= 1 x16	+ 1x8 +	1x4 +	0x2 + 1	x1 + 1x	$0.5 + 1 \times 0.25 = 29.75_{(10)}$
số 10101 (hệ	2) san	g hệ thá	ip phân	sẽ là:			
10101	(2) = 1	$x2^4 + 0$	$x2^3 + 1$	$x2^2 + 0$	$x2^{1} + 1x$	$2^0 = 16$	+ 0 + 4 + 0 + 1 = 21(10)



2.1.4. Hệ đếm bát phân

- Nếu dùng 1 tập hợp 3 bit thì có thể biểu diễn 8 trị khác nhau: 000, 001, 010, 011, 100, 101, 110, 111. Các trị này tương đương với 8 trị trong hệ thập phân là 0, 1, 2, 3, 4, 5, 7. Tập hợp các chữ số này gọi là hệ bát phân, là hệ đếm với b = 8 = 2³. Trong hệ bát phân, giá trị vi trí là lũy thừa của 8.
- Ví du:
- $235.64_{(8)}$ =**2**x8² + **3**x8¹ + **5**x8⁰ + **6**x8⁻¹ + **4**x8⁻² = 157. 8125₍₁₀₎

15



2.1.5. Hệ đếm thập lục phân (Hexadecimal System, b=16)

Hệ đếm thập lục phân là hệ cơ số b=16 = 2⁴, tương đương với tập hợp 4 chữ số nhị phân (4 bit). Khi thể hiện ở dạng hexa-decimal, ta có 16 chữ số gồm 10 chữ số từ 0 đến 9, và 6 chữ in A, B, C, D, E, F để biểu diễn các giá trị số tương ứng là 10, 11, 12, 13, 14, 15. Với hệ thập lục phân, giá trị vị trí là lũy thừa của 16

16



2.1.5. Hệ đếm thập lục phân (Hexadecimal System, b=16)

- Ví du:
- $34F5C_{(16)} = 3x16^4 + 4x16^3 + 15x16^2 + 5x16^1 + 12x16^0 = 216294_{(10)}$
- Ghi chứ: Một số ngôn ngữ lập trình quy định viết số hexa phải có chữ H ở cuối chữ số. Ví dụ: Số 15 viết là F_H.

17



2.1.6. Chuyển đổi một số từ hệ thập phân sang hệ cơ số b

- Đổi phần nguyên từ hệ thập phân sang hệ cơ số b.
 - Lấy số nguyên thập phân N₍₁₀₎ lần lượt chia cho b cho đến khi thương số bằng 0. Kết quả số chuyển đổi N_(b) là các số dư trong phép chia viết theo thứ tự ngược lại.
- Đổi phần thập phân từ hệ thập phân sang hệ cơ số b
 - Lấy phần thập phân N₍₁₀₎ lần lượt nhân với b cho đến khi phần thập phân của tích số bằng 0. Kết quả số chuyển đổi N_(b) là các số phần nguyên trong phép nhân viết ra theo thứ tự tính toán.

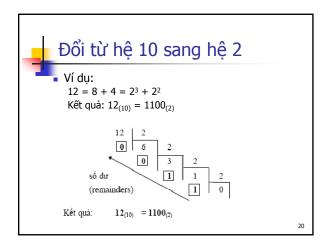


Lưu ý 1: Đổi từ hệ 10 sang hệ 2

- Chuyển đổi phần nguyên và phần lẻ riêng
- Chuyển đổi phần nguyên: 2 cách
 - Phân tích thành tổng các số lũy thừa của 2
 - Chia cho 2 được thương và số dư, sau đó lại lấy thương chia tiếp cho 2 cho đến khi thương = 0, viết các số dư theo thứ tự ngược lại

19

21





Đổi từ hệ 10 sang hệ 2

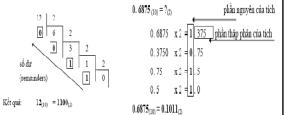
- Chuyển đổi phần lẻ
 - Lấy phần lẻ nhân 2 rồi lấy phần nguyên,...
 biểu diễn các phần nguyên theo chiều thuận
- Ví du:

```
0. 6875_{(10)} = ?_{(2)} phần nguyên của tích 0. 6875 x 2 = 1 375 phần thập phân của tích 0. 3750 x 2 = 0 . 75 0. 75 x 2 = 1 . 5 0. 5 x 2 = 1 . 0 0. 0.6875_{(10)} = 0.1011_{(2)}
```



Đổi từ hệ 10 sang hệ 2

 \bullet 12.6875₍₁₀₎ = 1100.1011₍₂₎



22

Đổi từ hệ 10 sang hệ 2

■ Bài tập: đổi số 35.375₍₁₀₎ sang hệ 2



Lưu ý 2: chuyển đổi nhị phân sang Hexa

- Duyệt từ phải sang trái, chia thành các nhóm 4 bit, sau đó thay từng nhóm 4 bit bằng một chữ số Hexa
- Ví dụ: $10 0011_2 = 23_{16}$

Hệ thập phân	Hệ nhị phân	Hệ mười sáu
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	В
12	1100	С
13	1101	D
14	1110	E
15	1111	F



Chuyển đổi thập phân sang Hexa

■ Thập phân \rightarrow Hexa: 14988 \rightarrow ?

14988 : 16 = 936 dư 12 tức là C 936 : 16 = 58 dư 8 58 : 16 = 3 dư 10 tức là A 3 : 16 = 0 dư 3 Như vậy, ta có: 14988₍₁₀₎ = 3A8C₍₁₆₎

25



2.1.7. Mênh đề logic

- Mệnh đề logic là mệnh đề chỉ nhận một trong
 2 giá trị : Đúng (TRUE) hoặc Sai (FALSE),
 tương đương với TRUE = 1 và FALSE = 0.
- Qui tắc: TRUE = NOT FALSE và FALSE = NOT TRUE
- Phép toán logic áp dụng cho 2 giá trị TRUE và FALSE ứng với tổ hợp AND (và) và OR (hoặc) như sau:

26



Mệnh đề logic

x	y	X AND y	X OR y
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE

2



2.2. Biểu diễn dữ liệu trong máy tính và đơn vi thông tin

- 2.2.1. Nguyên tắc chung
 - Thông tin và dữ liệu mà con người hiểu được tồn tại dưới nhiều dạng khác nhau, ví dụ như các số liệu, các ký tự văn bản, âm thanh, hình ảnh,... nhưng trong máy tính mọi thông tin và dữ liệu đều được biểu diễn bằng số nhị phân (chuỗi bit).
 - Để đưa dữ liệu vào cho máy tính, cần phải mã hoá nó về dạng nhị phân. Với các kiểu dữ liệu khác nhau cần có cách mã hoá khác nhau. Cu thể:

28



Nguyên tắc chung (tiếp)

- Các dữ liệu dạng số (số nguyên hay số thực) sẽ được chuyển đổi trực tiếp thành các chuổi số nhị phân theo các chuẩn xác định.
- Các ký tự được mã hoá theo một bộ mã cụ thể, có nghĩa là mỗi ký tự sẽ tương ứng với một chuỗi số nhị phân.
- phan.

 Các dữ liệu phi số khác như âm thanh, hình ảnh và nhiều đại lượng vật lý khác muốn đưa vào máy phải số hoá (digitalizing). Có thể hiểu một cách đơn giản khái niệm số hoá như sau: các dữ liệu tự nhiên thường là quá trình biến đổi liện tục, vì vậy để đưa vào máy tính, nó cần được biển đổi sang một dãy hữu hạn các giá trị số (nguyên hay thực) và được biểu diễn dưới dạng nhị phân.

4

Nguyên tắc chung (tiếp)

Với các tín hiệu như âm thanh, video, hay các tín hiệu vật lý khác, qui trình mã hoá được biểu diễn như sau:





Nguyên tắc chung (tiếp)

- Tuy rằng mọi dữ liệu trong máy tính đều ở dạng nhị phân, song do bản chất của dữ liệu, người ta thường phân dữ liệu thành 2 dạng:
 - Dạng cơ bản: gồm dạng số (nguyên hay thực) và dạng ký tự. Sổ nguyên khổng dấu được biểu diễn theo dạng nhị phân thông thường, số nguyên có dấu theo mã bù hai, còn số thực theo dạng dấu phảy động. Để biểu diễn một dữ liệu cơ bản, người ta sử dụng 1 số bit. Các bit này ghép lại với nhau để tạo thành từ: từ 8 bít, từ 16 bít,...
 - Dạng có cấu trúc: Trên cơ sở dữ liệu cơ bản, trong máy tính, người ta xây dựng nên các dữ liệu có cấu trúc phục vụ cho các mục đích sử dụng khác nhau. Tuỳ theo cách "ghép" chúng ta có màng, tập hợp,xâu, bản ghi,...

31



2.2.2. Đơn vi thông tin

- Đơn vị nhỏ nhất để biểu diễn thông tin gọi là bit.
 Một bit tương ứng với một sự kiện có 1 trong 2 trạng thái.
- Ví dụ: Một mạch đèn có 2 trạng thái là:
 - Tắt (Off) khi mạch điện qua công tắc là hở
 - Mở (On) khi mạch điện qua công tắc là đóng
- Số học nhị phân sử dụng hai ký số 0 và 1 để biểu diễn các số. Vì khả năng sử dụng hai số 0 và 1 là như nhau nên một chỉ thị chỉ gồm một chữ số nhị phân có thể xem như là đơn vị chứa thông tin nhỏ nhất

32



Đơn vị thông tin (tiếp)

Bit là chữ viết tắt của **BI**nary digi**T**. Trong tin học, người ta thường sử dụng các đơn vị đo thông tin lớn hơn như sau:

	Tên gọi	Ký hiệu	Giá trị
В	yte	В	8 bit
K	liloByte	KB	2^{10} B = 1024 Byte
N	legaByte	MB	2 ²⁰ B
G	igaByte	GB	2 ³⁰ B
Т	'eraByte	TB	2 ⁴⁰ B

3



2.3. Biểu diễn số nguyên

- Số nguyên gồm số nguyên không dấu và số nguyên có dấu.
- Về nguyên tắc đều dùng 1 chuỗi bit để biểu diễn.
- Đối với số nguyên có dấu, người ta sử dụng bit đầu tiên để biểu diễn dấu và bit này gọi là bit dấu.

34



2.3.1. Biểu diễn số nguyên không dấu

 Dạng tổng quát: giả sử dùng n bit để biểu diễn cho một số nguyên không dấu A:

$$a_{n-1}a_{n-2}...a_3a_2a_1a_0$$

• Giá trị của A được tính như sau:

$$A = a_{n-1} 2^{n-1} + a_{n-2} 2^{n-2} + \dots + a_1 2^1 + a_0 2^0$$

$$A = \sum_{i=1}^{n-1} a_i 2^i$$

Dải biểu diễn của A: từ 0 đến 2ⁿ - 1

25



Ví dụ:

Biểu diễn các số nguyên không dấu sau đây bằng 8 bit:

$$A = 45$$
 $B = 156$

Giải

A =
$$45 = 32 + 8 + 4 + 1 = 2^5 + 2^3 + 2^2 + 2^0$$

 \rightarrow A = 0010 1101
B = $156 = 128 + 16 + 8 + 4 = 2^7 + 2^4 + 2^3 + 2^2$
 \rightarrow B = 1001 1100



Cho các số nguyên không dấu X, Y được biểu diễn bằng 8 bit như sau:

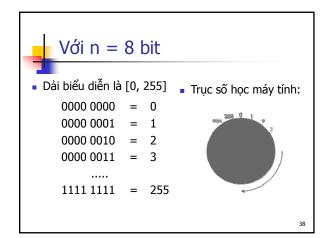
```
X = 0010 \ 1011
      Y = 1001 \ 0110
Xác định giá trị của X,Y
```

Giải:

Giái:

$$X = 0010 \ 1011 = 2^5 + 2^3 + 2^1 + 2^0$$

 $= 32 + 8 + 2 + 1 = 43$
 $Y = 1001 \ 0110 = 2^7 + 2^4 + 2^2 + 2^1$
 $= 128 + 16 + 4 + 2 = 150$





Biểu diễn số nguyên không dấu

- Với n = 16 bit:
 - dải biểu diễn: [0, 65535]
- Với n = 32 bit:
 - dải biểu diễn: [0, 2³²-1]



2.3.2. Biểu diễn số nguyên có dấu

- Khái niêm về số bù
 - Số bù 9 và số bù 10 (hê thập phân)
 - Giả sử có 1 số nguyên có dấu A được biểu diễn bởi n chữ số thập phân.
 - Số bù 9 của A: (10ⁿ 1) A
 - Số bù 10 của A: 10ⁿ A
 - Số bù 10 = số bù 9 + 1



Biểu diễn số nguyên có dấu

- Số bù 1 và số bù 2 (hệ nhị phân)
 - Giả sử có 1 số nguyên nhi phân A được biểu diễn = n bit nhi phân
 - Số bù 1 của A: (2ⁿ 1) A
 - Số bù 2 của A: 2ⁿ A
 - Số bù 2 = số bù 1 + 1

Số bù 1 và bù 2 (tiếp) Ví dụ: n = 4 bit, A = 0110 1111 10000 0110 0110 Số bù 1: 1001 1010 Số bù 2: Nhận xét: số bù 1 Nhận xét: A + số bù 2 của là đảo các bit nó, bỏ bit ngoài cùng đi, $0 \Leftrightarrow 1$ ta được 0000



Biểu diễn số nguyên có dấu bằng số bù 2

- Dùng n bit để biểu diễn số nguyên có dấu: $a_{n-1}a_{n-2}...a_2a_1a_0$
- Với số dương:
 - bit $a_{n-1} = 0$
 - các bit còn lại biểu diễn độ lớn của số dương đó
 - Dạng tổng quát của số dương: 0a_{n-2}...a₂a₁a₀
 - Giá trị của số dương:

$$A = \sum_{i=0}^{n-2} a_i \times 2^i$$

■ Dải biểu diễn: [0,2ⁿ⁻¹-1]



Biểu diễn số nguyên có dấu bằng số bù 2

- Với số âm: được biểu diễn bằng số bù 2 của số dương tương ứng
 - bit a_{n-1} = 1
 - Dạng tổng quát của số âm: $1a_{n-2}...a_2a_1a_0$
 - Giá trị của số âm:

$$A = -2^{n-1} + \sum_{i=0}^{n-2} a_i \times 2^i$$

■ Dải biểu diễn: [-2^{n-1,} -1]

44



Biểu diễn số nguyên có dấu bằng số bù 2

- Kết hợp lại, ta có dải biểu diễn của số nguyên có dấu n bit là:
 - [-2ⁿ⁻¹, 2ⁿ⁻¹ 1]
- Công thức tổng quát:

$$A = -a_{n-1} \times 2^{n-1} + \sum_{i=0}^{n-2} a_i \times 2^i$$

45



Một số ví dụ về số nguyên có dấu

 Xác định giá trị của các số nguyên có dấu 8 bit sau đây:

$$A = 0101 \ 0110$$

$$B = 11010010$$

Giải

$$A = 2^6 + 2^4 + 2^2 + 2^1 = 64 + 16 + 4 + 2 = +86$$

$$B = -2^7 + 2^6 + 2^4 + 2^1 = -128 + 64 + 16 + 2 = -46$$

46

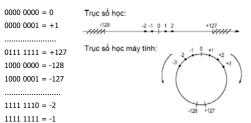


Bài tập

- Biểu diễn các số nguyên sau với n = 8 bit:
 - X=+58
 - Y=-80
- Xác định giá trị của số nguyên có dấu 8 bit: Z = 1100 1001

Trường hợp cụ thể

 Trường hợp 8 bit: biểu diễn các giá trị từ -128 đến +127





Trường hợp cụ thể

- ▼Với n = 16 bit, dải biểu diễn:
 - **.** [-32768, + 32767]
- Với n = 32 bit: -2³¹ đến 2³¹ − 1
- Với n = 64 bit: -2⁶³ đến 2⁶³ 1
- Chuyển đổi từ byte thành word:
 - đối với số dương thêm 8 bit 0 bên trái
 - +19 = 0001 0011 (8 bit)
 - +19 = 0000 0000 0001 0011 (16 bit)
 - đối với số âm thêm 8 bit 1 bên trái
 - -19 = 1110 1101 (8 bit)
 - -19 = 1111 1111 1110 1101 (16 bit)



Binary Code Decimal Code

 Dùng 4 bit để mã hóa từng chữ số thập phân từ 0 đến 9

> $0\rightarrow0000$ $1\rightarrow0001$ $8\rightarrow1000$ $9\rightarrow1001$

 Có 6 tổ hợp không dùng: 1010, 1011, 1100, 1101, 1110, 1111

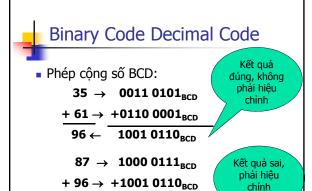
50



Binary Code Decimal Code

- 35→0011 0101_{BCD}
- $61 \rightarrow 0110\ 0001_{BCD}$
- 1087→0001 0000 1000 0111_{BCD}
- Cứ 1 chữ số thập phân đơn lẻ được mã hóa bằng 4 bit

51





Binary Code Decimal Code

- Hiệu chỉnh:
 - Nhận xét: 7 + 6 hay 8 + 9 đều vượt 9 nên có nhớ.
 - Hiệu chỉnh bằng cách cộng thêm 6 ở những vi trí có nhớ (>9)
 - 1 0001 1101
 - + 0110 0110 ← hiệu chỉnh

0001 1000 0011_{BCD} → kết quả đúng

4

Các kiểu lưu trữ số BCD

183 1 0001 1101_{BCD}

 BCD không gói (Unpacked BCD): mỗi số BCD 4 bit được lưu trữ trong 4 bit thấp của mỗi byte. Ví dụ: Số 35 được lưu trữ:

0011

0101

 BCD gói (packed BCD): hai số BCD được lưu trữ trong một byte. Ví dụ: Số 35 được lưu trữ:

0011 0101

2.4. Các phép toán số học với số nguyên Phép cộng số nguyên không dấu Bộ cộng n-bit Bộ cộng n-bit



2.4. Các phép toán số hoc với số nguyên

- Phép cộng số nguyên không dấu
 - Tiến hành cộng lần lượt từng bít từ phải qua
 - Khi cộng hai số nguyên không dấu n bits ta thu được một số nguyên không dấu cũng n
 - Nếu tổng của hai số đó lớn hơn 2ⁿ⁻¹ thì khi đó sẽ tràn số (C_{out} = 1) và kết quả sẽ là sai.
 - Để tránh hiện tượng này, ta dùng nhiều bit



Ví dụ phép cộng số nguyên không dấu

Với trường hợp 8 bit, nếu tổng nhỏ hơn 255 thì kết quả đúng



Phép đảo dấu

Phép đảo dấu thực chất là lấy bù 2

+37 = 0010 0101-37 = 1101 1011 bù 1: 0010 0100 bù 1: 1101 1010 bù 2: 1101 1011 = -37 bù 2: 0010 0101 = +37



Cộng hai số nguyên có dấu

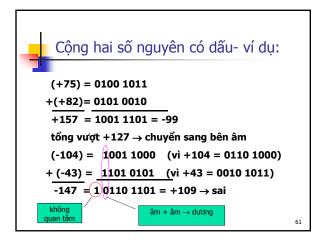
Khi công 2 số nguyên có dấu n bit, không quan tâm đến bit Cout, và kết quả nhận được là n bit:

- Cộng 2 số khác dấu kết quả luôn đúng
- Cộng 2 số cùng dấu:
 - nếu dấu kết quả cùng dấu với các số hạng thì kết quả
 - nếu kết quả có dấu ngược lại, khi đó có tràn xảy ra (Overflow) và kết quả bị sai
- Tràn xảy ra khi tổng nằm ngoài dải biểu diễn

$$[-(2^{n-1}),+(2^{n-1}-1)]$$

Cộng hai số nguyên có dấu- ví dụ:

(+70) = 0100 0110+(+42)= 0010 1010 +112 = 0111 0000 = +112 (+97) = 01100001+(-52) = 1100 1100(v) +52 = 0011 0100)+45 = 100101101 = +45

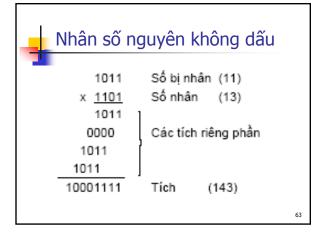




Nguyên tắc thực hiện phép trừ

- Phép trừ hai số nguyên: X-Y = X + (-Y)
- Nguyên tắc: lấy bù 2 của số trừ Y để được –Y, sau đó cộng với số bị trừ X

62





Nhân số nguyên không dấu

- Các tích riêng phần được xác định như sau:
 - nếu bít của số nhân = 0 thì tích riêng phần = 0
 - nếu bít của số nhân = 1 thì tích riêng phần = số bị nhân
 - tích riêng phần tiếp theo được dịch trái so với tích riêng phần trước đó
- Tích = tổng các tích riêng phần
- Nhân 2 số nguyên n bit, tích có độ dài là 2n bit (không bao giờ tràn)

64



Nhân hai số nguyên có dấu

- Sử dụng thuật giải nhân hai số nguyên không dấu
- Bước 1: chuyển đổi số bị nhân và số nhân thành số dương tương ứng
- Bước 2: nhận 2 số dương bằng thuật giải đã học, được tích của 2 số dương
- Bước 3: hiệu chỉnh dấu của tích như sau:
 - nếu 2 thừa số ban đầu cùng dấu thì không cần hiệu chỉnh
 - nếu 2 thừa số ban đầu là khác dấu thì ta lấy bù 2 của tích ở kết quả bước 2

4

Chia số nguyên không dấu

Số bị chia 10010011 1011 Số chia 1011 00001101 Thường 001110 1011 001111 1011 100 Phần dư



Chia số nguyên có dấu

- Bước 1: Chuyển đổi số bị chia và số chia về thành số dương tương ứng.
- Bước 2: Sử dụng thuật giải chia số nguyên không dấu để chia hai số dương, kết quả nhận được là thương Q và phần dư R đều là dương
- Bước 3: Hiệu chỉnh dấu của kết quả như sau:
 (Lưu ý: phép đảo dấu thực chất là phép lấy bù hai)

Số bị chia	Số chia	Thương	Số dư
dương	dương	giữ nguyên	giữ nguyên
dương	âm	đảo dấu	giữ nguyên
âm	dương	đảo dấu	đảo dấu
âm	âm	giữ nguyên	đảo dấu



2.5. Tính toán logic với số nhị phân

		AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

60



2.5. Tính toán logic với số nhị phân

	NOT
0	1
1	0

6



2.5. Tính toán logic với số nhị phân

- Thực hiện các phép toán logic với 2 số nhi phân:
 - Kết quả là 1 số nhị phân khi thực hiện các phép toán logic với từng cặp bit của 2 số nhị phân đó
 - Các phép toán này chỉ tác động lên từng cặp bit mà không ảnh hưởng đến bit khác.

70



2.5. Tính toán logic với số nhị phân

VD: A = 1010 1010 và B = 0000 1111

15. N = 1010 1010 Va B = 0000 1111						
	AND	OR	XOR	NOT		
1010 1010				01010101		
0000 1111				11110000		
	00001010	10101111	10100101			

Nhận xét: +Phép AND dùng để xoá một số bit và giữ nguyên 1 số bit còn lại. +Phép OR dùng để thiết lập 1 số bit và giữ nguyên 1 số bit khác.



2.6. Biểu diễn ký tự

Nguyên tắc chung:

- Các ký tự cũng cần được chuyển đổi thành chuỗi bit nhị phân gọi là mã ký tự.
- Số bit dùng cho mỗi ký tự theo các mã khác nhau là khác nhau.

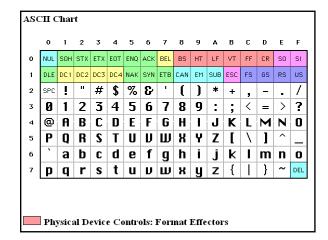
Vd : Bộ mã ASCII dùng 8 bit cho 1 ký tự. Bộ mã Unicode dùng 16 bit.



Bộ mã ASCII (American Standard Code for Information Interchange)

- Do ANSI (American National Standard Institute) thiết kế
- http://www.asciitable.com/
- ASCII là bộ mã được dùng để trao đổi thông tin chuẩn của Mỹ. Lúc đầu chỉ dùng 7 bit (128 ký tự) sau đó mở rộng cho 8 bit và có thể biểu diễn 256 ký tự khác nhau trong máy tính
- Bộ mã 8 bit \rightarrow mã hóa được cho 2^8 = 256 kí tự, có mã từ $00_{16} \div \text{FF}_{16}$, bao gồm:
 - 128 kí tự chuẩn có mã từ 00₁₆ ÷ 7F₁₆
 - 128 kí tự mở rộng có mã từ 80₁₆ ÷ FF₁₆

73





Bộ mã ASCII (tiếp)

- 95 kí tự hiển thị được:có mã từ 20₁₆ ÷
 7E₁₆
 - 26 chữ cái hoa Latin 'A' \div 'Z' có mã từ $41_{16} \div 5A_{16}$
 - = 26 chữ cái thường Latin 'a' \div 'z' có mã từ $61_{16} \div 7 {\rm A}_{16}$
 - = 10 chữ số thập phân '0' \div '9' có mã từ $30_{16} \div 39_{16}$

75



Bộ mã ASCII (tiếp)

- 95 ký tự hiển thị được:
 - Các dấu câu: . , ?!:; ...
 - Các dấu phép toán: + * / ...
 - Một số kí tự thông dụng: #, \$, &, @, ...
 - Dấu cách (mã là 20₁₆)
- 33 mã điều khiển: mã từ 00₁₆ ÷ 1F₁₆ và 7F₁₆ dùng để mã hóa cho các chức năng điều khiển

76



Điều khiển định dang

- BS Backspace Lùi lại một vị trí. Ký tự điều khiển con trỏ lùi lại một vị trí.

 HT Horizontal Tab Ký tự điều khiển con trỏ dịch đi một khoảng định trước

 Line Feed Ký tự điều khiển con trỏ xuống dòng

 VT Vertical Tab Ký tự điều khiển con trỏ dịch đi một số dòng
- FF Form Feed Ký tự điều khiển con trỏ chuyển xuống đâu trang tiếp theo.
- CR Carriage Return Ký tự điều khiển con trỏ về đâu dòng hiện hành.

77



Các ký tự mở rộng của bảng mã ASCII

- Được định nghĩa bởi:
 - Nhà chế tạo máy tính
 - Người phát triển phần mềm
- Ví du:
 - Bộ mã ký tự mở rộng của IBM: được dùng trên máy tính IBM-PC.
 - Bộ mã ký tự mở rộng của Apple: được dùng trên máy tính Macintosh.
 - Các nhà phát triển phần mềm tiếng Việt cũng đã thay đổi phần này để mã hoá cho các ký tự riêng của chữ Việt, ví dụ như bộ mã TCVN 5712.



Bộ mã Unicode

- Do các hãng máy tính hàng đầu thiết kế
- Là bộ mã 16-bit, Vậy số ký tự có thể biểu diễn (mã hoá) là 2¹⁶
- Được thiết kế cho đa ngôn ngữ, trong đó có tiếng Việt

79



2.7. Biểu diễn số thực

- 2.7.1. Nguyên tắc chung
 - Để biểu diễn số thực, trong máy tính người ta thường dùng ký pháp dấu phẩy động (Floating Point Number).
 - Tổng quát: một số thực X được biểu diễn theo kiểu số dấu phẩy động như sau:
 - X = M * R^E
 - M là phần định trị (Mantissa)
 - R là cơ số (Radix)
 - E là phần mũ (Exponent)

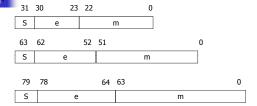
80



2.7.2. Chuẩn IEEE754/85

- Cơ số R = 2
- Các dang:
 - 32 bit (4 byte float trong C)
 - 48 bit (real trong Pascal)
 - 64 bit (8 byte)
 - 80 bit (10 byte)

Các dạng biểu diễn chính



- trường S nằm bên trái nhất biểu diễn dấu
- e: mũ
- m: định tri

82



Dang 32 - bit

- S là bit dấu
 - S = 0: số dương
 - S = 1: số âm
- e (8 bit) là mã excess 127 của phần mũ E:
 - E = e 127
 - khi e = 0 thì phần mũ = -127, khi e = 127 thì phần mũ = 0
 - e_{max} = 255 (8 bit)
 - giá trị 127 gọi là độ lệch (bias)
- m (23 bit) là phần lẻ của phần định trị M: M=1.m
- Công thức xác định giá trị của số thực:

$$X = (-1)^{S} * 1.m * 2^{e-127}$$

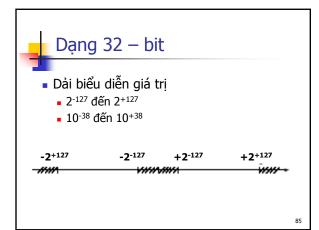
83



Dang 32 - bit

- Các quy ước đặc biệt
 - Các bit của e=0, các bit của m=0 thì $X=\pm0$ x000 0000 0000 0000 0000 0000 0000 $\rightarrow X=\pm0$

 - Các bit của e = 1, còn m có ít nhất 1 bit = 1 thì nó không biểu diễn cho số nào cả (NaN – Not A Number)





Dang 32 – bit. Ví du:

- Xác định giá trị của số thực được biểu diễn bằng 32 bit như sau:
- $S = 1 \rightarrow s\tilde{o} \hat{a} m$
- $e = 1000\ 0010_2 = 130 \rightarrow E = 130 127 = 3$
- V_{ay}^2 , X= -1.10101100*2³ = -1101.011 = -13.375

86



Dạng 32 – bit. Ví dụ (tiếp):

- Kết quả = +1.0

87



Dang 64 - bit

- S là bit dấu
- e (11 bit): mã excess-1023 của phần mũ $E \rightarrow E = e 1023$
- m (52 bit): phần lẻ của phần định trị M
- Giá trị số thực:

$$X = (-1)^{S} * 1.m * 2^{e-1023}$$

■ Dải giá tri biểu diễn: 10⁻³⁰⁸ đến 10⁺³⁰⁸

00



Dang 80 – bit

- S là bit dấu
- e (15 bit): mã excess-16383 của phần mũ E \rightarrow E = e 16383
- m (64 bit): phần lẻ của phần định trị M
- Giá trị số thực:

$$X = (-1)^{S} * 1.m * 2^{e-16383}$$

■ Dải giá trị biểu diễn: 10⁻⁴⁹³² đến 10⁺⁴⁹³²



Thực hiện phép toán số dấu phẩy động

- X1 = M1 * R^{E1}
- $X2 = M2 * R^{E2}$
- Ta có:
 - X1 * X2 = (M1 * M2) * R^{E1+ E2}
 - X1 / X2 = (M1 / M2) * R^{E1 E2}
 - X1 ± X2 = (M1* R^{E1-E2} ± M2) * R^{E2} , với E2 ≥ E1



Các khả năng tràn số

- Tràn trên số mũ (Exponent Overflow): mũ dương vượt ra khỏi giá trị cực đại của số mũ dương có thể (→∞)
- Tràn dưới số mũ (Exponent Underflow): mũ âm vượt ra khỏi giá trị cực đại của số mũ âm có thể (→0)
- Tràn trên phần định trị (Mantissa Overflow): công hai phần định trị có cùng dấu, kết quả bị nhớ ra ngoài bit cao nhất.
- Tràn dưới phần định trị (Mantissa Underflow): Khi hiệu chỉnh phần định trị, các số bị mất ở bên phải phần định trị.

91



Phép cộng và phép trừ

- Kiểm tra các số hạng có bằng 0 hay không.
- Hiệu chỉnh phần định trị.
- Cộng hoặc trừ phần định trị.
- Chuẩn hóa kết quả.

92





Lời hay ý đẹp

"Hạnh phúc lớn nhất của một nhà bác học là thấy học trò của mình vượt thầy" Lopée

"Nước trong quá thì không có cá, người xét nét quá thì không có bạn bè" Cổ ngôn Trung Hoa



Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn
Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Nội dung chương 3

- 3.1. Giới thiệu
- 3.2. Chức năng và các thành phần của máy tính
- 3.3. Liên kết hệ thống
- 3.4. Hoạt động của máy tính
- 3.5. Phần mềm máy tính

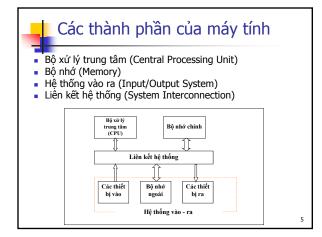
2





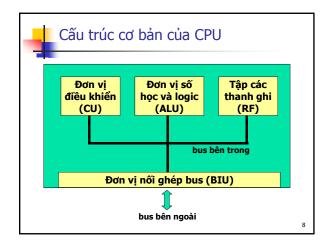
3.2. Chức năng và các thành phần của máy tính

- Chức năng cơ bản của hệ thống máy tính:
 - Xử lý dữ liệu
 - Lưu trữ dữ liêu
 - Trao đổi dữ liêu
 - Điều khiển











Các thành phần cơ bản của CPU

- Đơn vị điều khiển (Control Unit CU): điều khiển hoạt động của máy tính theo chương trình đã đinh sẵn.
- Đơn vị số học và logic (Arithmetic and Logic Unit -ALU): thực hiện các phép toán số học và các phép toán logic trên các dữ liệu cụ thể.
- Tập thanh ghi (Register File RF): lưu giữ các thông tin tạm thời phục vụ cho hoạt động của CPU.
- Đơn vị nối ghép bus (Bus Interface Unit BIU): kết nối và trao đổi thông tin giữa bus bên trong (internal bus) và bus bên ngoài (external bus).



Tốc độ của bộ xử lý

- Tốc độc của bộ xử lý:
 - Số lệnh được thực hiện trong 1 giây
 - MIPS (Million of Instructions per Second)
 - Khó đánh giá chính xác
- Tần số xung nhịp của bộ xử lý:
 - Bộ xử lý hoạt động theo một xung nhịp (clock) có tần số xác đinh.
 - Tốc độ của bộ xử lý được đánh giá gián tiếp thông qua tần số của xung nhịp.

10



Tốc đô của bô xử lý

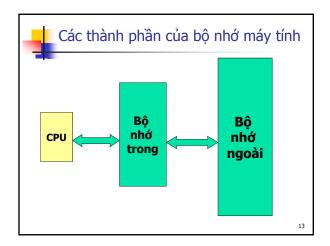
Dang xung nhip:

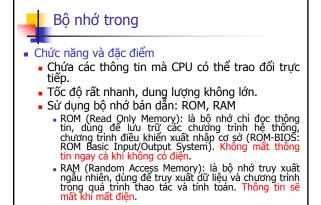


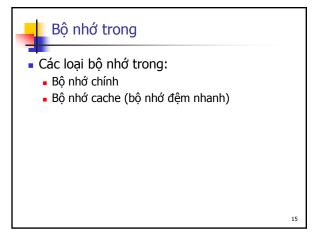
- T₀ là chu kỳ xung nhịp
- Mỗi thao tác của bộ xử lý cần kT₀, k∈N
- Tần số xung nhịp $f_0 = 1/T_0$
- Ví dụ: máy tính dùng bộ xử lý Pentium IV 2GHz
 - $f_0 = 2GHz = 2 \times 10^9 Hz$
 - $T_0 = 1/f_0 = 1/(2 \times 10^9) = 0.5 \text{ (ns)}$

Bô nhớ máy tính

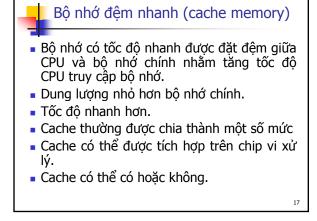
- Chức năng: lưu trữ chương trình và dữ liệu.
- Các thao tác cơ bản với bô nhớ:
 - Thao tác đoc (Read)
 - Thao tác ghi (Write)
- Các thành phần chính
 - Bộ nhớ trong (Internal Memory)
 - Bộ nhớ ngoài (External Memory)

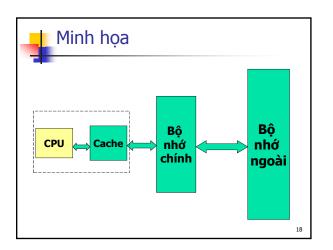














Bộ nhớ ngoài (External Memory)

- Chức năng và đặc điểm
 - Lưu giữ tài nguyên phần mềm của máy tính.
 - Được kết nối với hệ thống dưới dạng các thiết bị vào ra.
 - Dung lương lớn.
 - Tốc độ chậm
- Các loai bô nhớ ngoài:
 - Bộ nhớ từ: đĩa cứng, đĩa mềm
 - Bộ nhớ quang: đĩa CD, DVD
 - Bộ nhớ bán dẫn: Flash disk, memory card

19

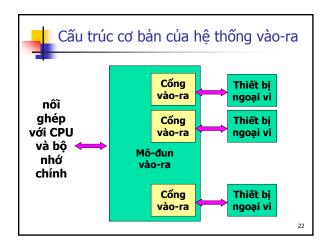




Hệ thống vào ra

- Chức năng: trao đổi thông tin giữa máy tính với thế giới bên ngoài.
- Các thao tác cơ bản:
 - Vào dữ liệu (Input)
 - Ra dữ liệu (Output)
- Các thành phần chính:
 - Các thiết bị ngoại vi (Peripheral Devices)
 - Các mô-đun vào-ra (IO Modules)

21





Các thiết bị ngoại vi

- Chức năng: chuyển đổi dữ liệu giữa bên trong và bên ngoài máy tính.
- Các loại thiết bị ngoại vi cơ bản:
 - Thiết bị vào: bàn phím, chuột, máy quét,...
 - Thiết bị ra: màn hình, máy in, máy chiếu,...
 - Thiết bị nhớ: các ổ đĩa,...
 - Thiết bi truyền thông: MODEM,....









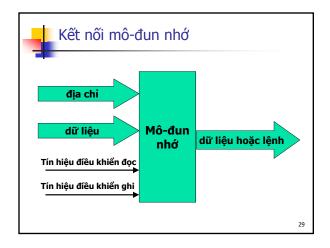
- Chức năng: nối ghép các thiết bị ngoại vi với máy tính.
- Mỗi mô-đun vào-ra có một hoặc một vài cổng vào-ra (I/O Port).
- Mỗi cổng vào-ra được đánh một địa chỉ xác định.
- Các thiết bị ngoại vi được kết nối và trao đổi dữ liệu với máy tính thông qua các cổng vào-ra.

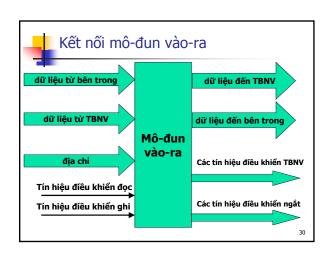
27

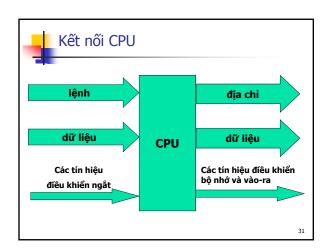


■ Mô-đun vào-ra

■ ☞cần được kết nối với nhau





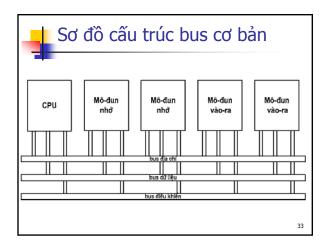




3.3.2. Cấu trúc bus cơ bản

- Bus: tập hợp các đường kết nối dùng để vận chuyển thông tin giữa các mô-đun của máy tính với nhau.
- Các bus chức năng:
 - Bus đia chỉ
 - Bus dữ liêu
 - Bus điều khiển
- Độ rộng bus: là số đường dây của bus có thể truyền các bit thông tin đồng thời (chỉ dùng cho bus địa chỉ và bus dữ liệu).

32





Bus địa chỉ

- Chức năng: vận chuyển địa chỉ để xác định ngăn nhớ hay cổng vào-ra.
- Độ rộng bus địa chỉ: xác định dung lượng bộ nhớ cực đại của hệ thống.
- Nếu độ rộng của bus địa chỉ là N bit:

 $A_{N-1}, A_{N-2}, ..., A_2, A_1, A_0$

- →có thể đánh địa chỉ tối đa cho 2^N ngăn nhớ
- Ví dụ: bộ xử lý Pentium có bus địa chỉ 32 bit →không gian địa chỉ là 2³²byte=4GB (đánh địa chỉ theo byte)

24



Bus dữ liêu

- Chức năng:
 - vận chuyển lệnh từ bộ nhớ đến CPU
 - vận chuyển dữ liệu giữa CPU, các mô-đun nhớ và các mô-đun vào-ra với nhau
- Độ rộng bus dữ liệu: xác định số bit dữ liệu có thể được trao đổi đồng thời.
 - M bit: D_{M-1}, D_{M-2}, ..., D₂, D₁, D₀
 - M thường là 8, 16, 32, 64, 128 bit.
- Ví dụ: các bộ xử lý Pentium có bus dữ liệu 64 bit.



Bus điều khiển

- Chức năng: vận chuyển các tín hiệu điều khiển
- Các loai tín hiệu điều khiển:
 - Các tín hiệu phát ra từ CPU để điều khiển môđun nhớ và mô-đun vào-ra.
 - Các tín hiệu từ mô-đun nhớ hay mô-đun vào-ra gửi đến yêu cầu CPU.



Đặc điểm của cấu trúc đơn bus

- Bus hệ thống chỉ phục vụ được một yêu cầu trao đổi dữ liệu tại một thời điểm.
- Bus hệ thống phải có tốc độ bằng tốc độ bus của mô-đun nhanh nhất trong hệ thống
- Bus hệ thống phụ thuộc vào cấu trúc bus (các tín hiệu) của bộ xử lý → các mô-đun nhớ và mô-đun vào-ra cũng phụ thuộc vào bộ xử lý.
- Vì vậy cần phải phân cấp bus → đa bus

37



Phân cấp bus trong máy tính

- Phân cấp bus cho các thành phần
 - Bus của bô xử lý
 - Bus của bộ nhớ chính
 - Các bus vào-ra
- Phân cấp bus khác nhau về tốc độ
- Bus bộ nhớ chính và các bus vào ra không phụ thuộc vào bộ xử lý cụ thể.

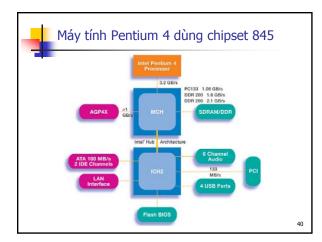
38



Các bus điển hình trong PC

- Bus của bộ xử lý (Front Side Bus-FSB): có tốc độ nhanh nhất
- Bus của bộ nhớ chính: nối ghép với các mô-đun RAM
- AGP bus (Accelerated Graphic Port): női ghép card màn hình tăng tốc
- PCI bus (Peripheral Component Interconnect); nổi ghép các thiết bị ngoại vi có tốc độ trao đổi dữ liệu nhanh.
- USB (Universal Serial Bus): bus női tiếp đa năng.
- IDE (Integrated Device Electronics): bus kết nối với ổ đĩa cứng hoặc ổ đĩa CD, DVD

39





Các kiểu bus

- Bus dành riêng (Dedicated):
 - Các đường địa chỉ và dữ liệu tách rời
 - Ưu điểm: điều khiển đơn giản
 - Nhược điểm: có nhiều đường kết nối
- Bus don kênh (Multiplexed):
 - Các đường dùng chung cho địa chỉ và dữ liệu
 - Có đường điều khiển để phân biệt có địa chỉ hay có dữ liệu
 - Ưu điểm: có ít đường dây
 - Nhược điểm: điều khiển phức tạp hơn, hiệu năng hạn chế



Phân xử bus

- Có nhiều mô-đun điều khiển bus như CPU và bộ điều khiển vào-ra
- Chỉ cho phép một mô-đun điều khiển bus tai môt thời điểm
- Phân xử bus có thể:
 - tập trung: có 1 bộ điều khiển bus (Bus Controller / Arbiter) hoặc là 1 phần của CPU hay mạch tách rời.
 - phân tán: mỗi mô-đun có thể chiếm bus và có đường điều khiển đến tất cả các mô-đun khác.



3.4. Hoạt động của máy tính

- 3.4.1. Thực hiện chương trình
- 3.4.2. Ngắt
- 3.4.3. Hoạt động vào-ra

43



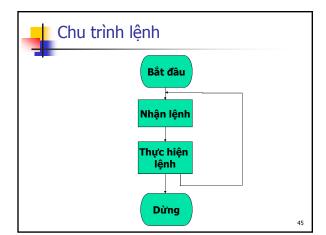
3.4.1. Thực hiện chương trình

- Là hoạt động cơ bản của máy tính
- Máy tính lăp đi lăp lai hai bước:
 - Nhân lênh
 - Thực hiện lệnh

chu trình lệnh

 Thực hiện chương trình bị dừng nếu thực hiện lệnh bị lỗi hoặc gặp lệnh dừng

44





Nhận lệnh

- Bắt đầu mỗi chu trình lệnh, CPU nhận lệnh từ bô nhớ chính.
- Bộ đếm chương trình PC (Program Counter) của CPU giữ địa chỉ của lệnh sẽ được nhận.
- CPU nhận lệnh từ ngăn nhớ được trỏ bởi
 PC.
- Lệnh được nạp vào thanh ghi lệnh IR (Instruction Register).
- Sau khi lệnh được nhận vào, nội dung PC tự động tăng để trỏ sang lệnh kế tiếp.

46



Thực hiện lệnh

- Bộ xử lý giải mã lệnh đã được nhận và phát tín hiệu điều khiển thực hiện thao tác mà lệnh yêu cầu.
- Các kiểu thao tác của lênh:
 - Trao đổi dữ liệu giữa CPU và bộ nhớ chính
 - Trao đổi dữ liệu giữa CPU và mô-đun vào-ra
 - Xử lý dữ liệu: thực hiện các phép toán số học hoặc phép toán logic với các dữ liệu.
 - Điều khiển rẽ nhánh
 - Kết hợp các thao tác trên.

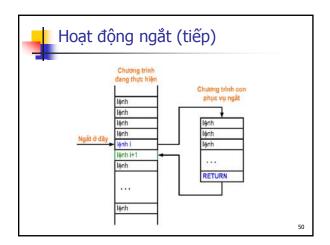
3.4.2. Ngắt (Interupt)

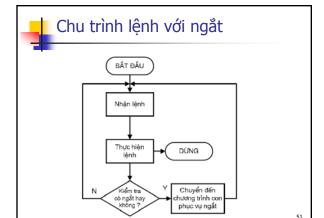
- Khái niệm chung về ngắt: Ngắt là cơ chế cho phép CPU tạm dừng chương trình đang thực hiện để chuyển sang thực hiện một chương trình khác, gọi là chương trình con phục vụ ngắt.
- Các loai ngắt:
 - Ngắt do lỗi khi thực hiện chương trình, ví dụ: tràn số, chia cho 0.
 - Ngắt do lỗi phần cứng, ví dụ: lỗi RAM
 - Ngắt do mô-đun vào-ra phát tín hiệu ngắt đến CPU yêu cầu trao đổi dữ liệu.



Hoạt động ngắt

- Sau khi hoàn thành mỗi môt lênh, bô xử lý kiểm tra tín hiệu ngắt.
- ullet Nếu không có ngắt ightarrow bộ xử lý nhận lệnh tiếp theo của chương trình hiện tại.
- Nếu có tín hiệu ngắt:
 - Tạm dừng chương trình đang thực hiện
 - Cất ngữ cảnh (các thông tin liên quan đến chương trình bị ngắt)
 - Thiết lập PC trỏ đến chương trình con phục vụ ngắt
 - Chuyển sang thực hiện chương trình con phục vụ ngắt.
 - Cuối chương trình con phục vụ ngắt, khôi phục ngữ cảnh và tiếp tục chương trình đảng bị tạm dừng.







Xử lý với nhiều tín hiệu ngắt

- Xử lý ngắt tuần tư
 - Khi một ngắt đang được thực hiện, các ngắt khác sẽ bị
 - Bộ xử lý sẽ bỏ qua các ngắt tiếp theo trong khi đang xử lý một ngắt.
 - Các ngắt vẫn đạng đợi và được kiểm tra sau khi ngắt đầu tiến được xử lý xong
 - Các ngắt được thực hiện tuần tự
- Xử lý ngắt ưu tiên
 - Các ngắt được định nghĩa mức ưu tiên khác nhau
 - Ngắt có mức ưu tiên thấp hơn có thể bị ngắt bởi ngắt uu tiên cao hơn \rightarrow xảy ra ngắt lồng

52



3.4.3. Hoạt động vào-ra

- Hoat đông vào-ra: là hoat đông trao đổi dữ liêu giữa thiết bị ngoại vị với bên trong máy
- Các kiểu hoat đông vào-ra:
 - CPU trao đổi dữ liêu với mô-đun vào-ra
 - Mô-đun vào-ra trao đổi dữ liêu trưc tiếp với bô nhớ chính.

53



3.5. Phần mềm máy tính (Software Computer)

- Thế nào là phần mềm máy tính ?
 - "Phần mềm là thuật ngữ chung cho các bộ sựu tập có tổ chức dữ liệu và lệnh của máy tính, thường được phân thành hai loại chính: phân mềm hệ thống (system software) cung cấp các chức năng xác định cơ bản của máy tính và phân mềm ứng dụng (application software) được sử dụng bởi người dùng để hoàn thành những nhiệm vụ xác định."

penprojects.org/software-definition.htm

- "Phần mềm là thuật ngữ tổng quát cho rất nhiều loại chương trình khác nhau được sử dụng để thao tác với máy tính và các thiết bị liên quan."
- Phân biệt phần mềm (software) với phần cứng (hardware)



3.5.1. Dữ liệu và giải thuật

- Mỗi bài toán phải giải quyết gồm 2 phần:
 - phần dữ liêu
 - phần xử lý
- Phần dữ liệu liên quan đến thông tin của bài toán:
 - đầu vào: dữ liêu được cung cấp để xử lý
 - dầu ra: kết quả xử lý
- Phần xử lý: những thao tác phải được máy tính tiến hành nhằm đáp ứng yêu cầu của người dùng.

55



3.5.2. Chương trình và ngôn ngữ lập trình

- Thuật toán mới chỉ ra cách giải quyết một bài toán theo kiểu tư duy của con người. Để máy có thể hiểu và tiến hành xử lý được ta phải biến các bước thao tác thành các chỉ thị (statement) và biểu diễn trong dạng mà máy tính hiểu được. Quá trình này gọi là *lập trình*. Giải thuật được biểu diễn dưới dạng một tập các chỉ thị của một ngôn ngữ nào đó gọi là *chương trình*. Ngôn ngữ dung để lập trình gọi là *ngôn ngữ lập trình* ngôn ngữ dùng để trao đổi với máy tính, máy tính hiểu và thực thi nhiệm vụ đã chỉ ra.
- Tương tự với dữ liệu, máy tính không thể xử lý dữ liệu một cách hình thức như trong giải tích mà nó phải là những con số hay những giá trị cụ thể.

56



Chương trình

Chương trình = Cấu trúc dữ liệu + Giải thuật

Program = Data Structure + Algorithm

N. Wirth

5



Ngôn ngữ lập trình

- Có nhiều loại ngôn ngữ lập trình. Sự khác nhau giữa các loại liên quan đến mức độ phụ thuộc của chúng vào kiến trúc và hoạt động máy tính, phụ thuộc vào lớp/lĩnh vực ứng dụng. Có nhiều cách phân loại khác nhau và do đó các ngôn ngữ lập trình được phân thành các nhóm khác nhau. Người ta phân các ngôn ngữ theo một cách chung nhất thành 3 nhóm:
 - Ngôn ngữ máy
 - Hợp ngữ
 - Ngôn ngữ bâc cao

58



Ngôn ngữ máy

Mỗi loại máy tính đều có ngôn ngữ máy riêng. Đó chính là loại ngôn ngữ duy nhất để viết chương trình mà máy tính hiểu trực tiếp và thực hiện được. Các chỉ thị (lệnh) của ngôn ngữ này viết bằng mã nhị phân hay mã hec-xa. Nó gắn chặt với kiến trúc phân cứng của máy và do vậy nó khai thác được các đặc điểm phần cứng. Tuy nhiên, nó lại không hoàn toàn thuận lợi cho người lập trình do tính khó nhớ của mã, tính thiếu cấu trúc,...Vì thế, để viết một ứng dung bằng ngôn ngữ máy thì quả là việc không dễ, nhất là phải tiến hành các thay đổi,chinh sửa hay phát triển thêm về sau.



Hơp ngữ

Hợp ngữ cho phép người lập trình sử dụng một số từ tiếng Anh viết tắt để thể hiện các câu lệnh thực hiện. Thí dụ để cộng nội dung của 2 thanh ghi AX và BX rồi ghi kết quả vào AX, ta có thể dùng câu lệnh hợp ngữ sau:

ADD AX, BX

 Một chương trình hợp ngữ phải được dịch ra ngôn ngữ máy nhờ chương trình hợp dịch trước khi máy tính có thể thực hiện.



Ngôn ngữ bậc cao

- FORTRAN, COBOL, Pascal, C/C++, VB, VC++, Delphi, Java, .NET,...
- Các chương trình viết trong ngôn ngữ này, trước khi để máy có thể thực thi cần phải chuyển đổi sang ngôn ngữ máy. Quá trình chuyển đổi đó gọi là quá trình dịch.

61

63



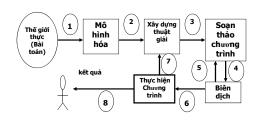
Hai phương thức dịch

- Thông dịch (Interpreter): Bộ thông dịch, đọc từng lệnh của chương trình nguồn, phần tích cú pháp của câu lệnh đó và nếu đúng thì thực hiện. Quá trình bắt đầu từ lệnh đầu tiến của chương trình đến lệnh cuối cùng nếu không có lỗi. Bộ thông dịch này giống như vai trò của 1 thông dịch viên (translator).
- Biển dịch (Compiler): Khác với thông dịch, trình biên dịch dịch toàn bộ chương trình nguồn sang ngôn ngữ địch. Với chương trình địch này, máy đã có thể hiểu được và biết cách thực thi. Quá trình biên dịch sẽ tạo ra chương trình địch chỉ khi các lênh trong chương trình nguồn không có lỗi.

62



Quy trình giải quyết một bài toán trên máy tính



Quy trình giải quyết...(tiếp)

- B1 Xác định bài toán: Thuật ngữ mới cho bước này là xác định yêu cầu người dùng, người mong muốn có phần mềm để sử dụng.
- B2 Phân tích bài toán: Tìm hiểu nhiệm vụ (chức năng) mà phần mềm cần xây dựng phải có và các dữ liệu cần thiết. Qua đó xây dựng các giải pháp khả thi. Nói một cách ngắn gọn, bước này tìm hiểu hệ thống là gì? Và làm gì?
- B3 Thiết kế hệ thống: thực hiện thiết kế kiến trúc hệ thống, thiết kế các mô đun chương trình, thiết kế giao tiếp, thiết kế an toàn,... Như vậy, nhiệm vụ thiết kế mô đun chính là xây dựng giải thuật cho mô đun đó và cách diễn tả giải thuật.



Quy trình giải quyết...(tiếp)

- B4 Xây dựng chương trình: Viết code cho các mô đun theo ngôn ngữ lập trình đã xác định.
- B5 Quay lại soạn thảo: khi quá trình dịch phát hiên lỗi cú pháp trong chương trình nguồn
- B6 Kiểm thử chương trình: nhằm kiểm tra tính đúng đắn của từng mô đun và cả hệ thống trước khi bàn giao cho khách hàng.
- B7 Xem lại giải thuật khi kết quả thực hiện không đúng (lỗi lôgíc).
- B8 Triển khai: bước này gồm cả nhiệm vụ viết tài liệu phần mềm, hướng dẫn sử dụng và bảo trì phần mềm. Đây cũng là mục đích của phần mềm được yêu cầu và nhằm kẻo dài vòng đời phần mềm (Software Life Cycle).



3.5.3. Phân loại phần mềm máy tính

- Theo quan điểm sử dung chung:
 - Phần mềm hệ thống: Là phần mềm điều khiển hoạt động bên trong của máy tính và cung cấp môi trường giao tiếp giữa người dùng và máy tính nhằm khai thác hiệu quả phần cứng phục vụ cho nhu cầu sử dụng. Loại phần mềm này đòi hỏi tính ổn định, tính an toàn cao. Chẳng hạn các hệ điều hành máy đơn hay hệ điều hành mạng, các tiện ích hệ thống,...
 - Phần mềm ứng dụng. Là phần mềm dùng để giải quyết các vấn đề phục vụ cho các hoạt động khác nhau của con người như quản lý, kế toán, soạn thảo văn bản, trò chơi.... Nhu cầu về phần mềm ứng dụng ngày càng tăng và đa dạng.



Phân loại phần mềm máy tính (tiếp)

- Theo đặc thù ứng dụng và môi trường:
 - Phần mềm thời gian thực (Real-time SW)
 - Phần mềm nghiệp vu (Business SW)
 - Phần mềm tính toán KH&KT (Eng.&Scie. SW)
 - Phần mềm nhúng (Embedded SW)
 - Phần mềm trên Web (Web-based SW)
 - Phần mềm trí tuê nhân tao (IA SW)
 -

67



Virus máy tính?

- Nó là cái gì ? Là một chương trình máy tính (do con người viết ra) có khả năng tự nhân bản và lây nhiễm và gây hại cho máy tính, không được sự cho phép của người dùng.
- Giống như virus sinh học, virus máy tính có khả năng nhân bản, lây lan nhanh chóng, có khả năng biến đổi thành các dạng khác, và nói chung là có hại.
- Virus máy tính thường phá hủy dữ liệu, làm sai lệch thông tin, ăn cắp thông tin cá nhân phục vụ những ý đồ xãu
- Lây lan qua: đĩa mềm, CD, ổ USB, thư điện tử,...
- Thường nghe nói: malware, adware, worms, Trojan

68



Phòng và diệt virus?

- Sử dụng các phần mềm cảnh báo và diệt virus, phần mềm gián điệp như: Nortan Anti Virus, Kaspersky, Bit Defender, BKAV,...
- Cảnh giác với các thư lạ, những thông tin mời mọc hấp dẫn trên mạng,...
- Cảnh giác với các ổ đĩa chứa dữ liệu không rõ ràng.
- Nói chung, vẫn khó tránh. Trong trường hợp bị lây nhiễm mà không tự khắc phục được, dữ liệu lại quan trong thì nên tìm đến các chuyên gia.

69





Lời hay ý đẹp

"Đường đi không khó vì ngắn sông cách núi, mà khó vì lòng người ngại núi e sông"

Nguyễn Bá Học

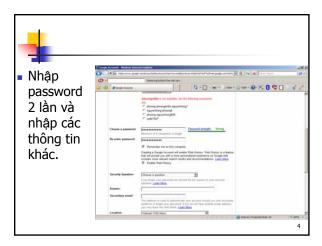
"Nếu anh thấy một gia đình hạnh phúc, anh nên tin rằng ở trong gia đình có một người đàn bà biết quên mình"

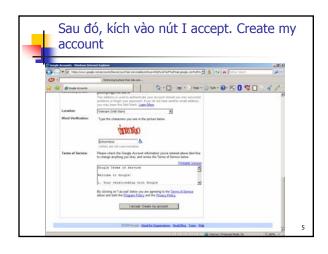
René Bazin















Tạo yahoo mail (của hãng Yahoo)

- Vào trang: www.mail.yahoo.com
- Làm tương tự như với gmail đã hướng dẫn ở trên.



Web mail vs. POP3



Sử dụng gmail

Tạo thư: ComposeXem thư đến: Inbox

Xem thư đã gửi: Sent mail

Thư rác: SpamThùng rác: Trash





Chương 4: Mạng máy tính

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn
Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Nội dung chương này

- 4.1. Lịch sử phát triển, khái niệm mạng máy tính
- 4.2. Phân loại mạng máy tính
- 4.3. Các thành phần cơ bản của một mạng máy tính
- 4.4. Mang Internet

2



4.1. Lịch sử phát triển của mạng máy tính

- Máy tính ra đời từ những năm 1950. Đến đầu những năm 1960 mạng máy tính bắt đầu xuất hiện. Lúc đầu mạng có dạng là một máy tính lớn nối với nhiều trạm cuối (terminal). Đến đầu những năm 1970 mạng máy tính là các máy tính độc lập được nối với nhau. Qui mô và mức độ phức tạp của mạng ngày càng tăng.
- Hiện nay mạng máy tính phát triển rất mạnh ở mọi lĩnh vực mọi nơi. Ngày càng hiếm các máy tính đơn lẻ, không nối mạng. Ngay các máy tính cá nhân ở gia đình cũng được kết nối Internet qua đường điện thoại. Mạng trở thành một yếu tố không thể thiếu của công nghệ thông tin nói riêng, cũng như đời sống nói chung.



4.1. Khái niệm mang máy tính

- Mạng máy tính hay mạng (computer network, network) là một tập hợp gồm nhiều máy tính hoặc thiết bị xử lý thông tin được kết nối với nhau qua các đường truyền và có sự trao đổi dữ liệu với nhau.
- Nhờ có mạng máy tính, thông tin từ một máy tính có thể được truyền sang máy tính khác. Có thể ví mạng máy tính như một hệ thống giao thông vận tải mà hàng hoá trên mạng là dữ liệu, máy tính là nhà máy lưu trữ xử lý dữ liệu, hệ thống đường truyền như là hệ thống đường sá giao thông.
- Ví dụ về mạng máy tính: mạng tại TTMT-Khoa CNTT, mạng của Tổng cục thuế, mạng Internet,...



4.2. Phân loại mạng máy tính

- Theo mối quan hệ giữa các máy trong mạng
 - Mạng bình đẳng (peer-to-peer) các máy có quan hệ ngang hàng, một máy có thể yêu cầu một máy khác phục vụ.
 - Mạng khách/chủ (client/server). Một số máy là server (máy chủ) chuyên phục vụ các máy khác gọi là máy khách (client) hay máy trạm (workstation) khi có yêu cầu. Các dịch vụ có thể là cung cấp thông tin, tính toán hay các dịch vụ Internet.



4.2. Phân loại mạng máy tính (tiếp)

- Theo qui mô địa lý
 - LAN (Local Area Network) mạng cục bộ ở trong phạm vi nhỏ, ví dụ bán kính 500m, số lượng máy tính không quá nhiều, mạng không quá phức tạp.
 - WAN (Wide Area Network) mạng diện rộng, các máy tính có thể ở các thành phố khác nhau. Bán kính có thể 100-200 km. Ví dụ mạng của Tổng cục thuế.
 - GAN (Global Area Network) mạng toàn cầu, máy tính ở nhiều nước khác nhau. Thường mạng toàn cầu là kết hợp của nhiều mạng con. Ví dụ mạng Internet.







4.3. Các thành phần cơ bản của mạng máy tính

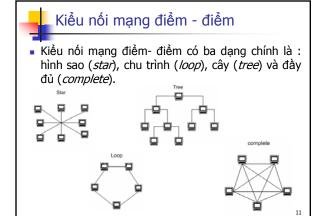
- Hệ điều hành mạng: Hệ điều hành mạng là một phần mềm điều khiển sư hoạt động
- Các phần mềm mạng cho máy tính: Hiện nay nói chung các hệ điều hành đều sẵn có khả nặng kết nối mạng. Trong trường hợp hệ điều hành của máy tính không có sẵn khả năng kết nối mang thì các phần mềm này là cần thiết.
- Các ứng dụng trên mạng: Ví dụ như Email, hệ quản trị cơ sở dữ liệu.

Kiến trúc mạng máy tính

- Kiến trúc mạng máy tính (network architecture) thể hiện cách kết nối máy tính với nhau và qui ước truyền dữ liệu giữa các máy tính như thế nào. Cách nổi các máy tính với nhau gọi là *hình trạng* (*topology*) của mạng. Tập các qui ước truyền thống gọi là *giao thức* (*protocol*).
- Có hai kiểu nối mạng chủ yếu là điểm-điểm (point to point) và quảng bá (broadcast). Trong kiểu điểm-điểm các đường truyền nối các nút thành từng cặp. Như vậy một nút sẽ gửi dữ liệu đến nút lân cận nó (nút được nối trực tiếp với nó). Nút lân cập cố chuyển tiến dữ liệu như vậy cho đển khi cận sẽ chuyển tiếp dữ liệu như vậy chó đến khi dữ liệu đến đích.

10

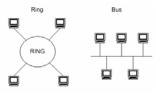






Kiếu nối mạng quảng bá

Trong kiểu quảng bá các nút nối vào đường truyền chung. Như vậy khi một nút gửi dữ liệu các nút còn lại đều nhận được. Do đó dữ liệu gửi đi cần có địa chỉ đích. Khi một nút nhận được dữ liệu nó sẽ kiểm tra địa chỉ đích xem có phải gửi cho mình không.





4.4. Mạng Internet - Khái niệm

- Internet là một mạng máy tính có qui mô toàn cầu (GAN), gồm rất nhiều mạng con và máy tính nối với nhau bằng nhiều loại phương tiện truyền. Internet không thuộc sở hữu của ai cả. Chỉ có các uỷ ban điều phối và kỹ thuật giúp điều hành Internet.
- Ban đầu là mạng của Bộ Quốc phòng Mỹ (DoD) dùng để đảm bảo liên lạc giữa các đơn vị quân đội. Sau đó phát triển thành mạng cho các trường đại học và viện nghiên cứu. Cuối cùng mạng có qui mô toàn cầu và trở thành mạng Internet.

4

Các dịch vụ chính của Internet

- Ta có thể dùng Internet để thực hiện nhiều dịch vụ mạng. Các dịch vụ thông dụng nhất trên Internet hiện nay là:
 - Truyền thông tin (FTP, File Transfer Protocol)
 - Truy nhập máy tính từ xa (telnet)
 - Web (WWW) để tìm kiếm và khai thác thông tin trên mạng
 - Thư điện tử (E-mail)
 - Tán gẫu (Chat) trên mạng...

14



Lơi ích của Internet

Trong thời đại của công nghệ thông tin hiện nay Internet có nhiều lợi ích như truyền tin, phổ biến tin, thu thập tin, trao đổi tin một cách nhanh chóng thuận tiện rẻ tiền hơn so với các phương tiện khác như điện thoại, fax. Internet ảnh hưởng đến toàn bộ thể giới đến mọi ngành, mọi lĩnh vực xã hội. Hiện nay Internet thành một yếu tố quan trọng không thiếu được trong thời đại hiện nay, có mặt ở mọi nơi, mọi lĩnh vực, mọi ngành.

15



Làm sao để có được các dịch vụ Internet

- Để kết nối được Internet ta cần:
 - Máy tính có Modem (Dial-up, ADSL) hoặc card mang.
 - Có thuê bao kết nối với Internet: qua mạng, qua đường điện thoại, đường thuê riêng của bưu điện. Thông thường hiện nay kết nối qua điện thoại hoặc qua ADSL
 - Có tài khoản Internet ở trên mạng hay ở một nhà cung cấp dịch vụ Internet (Internet Service Provider, ISP), ví dụ như VNN, FPT.
 - Có phần mềm Internet thông dụng như Web browser để xem trang web, ví dụ IE, FireFox , phần mềm để xem thư hay chat như Outlook, Messenger.

16



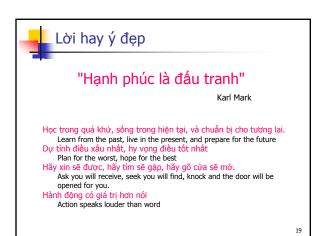
Cần cảnh giác với:

- Virus
- Các thông tin độc hại trên mạng



Hỏi - đáp







Chương 5: Hệ điều hành

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn
Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Nội dung chương này

- 5.1. Các khái niệm cơ bản
 - 5.1.1. Khái niệm hệ điều hành
 - 5.1.2. Tệp (file)
 - 5.1.3. Quản lý tệp của hệ điều hành
- 5.2. Môt số hê điều hành
 - 5.2.1. Hệ điều hành MS-DOS
 - 5.2.2. Hệ điều hành Windows
 - 5.2.3. Hệ điều hành Linux
- 5.3. Hê lênh của hê điều hành

2



Nội dung chương này

- 5.4. Hê điều hành Windows
 - 5.4.1. Sự ra đời và phát triển
 - 5.4.2. Khởi động và thoát khỏi Windows
 - 5.4.3. Một số thuật ngữ và thao tác thường sử dung
 - 5.4.4. Cấu hình Windows (Control Panel)
 - 5.4.5. Windows Explorer
 - 5.4.6. Gọi thực hiện chương trình
 - 5.4.7. Chế độ Command Prompt
 - 5.4.8. Recycle Bin

3



5.1. Các khái niệm cơ bản

- 5.1.1. Khái niệm hệ điều hành
- 5.1.2. Tệp (file)
- 5.1.3. Quản lý tệp của hệ điều hành

4



5.1.1. Khái niệm hệ điều hành

- Hệ điều hành là hệ thống chương trình đảm bảo quản lý tài nguyên của hệ thống tính toán và cung cấp các dịch vụ cho người sử dụng.
- Thông thường trong các máy tính hiện nay, hệ điều hành được cài đặt trên đĩa.
- Nhiệm vụ cụ thể của hệ điều hành là:
 - Khởi động máy tính, tạo môi trường giao tiếp cho người sử dụng.
 - Tự động điều khiển và kiểm soát hoạt động của các thiết bị (ổ đĩa, bàn phím, màn hình, máy in,...).
 - Quản lý việc cấp phát tài nguyên của máy tính như bộ xử lý trung tầm, bộ nhớ, các thiết bị vào ra...
 - Quản lý các chương trình đang thực hiện trên máy tính.
 - Thực hiện giao tiếp với người sử dụng để nhận lệnh và thực hiện lệnh.



5.1.1. Khái niệm hệ điều hành

- Hệ điều hành là phần mềm hệ thống, nên phụ thuộc vào cấu trúc của máy tính. Mỗi loại máy tính có hệ điều hành khác nhau. Ví dụ:
 - Máy tính lớn IBM360 có hệ điều hành là DOS, TOS.
 - Máy tính lớn EC-1022 có hệ điều hành là OC-EC.
 - Máy tính cá nhân PC-IBM có hệ điều hành MS-DOS.
 - Mạng máy tính có các hệ điều hành mạng NETWARE, UNIX, WINDOWS-NT...
 - ...



5.1.2. Tệp (file)

- Tệp là tập hợp các dữ liệu có liên quan với nhau và được tổ chức theo 1 cấu trúc nào đó, thường được lưu trữ bên ngoài máy tính.
- Nội dung của tệp có thể là chương trình, dữ liệu, văn bản,...
- Mỗi tập tin được lưu lên đĩa với một tên riêng phân biệt. Mỗi hệ điều hành có qui ước đặt tên khác nhau, tên tập tin thường có 2 phần:
 - phần tên (name): bắt buộc phải có của một tập tin
 - phần mở rộng (extension): có thể có hoặc không.

7



5.1.2. Tệp (file) (tiếp)

- Phần tên: Bao gồm các ký tự chữ từ A đến Z, các chữ số từ 0 đến 9, các ký tự khác như #, \$, %, ~, ^, @, (,), !, _, khoảng trắng. Phần tên do người tạo ra tập tin đặt. Với MS-DOS phần tên có tối đa là 8 ký tự, với Windows phần tên có thể đặt tối đa 128 ký tự.
- Phần mở rộng: thường dùng 3 ký tự trong các ký tự nêu trên. Thông thường phần mở rộng do chương trình ứng dung tạo ra tập tin tự đặt.
- Giữa phần tên và phần mở rộng có một dấu chấm (.) ngăn cách.

8



Kiểu của file

- Ta có thể căn cứ vào phần mở rộng để xác đinh kiểu của file:
 - COM, EXE: Các file khả thi chạy trực tiếp được trên hệ điều hành.
 - TXT, DOC, ... : Các file văn bản.
 - PAS, BAS, ...: Các file chương trình PASCAL, DELPHI, BASIC, ...
 - WK1, XLS, ...: Các file chương trình bảng tính LOTUS, EXCEL ...
 - BMP, GIF, JPG, ... : Các file hình ảnh.
 - MP3, DAT, WMA, ...: Các file âm thanh, video.

Kí hiệu đại diện (wildcard)

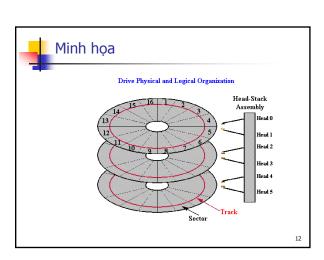
- Để chỉ một nhóm các tập tin, ta có thể sử dụng hai ký hiệu đại diện:
 - Dấu ? dùng để đại diện cho một ký tự bất kỳ trong tên tập tin tại vị trí nó xuất hiện.
 - Dấu * dùng để đại diện cho một chuỗi ký tự bất kỳ trong tên tập tin từ vị trí nó xuất hiện.
- Ví du:
 - Bai?.doc Bai1.doc, Bai6.doc, Baiq.doc, ...
 - Bai*.doc Bai.doc, Bai6.doc, Bai12.doc, Bai Tap.doc, ...
 - BaiTap.* BaiTap.doc, BaiTap.xls, BaiTap.ppt, BaiTap.dbf, ...
- Lưu ý: Nên đặt tên mang tính gọi nhớ

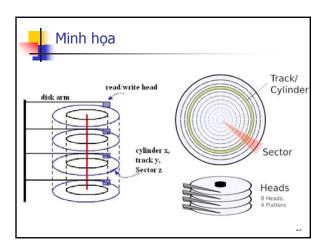
10



5.1.3. Quản lý tệp của hệ điều hành

- Cấu trúc đĩa từ
 - Hệ thống đĩa từ gồm nhiều mặt (side) gắn số hiệu là 0, 1,... Về mặt logic mỗi mặt đĩa có một đầu ghi/ đọc (header), đôi khi người ta còn đồng nhất 2 khái niệm này.
 - Mỗi mặt chia thành các rãnh (track các đường tròn đồng tâm). Các rãnh được đánh số từ ngoài vào trong bắt đầu từ 0.
 - Mỗi rãnh chia thành các cung (Sector), mỗi sector thông thường có dung lượng 512 byte.
 - Một từ trụ (cylinder) gồm các rãnh có cùng bán kính nằm trên các mặt đĩa khác nhau.



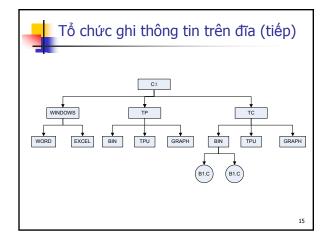




Tổ chức ghi thông tin trên đĩa

- Thông tin lưu trữ trên đĩa dưới dạng các tệp. Mỗi tệp chiếm 1 hoặc nhiều sectors tuỳ dung lượng tệp.
- Để thuận lợi cho việc quản lý tệp, hệ điều hành cho phép chia đĩa thành các vùng, mỗi vùng chia thành các vùng con,.... Mỗi vùng có 1 vùng con riêng để lưu trữ thông tin về vùng đó, vùng con này được gọi là thư mục (Directory). Tệp được lưu trữ ở các vùng, vì vậy ta có thể thấy tổ chức lưu trữ này có dạng cây (Tree). Ví dụ:

14





Thư muc

Thư mục là nơi lưu giữ các tập tin theo một chủ đề nào đó theo ý người sử dụng. Đây là biện pháp giúp ta quản lý được tập tin, dễ dàng tìm kiếm chúng khi cần truy xuất. Các tập tin có liên quan với nhau có thể được xếp trong cùng một thư mục. Sau đây là biểu tượng của thư mục hay còn gọi là Folder trong Windows

16



Thư mục (tiếp)

- Trên mỗi đĩa có một thư mục chung gọi là thư mục gốc. Thư mục gốc không có tên riêng và được ký hiệu là \ (dấu xổ phải: backslash). Dưới mỗi thư mục gốc có các tập tin trực thuộc và các thư mục con. Trong các thư mục con cũng có các tập tin trực thuộc và thư mục con của nó. Thư mục chứa thư mục con gọi là thư mục cha.
- Thư mục đang làm việc gọi là thư mục hiện hành.
- Tên của thư mục tuân thủ theo cách đặt tên của tập tin.



Cách xác định tên đầy đủ của tệp

- Tên tệp đầy đủ gồm nơi lưu trữ tệp đường dẫn từ gốc đến tệp (Path) và tên tệp. Đường dẫn được chỉ ra nhánh cây thư mục chứa tệp, trong đó sử dụng ký hiệu "\" ngăn cách tên các thư mục.
- Ví du:
 - C:\TC\BIN\B1.C



File hệ thống

- Hệ điều hành được phân chia thành các phần, phù hợp với các chức năng riêng của công việc. Những phần này được lưu trên đĩa dưới dạng các tệp (File).
- Ví dụ: Hệ điều hành MS-DOS gồm tập các tệp, trong đó có 3 tệp cơ bản:
 - MSDOS.SYS têp.
 - IO.SYS tệp điều khiển vào ra.
 - COMMAND.COM têp lệnh.

19



5.2. Một số hệ điều hành

Hiện nay có nhiều hệ điều hành khác nhau như MS-DOS, UNIX, LINUX, Windows 95, Windows 98, Windows 2000, Windows XP, Windows 2003, và Windows VISTA là một sản phẩm mới của MicroSoft. Mỗi hệ điều hành có các đặc trưng khác nhau, tuy vậy trong mỗi hệ điều hành có thể tích hợp nhiều hình thái giao tiếp người- máy khác nhau: dòng lệnh, bảng chọn, biểu tượng,...

20



5.2. Một số hệ điều hành (tiếp)

- 5.2.1. Hê điều hành MS-DOS
- 5.2.2. Hệ điều hành Windows
- 5.2.3. Hệ điều hành Linux

21



5.2.1. Hệ điều hành MS-DOS

- Hình thức giao tiếp: văn bản text
- Thực hiện các chức năng bằng câu lênh



22



5.2.2. Hệ điều hành Windows

- Hình thức giao tiếp: đồ họa (bảng chọn, biểu tượng)
- Thực hiện chức năng thông qua giao diện đồ họa hoặc phím tắt.











5.2.3. Hê điều hành Linux

- Hình thức giao tiếp: dòng lệnh, đồ họa
- Thực hiện các chức năng: câu lệnh, giao diên đồ hoa







5.3. Hệ lệnh của hệ điều hành

- Thao tác với tệp: Sao chép, di chuyển, xoá, đổi tên , xem nội dung tệp
- Thao tác với thư mục: tạo, xoá, sao chép
- Thao tác với đĩa: tạo khuôn (Format), sao chép đĩa

2



5.4. Hệ điều hành Windows

- 5.4.1. Sự ra đời và phát triển
- 5.4.2. Khởi động và thoát khỏi Windows
- 5.4.3. Một số thuật ngữ và thao tác thường sử dụng
- 5.4.4. Cấu hình Windows (Control Panel)
- 5.4.5. Windows Explorer

26



5.4.1. Sự ra đời và phát triển

Windows là một bộ chương trình do hãng Microsoft sản xuất. Từ version 3.0 ra đời vào tháng 5 năm 1990 đến nay, Microsoft đã không ngừng cải tiến làm cho môi trường này ngày càng được hoàn thiện.

27



5.4.1. Sự ra đời và phát triển (tiếp)

- Windows 95
- Windows 98, Windows Me
- Windows NT 4.0, WIndows 2000, Windows XP, Windows 2003
- Windows Vista

28



5.4.2. Khởi động và thoát khỏi Windows XP

- Khởi động Windows XP:
 - Tự khởi động khi bật máy
 - Người dùng phải đăng nhập (login): nhập username và password vào
 - Mỗi người dùng có một tập hợp thông tin thiết lập riêng gọi là user profile
- Thoát khỏi Windows XP:
 - Đóng tất cả các cửa sổ, chương trình đang mở
 - Nhán Alt+F4 hoặc chọn Start/ Turn off computer

4

5.4.3. Một số thuật ngữ và thao tác thường sử dụng

- Biểu tượng (icon):
 - là các hình vẽ nhỏ đặc trưng cho một đối tượng nào đó của Windows hoặc của các ứng dụng chạy trong môi trường Windows.
 - Phía dưới biểu tượng là tên biểu tượng. Tên này mang một ý nghĩa nhất định, thông thường nó diễn giải cho chức năng được gán cho biểu tượng (ví dụ nó mang tên của 1 trình ứng dung).

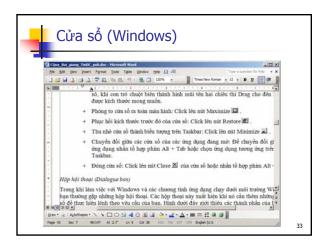


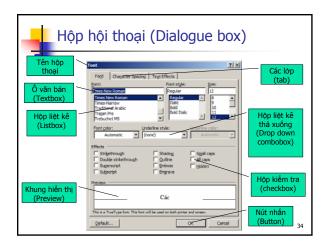


Cửa sổ (Windows)

- Cửa sổ là khung giao tiếp đồ họa của 1 ứng dụng hoặc 1 lệnh.
- Bố cục của 1 cửa sổ: gồm thanh tiêu đề, thanh thực đơn, 1 số thành phần khác phụ thuộc vào loại cửa sổ,...
- Các hộp giao tiếp
- Các thao tác trên một cửa sổ: di chuyển cửa sổ, thay đổi kích thước cửa sổ, phóng to, thu nhỏ, phục hồi kích thước cửa sổ, chuyển đổi qiữa các cửa sổ, đóng cửa sổ...

32







Sử dụng chuột trong windows

- Chuột là thiết bị không thể thiếu khi làm việc trong môi trường Windows XP. Con trỏ chuột (mouse pointer) cho biết vị trí tác động của chuột trên màn hình. Hình dáng của con trỏ chuột trên màn hình thay đổi theo chức năng và chế độ làm việc của ứng dụng. Khi làm việc với thiết bị chuột bạn thường sử dụng các thao tác cơ bản sau:
 - Point: trổ chuột trễn mặt phẳng mà không nhấn nút nào cả.
 - Click: nhấn nhanh và thả nút chuột trái. Dùng để lựa chọn thông số, đối tượng hoặc câu lệnh.
 - Double Click (D Click): nhấn nhanh nút chuột trái hai lần liên tiếp. Dùng để khởi động một chương trình ứng dụng hoặc mở thư mục/ tập tin.



Sử dụng chuột trong windows (tiếp)

- **Drag** (kéo thả): nhấn và giữ nút chuột trái khi di chuyển đến nơi khác và buông ra. Dùng để chọn một khối văn bản, để di chuyển một đối tượng trên màn hình hoặc mở rộng kích thước của cửa số...
- Right Click (R_Click): nhấn nhanh và thả nút chuột phải. Dùng mở menu tương ứng với đối tượng để chọn các lệnh thao tác trên đối tượng đó.
- Chú ý:
 - Đa số chuột hiện nay có bánh xe trượt hoặc nút đẩy ở giữa dùng để cuộn màn hình làm việc được nhanh hơn và thuận tiện hơn.
 - Trong Windows các thao tác được thực hiện mặc nhiên với nút chuột trái, vì vậy để tránh lặp lại, khi nói Click (nhắn chuột) hoặc D Click (nhắn đụp chuột) thì được ngàm hiểu đó là nút chuột trái. Khi nào cần thao tác với nút chuột phải sẽ mô tả rõ ràng.



5.4.4. Cấu hình Windows (Control Panel)

- Giới thiệu về Control Panel
 - Control Panel là một chương trình cho phép người sử dụng xem và chỉnh sửa các tham số của hệ thống máy tính như dạng hiện của dữ liệu ngày tháng, dữ liệu số, thiết lập hoặc thay đổi cấu hình cho phù hợp với công việc hoặc sở thích của người dùng, cài đặt phần cứng, phần mèm.
 - Khởi động: Start / Settings / Control Panel



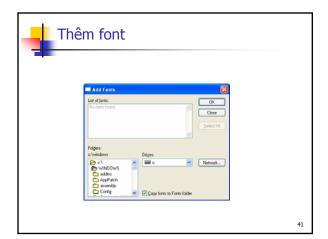


Dể cài đặt thêm những Font chữ khác hoặc loại bỏ các Font chữ, ta chọn chương trình Fonts



Cài đặt và loại bỏ font chữ (tiếp)

- Loai bỏ font chữ. Từ cửa sổ Fonts
 - Chọn những Font cần loại bỏ
 - Chọn File/ Delete (hoặc nhấn phím Delete).
- Thêm font chữ mới Từ cửa sổ Fonts, chọn lệnh File/Install New Font, xuất hiện hộp thoại Add Fonts. Trong hộp thoại này chỉ ra nơi chứa các Font nguồn muốn thêm bằng cách chọn tên ổ đĩa chứa các tập tin Font chữ, sau đó chọn các tên Font và Click OK.

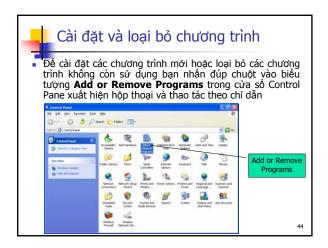


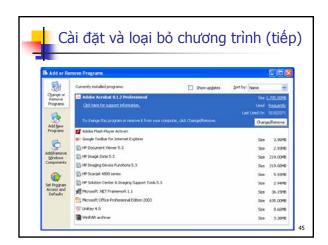


Thay đổi dạng hiện màn hình

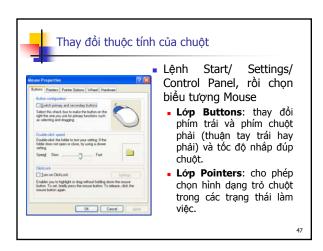
- Chọn lệnh **Start/ Settings/ Control Panel/ Display** hoặc **R_Click trên màn hình** nền (Desktop), chọn Properties. Xuất hiện cửa số Display Properties với các thành phần như sau:
- **Desktop:** Chọn ảnh nền cho Desktop bằng cách Click chọn các ảnh nền có sẵn hoặc Click vào nút Browse để chọn tập tin ảnh không có trong danh sách những ảnh có
- Screen Saver: xác lập màn hình nghỉ
- Screen Saver: xac lạp màn ninh nghi Settings: Thay đổi chế độ màu và độ phân giải của màn hình. Chế độ màu càng cao thì hình ảnh càng đẹp và rõ nét. Các chế độ màu: 64.000 màu (16 bits), 16 triệu màu (24 bits). Chế độ màu trên mỗi máy tính có thể khác nhau tùy thuộc vào dung lượng bộ nhớ của cạrd màn hình. Độ phân giải càng lớn thì màn hình càng hiện thị được nhiều thông tin .

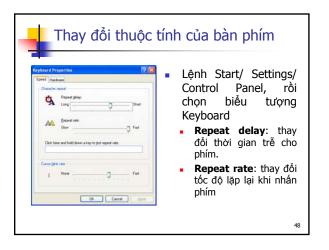














Thay đổi thuộc tính vùng (Regional Settings)

- Lệnh Start / Settings / Control Panel / Regional and Language Options
- Lóp Regional Options: Thay đổi thuộc tính vùng địa lý, sau đó sẽ kéo theo sự thay đổi các thuộc tính của Windows. Click chọn Customize, cửa sổ Customize để thay đổi qui ước về dạng số, tiền tệ, thời gian, ngày tháng.



Số, tiền tệ, ngày

Number: Thay đổi định dạng số, cho phép định dạng việc hiển thị

Decinal symbol: Thay đổi ký hiệu phân cách hàng thập phân.

No, of digits after decimal: Thay đổi số các số lẻ ở phân thập phân.

Digit grouping symbol: Thay đổi ký hiệu phân niôm hàng ngân.

Digit grouping: Thay đổi số ký số trong một niôm (3 số / 4 số / ...)

Negative sign symbol: Thay đổi ký hiệu của số âm.

Negative sign symbol: Thay đổi dạng thể hiện của số âm.

Negative sumber format: Thay đổi dạng thể hiện của số âm.

Display leading zeros: 0.7 hay .7.

Meassurement system: Chọn hệ thống đo lường như cm, inch, ...

List separator: Chọn đầu phân cách giữa các mục trong đạnh sách

Currency: Thay đổi định dạng tiền tệ (\$,VND,...)

Time: Thay đổi định dạng giớ theo chế độ 12 giờ hay 24 giờ

Date: Thay đổi định dạng ngày tháng (Date), cho phép chọn cách thể hiện najay



Cài đặt / loại bỏ máy in

- Cài đặt thêm máy in:
 - Với một số máy in thông dụng Windows đã tích hợp sẵn chương trình điều khiển (driver) của các máy in, tuy nhiên cũng có những máy in mà trong Windows chưa có chương trình điều khiển. Muốn sử dụng những máy in này ta cần phải gọi thực hiện chương trình **Printers** and Faxes trong Control Panel.
- Các bước cài đặt máy in:
 - Chon lệnh Start/ Settings/ Printers and Faxes
 - Click chon Add a Printer, xuất hiện hộp thoại Add
 - Làm theo các bước hướng dẫn của hệ thống

51



Cài đặt / loại bỏ máy in (tiếp)

- Loại bỏ máy in đã cài đặt
 - Chọn lệnh Start/ Settings/ Printers and Faxes
 - Click chuột chọn máy in muốn loại bỏ
 - Nhấn phím Delete, sau đó chọn Yes

52



5.4.5. Windows Explorer

- Là một chương trình được hỗ trợ từ phiên bản Windows 95 cho phép người sử dụng thao tác với các tài nguyên có trong máy tính như tập tin, thư mục, ổ đĩa và những tài nguyên khác có trong máy của bạn cũng như các máy tính trong hệ thống mạng (nếu máy tính của bạn có nối mạng).
- Với Windows Explorer, các thao tác như sao chép, xóa, đổi tên thư mục và tập tin,... được thực hiện một cách thuận tiên và dễ dàng



Khởi động chương trình Windows Explorer

- Thực hiện một trong những cách sau:

 - R_Click lên Start, sau đó chọn Explore
 - R_Click lên biểu tượng My Computer, sau đó chọn Explore ...

54





Cửa sổ làm việc của Windows Explorer

- Cửa sổ trái (Folder) là cấu trúc cây thư mục. Nó trình bày cấu trúc thư mục của các đĩa cứng và các tài nguyên kèm theo máy tính, bao gồm ổ đĩa mềm, ổ đĩa cứng, ổ đĩa CD...
 - Những đối tượng có dấu cộng (+) ở phía trước cho biết đối tượng đó còn chứa những đối tượng khác trong nó nhưng không được hiễn thị. Nếu Click vào dấu + thì Windows Explorer sẽ hiển thị các đối tượng chứa trong đối tượng đó. Khi đó, dấu + sẽ đổi thành dấu -, và nếu Click vào dấu - thì đối tượng sẽ được thu gọn trở lại.
- Cửa sổ phải liệt kê nội dung của đối tượng được chọn tương ứng bên cửa sổ trái

56



Thanh địa chỉ (Address Bar)

 Cho phép nhập đường dẫn thư mục/ tập tin cần tới hoặc để xác định đường dẫn hiện hành

57





Thao tác với thư mục và tệp

- Mở tập tin, thư mục
- Chọn tập tin, thư mục
- Tao thư mục
- Sao chép tâp tin, thư muc
- Di chuyển tập tin, thư mục
- Xóa tập tin, thư mục
- Phục hồi tập tin, thư mục
- Đổi tên tập tin, thư mục
- Thay đổi thuộc tính tập tin, thư mục

Mở tập tin, thư mục

- Có ba cách thực hiện :
 - Cách 1: D_Click lên biểu tượng của tập tin/ thư mục.
 - Cách 2: R Click lên biểu tượng của tập tin/ thư mục và chọn mục Open.
 - Cách 3: Chọn tập tin/ thư mục và nhấn phím Enter.
- Néu tập tin thuộc loại tập tin văn bản thì chương trình ứng dụng két hợp sẽ được khởi động và tài liệu sẽ được nạp vào.
- Trong trường hợp chương trình ứng dụng không được cài đặt trong máy tính thì Windows sẽ mở hộp thoại Open With và cho chọn chương trình kết hợp. Nếu tập tin thuộc dạng chương trình ứng dụng thì chương trình tương ứng sẽ được khởi đồng



Chọn tập tin, thư mục

- Chon môt tâp tin/ thư muc: Click lên biểu tương tập tin/ thư mục.
- Chọn một nhóm tập tin/ thư mục: có thể thực hiện theo 2 cách:
 - Các đối tượng cần chọn là một danh sách gồm các đối tượng liên tục: Click lên đối tượng đầu danh sách để chọn, sau đó nhấn giữ phím Shift và Click lên đối tượng ở cuối danh sách.
 - Các đối tượng cần chọn nằm rời rạc nhau: nhấn giữ phím Ctrl và Click chọn các đối tượng tương ứng.



Tạo thư mục

- Chọn nơi chứa thư mục cần tạo (thư mục / ổ đĩa ở cửa sổ bên trái).
- Chọn menu **File/ New/ Folder** hoặc R Click/ New/ Folder.
- Nhập tên thư mục mới, sau đó gõ Enter để kết thúc



Sao chép tập tin và thư mục

- Chọn các thư mục và tập tin cần sao chép. Sau đó có thể thực hiện theo một trong hai cách sau:
 - Cách 1: Nhấn giữ phím Ctrl và Drag đối tượng đã chọn đến nơi cần chép.
 - Cách 2: Nhấn tổ hợp phím Ctrl + C (hoặc Edit/ Copy hoặc R_Click và chọn Copy) để chép vào Clipboard, sau đó chọn nơi cần chép đến và nhấn tổ hợp phím Ctrl + V (hoặc Edit/ Paste hoặc R_Click và chọn Paste)



Di chuyển tập tin và thư mục

- Chọn các thư mục và tập tin cần di chuyển. Sau đó có thể thực hiện theo một trong hai cách sau:
 - Cách 1: Drag đối tượng đã chọn đến nơi cần di chuyển.
 - Cách 2: Nhấn tổ hợp phím Ctrl + X (hoặc Edit/ Cut hoặc R_Click và chọn Cut) để chép vào Clipboard, sau đó chọn nơi cần di chuyển đến và nhấn tổ hợp phím Ctrl + V (hoặc Edit/ Paste hoặc R_Click và chọn Paste).



Xóa tập tin và thư mục

- Chọn các thư mục và tập tin cần xóa.
- Chon File/ Delete
- hoặc: Nhấn phím Delete
- hoăc: R Click và chon muc Delete.
- Xác nhận có thực sự muốn xoá hay không (Yes/No)



Phục hồi tập tin và thư mục

- Các đối tượng bị xóa sẽ được đưa vào Recycle Bin. Nếu muốn phục hồi các đối tượng đã xóa, bạn thực hiện các thao tác sau đây:
- D_Click lên biểu tượng Recycle Bin
- Chọn tên đối tượng cần phục hồi.
- Thực hiện lệnh File/ Restore hoặc R Click và chon muc Restore.
- Chú ý: Nếu muốn xóa hẳn các đối tượng, ta thực hiện thao tác xóa một lần nữa đối với các đối tượng ở trong Recycle Bin. Nếu muốn xoá hẳn tất cả cắc đối tượng trong Recycle Bin, R_Click lên muc Recycle Bin và chọn mục Empty Recycle Bin.



Đổi tên tập tin và thư mục

- Chọn đối tượng muốn đổi tên
- Thực hiện lệnh File/ Rename hoặc nhấn phím F2 hoặc R_Click trên đối tượng và chon muc Rename.
- Nhập tên mới, sau đó gõ Enter để kết thúc.
- <u>Chú ý:</u> với tập tin đang sử dụng thì các thao tác di chuyển, xoá, đổi tên không thể thực hiện được.

67



Thay đổi thuộc tính tập tin và thư mục

- Nhấn chuột phải lên đối tượng muốn thay đổi thuộc tính và chọn mục **Properties**
- Thay đổi các thuộc tính.
- Chọn Apply để xác nhận thay đổi, ngược lại thì nhấn Cancel.

68



5.4.6. Gọi thực hiện chương trình

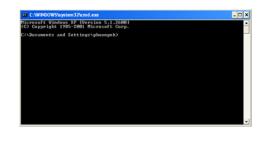
- Nếu là chương trình thực thi thì ta:
 - Kích đúp vào biểu tượng của nó
 - Hoặc Start / Run rồi gõ tên chương trình vào

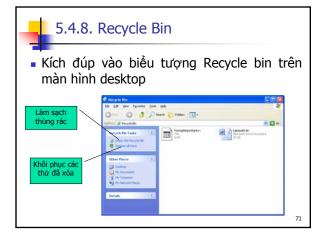


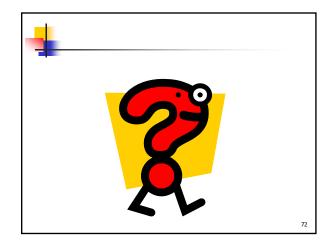


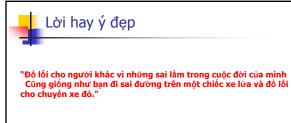
5.4.7. Chế độ Command Prompt

Start / Run, gõ vào cmd rồi nhấn OK









http://www.conduongthanhcong.com/



Chương 6: Các hệ thống ứng dụng

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viên Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Nội dung chương này

- 6.1. Hệ thống thông tin quản lý
- 6.2. Hệ thông tin bảng tính
- 6.3. Hệ quản trị cơ sở dữ liệu
- 6.4. Các hệ thống thông minh

2



6.1. Hệ thống thông tin quản lý

- Khái niêm
 - Hệ thống thông tin quản lý là hệ thống bao gồm phần cứng, phần mềm, con người, quy trình thu thập, phân tích, xử lý, đánh giá và phân phối, chia sẻ những thông tin cần thiết một cách kịp thời và chính xác dựa trên nhu cầu của tổ chức.
 - HTTTQL thủ công: sử dụng giấy, bút, không sử dung máy tính
 - Năm thành phần cơ bản: (1) cơ sở hạ tầng (phần cứng và hệ thống truyền thông), (2) phần mềm, (3) cơ sở dữ liệu, (4) quy trình và (5) nhân sự.



6.1. Hệ thống thông tin quản lý

- Chức năng
 - Nhập dữ liệu
 - Xử lý thông tin
 - Xuất dữ liêu
 - Lưu trữ thông tin
 - Thông tin phản hồi

4



6.1. Hệ thống thông tin quản lý

- Các dang thông tin
 - Theo quan điểm cá nhân
 - Theo quan điểm tổ chức
- Đặc tính của thông tin
 - Chính xác
 - Đầy đủ
 - Thống nhất
 - Thích hợp và dễ hiểu
 - Kịp thời

_



6.1. Hệ thống thông tin quản lý

- Xây dựng và phát triển HTTT: phương pháp chu kỳ hệ thống SDLC (Systems Development Life Cycle)
 - Lập kế hoạch
 - Phân tích
 - Thiết kế
 - Cài đặt
 - Kiểm định
 - Vận hành
 - Bảo trì



6.2. Hệ thông tin bảng tính

- Máy tính: Hỗ trợ việc tính toán, nhất là kế toán và phân tích thống kê.
- Phần mềm thông dụng: Phầm mềm bảng tính (PMBT) spreadsheet software
- PMBT: giúp tính toán các số liệu, từ đó cho phép xây dựng và làm việc với những tình huống mô phỏng thế giới thực.



Hệ thông tin bảng tính

Bảng tính - phần mềm của dự toán

- Tạo thay đổi lớn trong hoạt động kinh doanh
- Giúp thao tác với con số, phương thức khó làm bằng tay
- Rút ngắn khoảng cách thời gian thực hiện
- Giúp khám phá mối liên hệ giữa các con số => cơ sở dư đoán tương lai

8

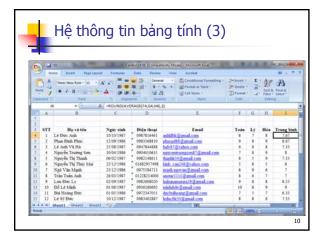


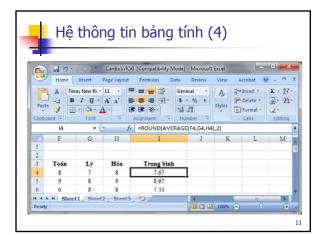
Hệ thông tin bảng tính (2)

Bảng tính: những ô lưới linh động

- Dạng ô lưới gồm: Các hàng đánh số từ 1
 và Các côt đánh số từ chữ A.
- Ô là giao của 1 hàng và 1 cột. Ví dụ ô A1 là giao của hàng 1 và cột A.
- Mỗi ô có thể chứa dữ liệu dạng số, chuỗi kí tự hoặc công thức hiển thị liên hệ giữa các con số.
- Giá trị số là vật liệu thô để tính toán

9







Hệ thông tin bảng tính (5)

- Các chức năng cơ bản của PMBT:
 - Tự động lặp các giá trị, tiêu đề và công thức: Giúp đơn giản hóa việc nhập các dữ liệu lặp.
 - Tự động tính lại: Khi có một sự thay đổi tại 1 ô thì toàn bộ bảng tính sẽ được tính toán lại.
 - Các hàm thư viện: thực hiện các công việc tính toán đã định sẵn. Giúp tiết kiệm thời gian và giảm nguy cơ phát sinh lỗi.



Hệ thông tin bảng tính (6)

- Các chức năng cơ bản của PMBT:
 - Macro: Giúp "thu" lại các thao tác lặp đi lặp lại và định nghĩa nó là 1 macro. Khi cần thực hiện các thao tác đó thì chỉ việc gọi macro tương ứng.
 - Bảng tính mẫu: Chỉ bao gồm các tiêu đề và công thức nhưng không chứa dữ liệu. Giúp tiết kiệm thời gian và công sức.
 - Liên kết: Cho phép tạo liên kết động giữa các bảng tính.
 - Cơ sở dữ liệu: Cho phép thao tác: lưu trữ và truy cập thông tin, tìm kiếm, báo cáo,...

13

15



Hệ thông tin bảng tính (7)

- Những đặc điểm nổi bật khác:
 - Công cụ giải phương trình, những bài toán tối ưu.
 - Lotus hỗ trợ Multimedia, Excel sử dụng trí tuê nhân tao...
 - Vẽ đô thị: từ các con số chuyển thành đô thị để biểu đạt thông tin: đồ thị tròn, đồ thi đường, đồ thi côt...

14



Hệ thông tin bảng tính (8)

- Kinh nghiêm sử dung:
 - Hãy hình dung bảng tính trước khi bạn đưa ra các giá trị và công thức vào
 - Kiểm tra nhiều lần mỗi công thức và giá trị
 - Làm bảng tính trở nên dễ đọc.
 - Kiểm tra kết quả bằng những cách khác
 - Xây dựng các hàm kiểm tra chéo
 - Đổi giá trị đầu vào và quan sát kết quả
 - Hãy tận dụng những hàm có sẵn
 - PMBT hỗ trợ quyết định chứ không thay quyết định.



6.3. Hệ quản trị cơ sở dữ liệu

- Khái niêm hê quản tri cơ sở dữ liêu
- Các tính năng của hệ quản trị cơ sở dữ liệu

16



Khái niệm cơ sở dữ liệu

- Là một tập hợp các dữ liệu
 - Biểu diễn một vài khía cạnh nào đó của thế giới thực
 - Có liên hệ logic thống nhất
 - Được thiết kể và bao gồm những dữ liệu phục vụ một mục đích nào đó.
- Là một bộ sưu tập các dữ liệu tác nghiệp được lưu trữ lại và được các hệ ứng dụng của một xí nghiệp cụ thể nào đó sử dụng.
- Là một hệ thống các thông tin có cấu trúc, lưu trữ trên các thiết bị lưu trữ thông tin.
- Ví dụ:
 - Trang niên giám điện thoại
 - Danh sách sinh viên.
 - Hệ thống tài khoản ngân hàng.

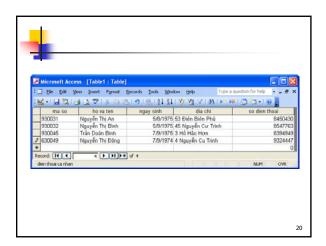


- Ưu điểm khi sử dụng CSDL:
 - việc lưu trữ một lượng thông tin khổng lồ trở nên dễ dàng.
 - Giúp nhanh chóng và mềm dẻo trong việc tra cứu thông tin.
 - Giúp dễ dàng sắp xếp và tổ chức thông tin
 - Giúp in và phân phối thông tin theo nhiều cách.



- Bên trong cơ sở dữ liệu:
 - Một CSDL được hình thành từ các file chứa một tập thông tin có liên quan.
 - Một file CSDL bao gồm:
 - Nhiều bản ghi (record): 1 bản ghi là thông tin liên quan đến 1 người, 1 sản phẩm hoặc 1 sự kiện nào đó.
 - Nhiều trường (field): Mỗi 1 đoạn thông tin riêng rẽ trong 1 record là 1 trường.
 - Ví dụ: 1 record trong csdl thư viện có các field cho tác giả, tựa đề sách, nhà XB, địa chỉ, ...
 - Mỗi trường được xác định bằng kiểu cụ thể: có các kiểu ngày, chữ, số,...

19





Hệ quản trị cơ sở dữ liệu

- Là một hệ thống phần mềm cho phép
 - Định nghĩa, tạo lập: xác định kiểu, cấu trúc, ràng buộc dữ liệu, lưu trữ dữ liệu trên các thiết bi nhớ.
 - Thao tác: truy vấn, cập nhật, kết xuất,...
 các CSDL cho các ứng dụng khác nhau
- Ví dụ: MS SQL Server, DB2, MS Access, Oracle, FoxPro,...



21



Các tính năng của hệ quản trị CSDL

- Quản lý dữ liệu tồn tại lâu dài
- Truy xuất dữ liệu một cách hiệu quả
- Hỗ trợ ít nhất một mô hình dữ liệu
- Đảm bảo tính độc lập dữ liệu, toàn vẹn dữ liệu
- Hỗ trợ các ngôn ngữ cấp cao nhất định cho phép người sử dụng định nghĩa cấu trúc của dữ liệu, truy nhập và thao tác dữ liệu
- Quản trị giao dịch
- Điều khiển truy nhập
- Sao lưu và phục hồi dữ liệu

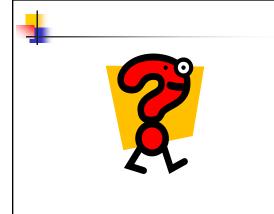


2



6.4. Các hệ thống thông minh

- Hê dưa trên tri thức, hê chuyên gia
- Tác tử thông minh
- Công nghệ cảm biến, công nghệ số: ngôi nhà thông minh, máy móc, trang thiết bị thông minh.



Lời hay ý đẹp

"Không có điều vĩ đại nào được thực hiện mà không có những con người vĩ đại. Và họ chi vĩ đại nếu họ quyết tâm trở thành người vĩ đại"

http://www.conduongthanhcong.com/



Phần II: Giải quyết bài toán

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn
Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viên Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Nội dung phần này

- Chương 1: Giải quyết bài toán bằng máy tính
 - Khái niệm về bài toán
 - Quá trình giải quyết bài toán bằng máy tính
 - Các phương pháp giải quyết bài toán bằng máy tính
 - Phân loại bài toán
- Chương 2: Thuật toán
 - Định nghĩa thuật toán
 - Biểu diễn thuật toán
 - Môt số thuật toán thông dung
 - Thuật toán đệ quy
 - Thuật giải heuristic

2



Chương 1: Giải quyết bài toán bằng máy tính

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn
Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

3



Nội dung chương này

- 1.1. Khái niêm về bài toán
- 1.2. Các bước giải quyết bài toán bằng máy tính
- 1.3. Các phương pháp giải quyết vấn đề bằng máy tính
- 1.4. Phân loại bài toán

4



1.1. Khái niệm về vấn đề và bài toán

- Vấn đề rông hơn bài toán?
- Pitago chia vấn đề ra:
 - Theorema là vấn đề cần được khẳng định đúng-sai
 - Problema là vấn đề cần tìm giải pháp để đạt được một mục tiêu xác định từ những điều kiện ban đầu.
- Diễn đạt bằng sơ đồ: A → B
 - A là giả thiết, điều kiện ban đầu
 - B là kết luận, mục tiêu cần đạt
 - → là suy luận, giải pháp cần xác định

1.2. Các bước giải quyết bài toán bằng máy tính

- Bước 1: Xác định vấn đề-bài toán
- Bước 2: Lựa chọn phương pháp giải
- Bước 3: Xây dựng thuật toán hoặc thuật giải
- Bước 4: Cài đặt chương trình
- Bước 5: Hiệu chỉnh chương trình
- Bước 6: Thực hiện chương trình

6



1.3. Các phương pháp giải quyết vấn đề bằng máy tính

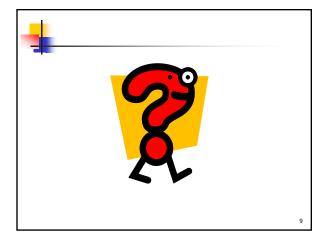
- Giải quyết vấn đề theo hướng xác định trực tiếp lời
 - xác định trực tiếp lời giải qua thủ tục tính toán hoặc thủ tục bao gồm một số hữu hạn các thao tác sơ cấp.
- Giải quyết vấn đề theo hướng tìm kiếm lời giải
 - nguyên lý "thử và sai"
 - các phương pháp
 - liệt kê hay vét cạn
 - thử ngẫu nhiên

 - quay luichia để trị



1.4. Phân loại bài toán

- Bài toán đa thức
- Bài toán không đa thức
- NP Problems





Lời hay ý đẹp

Chân thật và thẳng thắn là bước đầu cần thiết làm nảy nở và củng cố mối quan hệ bạn bè

Lênin



Chương 2: Thuật toán

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Nội dung chương này

- 2.1. Định nghĩa thuật toán
- 2.2. Biểu diễn thuật toán
- 2.3. Một số thuật toán thông dụng
- 2.4. Thuật toán đệ quy
- 2.5. Thuật giải heuristic

2



2.1. Định nghĩa thuật toán

- Là một khái niệm cơ sở của toán học và tin học.
- Bao gồm một dãy hữu hạn các lệnh/chỉ thị rõ ràng và có thể thi hành được để hướng dẫn thực hiện một hành động nhằm đạt được mục tiêu đề ra.
- Thuật toán là sự thể hiện của một phương pháp để giải quyết một vấn đề.

3



Ví dụ 1: Thuật toán tìm phần tử lớn nhất của một dãy hữu hạn các số nguyên

- Các bước:
 - 1. Đặt giá trị lớn nhất tạm thời là số nguyên đầu tiên.
 - 2. So sánh số nguyên kế tiếp trong dãy với giá trị lớn nhất tạm thời, nếu số nguyên này lớn hơn giá trị lớn nhất tạm thời thì đặt giá trị lớn nhất tạm thời bằng số nguyên này.
 - 3. Lặp lại bước 2 nếu còn số nguyên trong dãy chưa được xét.
 - 4. Dừng nếu không còn số nguyên nào trong dãy chưa được xét. Giá trị lớn nhất tạm thời lúc này chính là giá trị lớn nhất trong dãy số.

,



Ví dụ 2: Thuật toán giải phương trình bậc hai: $ax^2 + bx + c = 0$ ($a \neq 0$)

- 1. Nhập 3 hệ số a, b, c
- 2. Tính giá trị $\Delta = b^2 4*a*c$
- 3. Xét dấu Δ. Nếu Δ>0 thì thực hiện các thao tác sau đây:
 - 3.1. Tính các nghiệm theo các công thức:
 - x₁ = (-b-sqrt(△))/(2*a)
 x₂ = (-b+sqrt(△))/(2*a)
 - 3.2. Xuất kết quá: phương trình có hai nghiệm x₁ và x₂.
- 4. Nếu ∆ là 0 thì xuất kết quả: phương trình có nghiệm kép là -b/(2*a)
- 5. Nếu ∆<0 thì xuất kết quả: phương trình vô nghiệm
- 6. Dừng thuật toán

_



Các đặc trưng của thuật toán

- Nhập (input): có các giá trị nhập từ một tập hợp nhất định.
- Xuất (output): từ mỗi giá trị của tập hợp nhập, tạo ra giá trị xuất thuộc một tập hợp nhất định.
- Tính xác định (definiteness): các bước chính xác, rõ ràng.
- Tính hữu hạn (finiteness): cho ra kết quả sau một số hữu hạn bước.
- Tính hiệu quả: được đánh giá dựa trên một số tiêu chuẩn (khối lượng tính toán, không gian, thời gian sử dụng).
- Tính tổng quát: áp dụng được cho tất cả các bài toán có dạng như mong muốn



2.2. Biểu diễn thuật toán

- Sử dụng các ngôn ngữ:
 - Ngôn ngữ tự nhiên
 - Ngôn ngữ lưu đồ (sơ đồ khối)
 - Ngôn ngữ tưa ngôn ngữ lập trình (mã giả)
 - Ngôn ngữ lập trình

Ngôn ngữ lưu đồ

- Các thành phần:
 - Nút giới hạn: được biểu diễn bởi hình ôvan có ghi chữ bên trong, gồm có nút đầu và nút cuối:

BẮT ĐẦU

KẾT THÚC

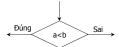
- Nút thao tác: là một hình chữ nhật có ghi các lệnh cần thực hiện: tăng k
- Nút nhập/xuất dữ liệu:

Đọc a và b

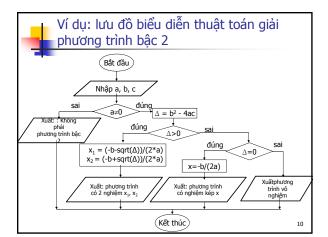


Ngôn ngữ lưu đồ (2)

 Nút điều kiên: là môt hình thoi có ghi điều kiên cần kiểm tra, thường có 1 cung đi vào và 2 cung đi ra (tương ứng với 2 trường hợp đúng/sai)



 Cung: là đường nối từ nút này đến nút khác của lưu đồ





Mã giả

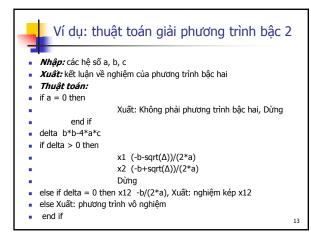
- Sử dung mênh đề có cấu trúc chuẩn hóa và vẫn dùng ngôn ngữ tự nhiên.
- Sử dụng các ký hiệu toán học, các biến, cấu trúc kiểu thủ tục.
- Hành động gán:
 - i ← i+1
- Tiên lơi, đơn giản, vẫn dễ hiểu.



Mã giả (2)

- Các cấu trúc thường gặp:
 - Cấu trúc chon:
 - if (điều kiện) then (hành động) end if
 - if (điều kiện) then (hành động 1) else (hành động 2)
 - end if Cấu trúc lặp
 - while (điều kiện) do (hành động) end while

 - repeat (hành động) until (điều kiện)
 for (biển)=(giá trị đầu) to (giá trị cuối) do (hành động) end for
 for (biển)=(giá trị cuối) downto (giá trị đầu) do (hành động) end for
 - Cấu trúc nhảy
 - goto nhãn x;





2.3. Một số thuật toán thông dụng

- Thuật toán kiểm tra số nguyên tố
- Thuật toán tìm USCLN, BSCNN của 2 số nguyên
- Thuật toán tìm phần tử lớn nhất trong một dãy
- Thuật toán sắp xếp
- Thuật toán tìm kiếm

14



Tìm phần tử lớn nhất trong một dãy hữu hạn số

- *Nhập:* dãy số a[1], a[2], a[3],... a[n]
- Xuất: max là giá trị lớn nhất trong dãy số đã cho
- Thuật toán:

max a[1]
for i = 2 to n do
 if max < a[i] then
 max a[i]
 end if
end for</pre>

Xuất: max là giá trị lớn nhất trong dãy số



2.4. Thuật toán đệ quy

- Có một số trường hợp, cách giải có thể vi phạm các tính chất của thuật toán nhưng lại khá đơn giản và được chấp nhận.
- Bài toán có thể được phân tích và đưa tới việc giải một bài toán cùng loại nhưng cấp độ thấp hơn.
- Ví du:
 - Định nghĩa giai thừa
 - 0! = 1
 n! = n*(n-1)! với n>0
 - Định nghĩa dấy số Fibonacci: 1, 1, 2, 3, 5, 8, 13,...
 - f₁ = 1, • f₂ = 1,
 - $f_n = f_{n-1} + f_{n-2}$

16



Thuật toán đệ quy (2)

- Thuật toán đệ quy tính giai thừa của 1 số tự nhiên:
 - Input: số tự nhiên n
 - Output: F(n) bằng n!
 - Thuật giải:
 - 1. F:=1
 - 2. if n>0 then F:=F(n-1)*n
 - 3. Output F

Thuật toán đệ quy (3)

- Thuật toán đệ quy tính số hạng thứ n của dãy số Fibonacci:
 - Input: số tự nhiên n
 - Output: F(n) bằng số hạng thứ n của dãy
 - Thuật giải:
 - 1. if n=1 or n=2 then F:=1
 - 2. if n>2 then F:=F(n-1)+F(n-2)
 - 3. Output F



Thuật toán đệ quy (4)

- Đặc điểm của thuật toán đệ quy:
 - Có 1 trường hợp cơ sở/trường hợp dùng
 - Có phần đệ quy bên trong thuật toán (nó gọi đến chính nó)
 - Có sự biến đổi tiến tới trường hợp cơ sở.

19



Bài tập

- Viết thuật toán tìm USCLN của hai số tự nhiên
- Viết thuật toán tìm BSCNN của hai số tự nhiên
- Viết thuật toán tìm phần tử lớn nhất trong một dãy số hữu hạn
- Viết thuật toán sắp xếp
- Viết thuật toán tìm kiếm

20



2.5. Thuật giải heuristic

- Thường tìm được lời giải tốt (những chưa chắc đã tốt nhất)
- Dễ dàng và nhanh chóng hơn so với giải thuật tối ưu
- Thể hiện một cách hành động khá tự nhiên, gần gũi với suy nghĩ và hành động của con người.

21



Thuật giải heuristic (2)

- Các nguyên lý
 - Nguyên lý vét cạn thông minh: trong bài toán tìm kiếm khi không gian tìm kiếm lớn => giới hạn không gian tìm kiếm hoặc thực hiện dò tìm đặc biệt dựa vào đặc thù của bài toán để nhanh chóng tìm ra mục tiêu
 - Nguyên lý tham lam: lấy tiêu chuẩn tối ưu toàn cục làm tiêu chuẩn chọn lựa hành động cục bộ của từng bước trong quá trình tìm kiếm lời giải
 - Nguyên lý thứ tự: thực hiện hành động theo thứ tự hợp lý của không gian khảo sát nhằm nhanh chóng đạt được một lời giải tốt.

22



Lời hay ý đẹp

Cuộc đời chi là một chuỗi những cơ hội ngẫu nhiên. Cái khó là nắm bất cơ hội mà hành động. Đừng bao giờ để vuột mất. Không phải ngày nào bạn cũng gặp cơ hội may đầu "

George Besnard-Shaw



Phần 3 - Lập trình bằng ngôn ngữ C

Nguyễn Hồng Phương

Email: nhuongnh-fit@mail.hut.edu.vn
Website: http://is.hut.edu.vn/~phuongnh
Bộ môn Các hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Mục tiêu

 Giúp sinh viên bước đầu học lập trình thông qua phương tiện là ngôn ngữ lập trình C

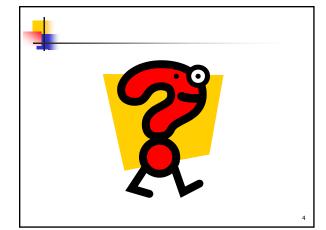
2



Nội dung phần 3

- Chương 1: Giới thiệu tổng quan về ngôn ngữ lập trình C
- Chương 2: Kiểu dữ liệu và biểu thức trong C
- Chương 3: Các cấu trúc lập trình trong C
- Chương 4: Con trỏ và mảng
- Chương 5: Xâu ký tự
- Chương 6: Hàm
- Chương 7: Cấu trúc
- Chương 8: Tệp







Lời hay ý đẹp

"The love we gave is the only love we keep" (Tình yêu cho đi là thứ tình yêu duy nhất ta còn giữ)



Chương 1: Tổng quan về ngôn ngữ C

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



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.



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 thế mạnh trong xử lý các dạng dữ liệu số, văn bản, cơ sở dữ liêu.
 - 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 lí 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



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.

8



1.2.1. Tập ký tự (tiếp)

26 chữ cái hoa:	A	ВС	٠.	. X	Y	Z					
26 chữ cái thường:	a	b c		ху	z						
10 chữ số:	0	1 2	3	4 5	6	7 8	9				
Các kí hiệu toán	+	-	*	/	=	<	>				
Các dấu ngăn cách:		;	,	:	sp	ace	t	ab			
Các dấu ngoặc:	()	[1	{	}					
Các kí hiệu đặc biệt:		?	\$	æ	#	^	\	!	١	"	^



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 đinh.
- 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	



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.



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

- Qui 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 (đấu cách – <i>space</i>) trong tên
int, char	trùng với từ khóa của ngôn ngữ C



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 du:

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



1.2.4. Các kiểu dữ liệu

- Là một tập hợp các giá trị mà một dữ liệu thuộc kiểu dữ liêu đó có thể nhân 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.
 Môt dữ liêu thuộc kiểu dữ liêu 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:
- _
- Công:Trừ:
- _
- Nhân:
- Chia lấy phần nguyên:
- Chia lấy phần dư:
- Chia lay phan du: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.



Biểu diễn hằng số nguyên

- Dang 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
 - 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

- Bằng ký hiệu của ký tự đó đặt giữa 2 dấu nháy đơn (").
- Bằng 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 '	1/11	39 hoặc 047 hoặc 0x27
Dấu nháy kép "	1/""	34 hoặc 042 hoặc 0x22
Dấu gạch chéo ngược \	1///	92 hoặc 0134 hoặc 0x5c
Kí tự xuống đòng	'\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",...



1.2.6. Biến (variable)

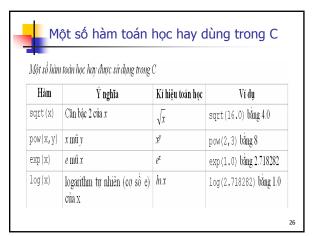
- Là đại lượng mà giá trị có thể thay đổi trong chương trình.
- Hằng và biến được sử dụng để lưu trữ dữ liệu, và phải thuộc một kiểu dữ liệu nào đó.
- Tên biến và hằng được đặt theo quy tắc đặt tên cho định danh.



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 bô.
- 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 (tiếp)

log10(x)	logarithm cơ số 10 của x	logx	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	tg x	tan(0.0) bằng 0.0
ceil(x)	phần nguyên giả của x, tức là số nguyên nhỏ nhất không nhỏ hơn x		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		floor(2.5) bằng 2 floor(-2.5) bằng -3



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 lênh
- Cuối mỗi câu lệnh đều có dấu chấm phẩy (;) để đánh dấu kết thúc câu lênh.

28



1.2.8. Câu lệnh (tiếp)

- Câu lệnh được chia thành 2 nhóm chính:
 - Nhóm các câu lênh 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 { }



1.2.9. Chú thích (Comment)

- Lời mô tả, giải thích vắn tắt cho một câu lệnh, một đoan 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: /*

*/





1.3. Cấu trúc cơ bản của một chương trình C

Gồm 6 phần có thứ tư như sau:

Phần1: 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 têp tiêu đề:
 - Thông báo cho chương trình dịch biết là chương trình có sử dụng những thư viên nào.
 - Ví du:

#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: Nôi dung của các hàm đã khai báo
 - Cài đặt (viết mã) cho các hàm đã khai báo nguyên mẫu ở phần 3.

35

4

Ví dụ một chương trình C đơn giản

```
/* Chuong trinh sau se nhap vao tu ban phim
  2 so nguyen va hien thi ra man hinh tong,
  hieu tich cua 2 so nguyen vua nhap vao */
#include <stdio.h>
#include <conio.h>
void main()
{
   // Khai bao cac bien trong chuong trinh
  int a, b;
  int tong, hieu, tich;
```

```
Ví dụ một chương trình C đơn giản (tiếp)

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

```
Ví dụ một chương trình C đơn giản (tiếp)

// 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);
// Doi nguoi dung an phim bat ki de ket thuc
getch();
}
```



1.4. Biên dịch chương trình viết bằng ngôn ngữ C

- Biên dịch chương trình viết bằng ngôn ngữ C (xem tài liệu phát)
 - 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



1.5. Bài tập

Bài tập 1:

Trong các định danh sau, định danh nào là không hợp lê:

- 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



1.5. Bài tập (tiếp)

Bài tập 3:

Cho biết biểu diễn dưới dạng số thực dấu phảy tĩnh của các hằng số thực sau:

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

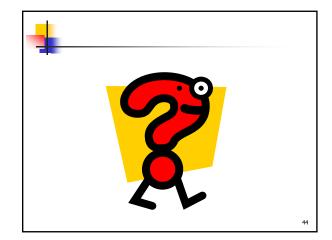
42



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() { }

/12





Lời hay ý đẹp

"Sau từ Yêu, Giúp đỡ là một từ đẹp nhất"



Chương 2: Kiểu dữ liệu và biểu thức trong C

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Nội dung chương này

- 2.1. Các kiểu dữ liệu chuẩn trong C
- 2.2. Khai báo và sử dụng biến, hằng
- 2.3. Các lệnh vào ra dữ liệu với các biến
- 2.4. Các lệnh vào ra khác
- 2.5. Các phép toán trong C
- 2.6. Biểu thức trong C
- 2.7. Một số toán tử đặc trưng

2



2.1. Các kiểu dữ liệu chuẩn trong C

Kiểu dữ liệu	Ý nghĩa	Kích thước	Miền biểu diễn
unsigned char	Kí tự	1 byte	0 255
char	Kí tự	1 byte	-128 +127
unsigned int	Số nguyên không dấu	2 byte	0 65,535
int	Số nguyên có dấu	2 byte	-32768 +32767

3



2.1. Các kiểu dữ liệu chuẩn trong C (tiếp)

Kiểu dữ liệu	Ý nghĩa	Kích thước	Miền biểu diễn
unsigned long	Số nguyên không dấu	4 bytes	0 ÷ 4,294,967,295
long	Số nguyên có dấu	4 bytes	-2,147,483,648 ÷ 2,147,483,647
float	Số thực dấu phẩy động, độ chính xác đơn	4 bytes	± 3.4E-38 ÷ ± 3.4E+38
double	Số thực dấu phẩy động độ chính xác kép	8 bytes	± 1.7E-308 ÷ ± 1.7E+308
long double	Số thực dấu phầy động	10 bytes	± 3.4E-4932 ÷ ± 1.1E+4932



2.2. Khai báo và sử dụng biến, hằng

- 2.2.1. Khai báo và sử dụng biến
- 2.2.2. Khai báo và sử dụng hằng



2.2.1. Khai báo và sử dụng biến

Cú pháp khai báo:

kiểu_dữ_liệu tên_biến;

kiểu_dữ_liệu tên_biến₁, ..., tên_biến_N;

 Ví dụ: Khai báo một biến x thuộc kiểu số nguyên 2 byte có dấu (int), biến y, z,t thuộc kiểu thực 4 byte (float) như sau:

int x;
float y,z,t;

6



Khai báo và khởi tạo giá trị cho biến

Cú pháp:

kiểu_dữ_liệu tên_biến = giá_tri_ban_đầu; Hoăc:

kiểu_dữ_liệu biến,=giá_trị, biến,=giá_trị,;

Ví du:

int a = 3;// sau lenh nay bien a se co gia tri bang 3 float x = 5.0, y = 2.6; // sau lenh nay x co gia // tri 5.0, y co gia tri 2.6



2.2.2. Khai báo hằng

- Cách 1: Dùng từ khóa #define:
 - Cú pháp:

define tên_hàng giá_tri

Ví du:

#define MAX_SINH_VIEN 50 #define CNTT "Cong nghe thong tin" #define DIEM CHUAN 23.5

8



2.2.2. Khai báo hằng (tiếp)

- Cách 2: Dùng từ khóa **CONS**†:
 - Cú pháp:

const kiểu dữ liệu tên hằng = giá tri;

■ Ví dụ:

const int MAX_SINH_VIEN = 50;
const char CNTT[20] = "Cong nghe thong tin";
const float DIEM_CHUAN = 23.5;

_



2.3. Các lệnh vào ra dữ liệu

- C cung cấp 2 hàm vào ra cơ bản:
 - printf()
 - scanf().
- Muốn sử dụng 2 hàm printf() và scanf() ta cần khai báo tệp tiêu đề stdio.h:

#include <stdio.h>

Hoăc

#include "stdio.h"

10



2.3.1. Hàm printf

- Muc đích:
 - Hiển thị ra màn hình các loại dữ liệu cơ bản như: Số, kí tư và xâu kí tư
 - Một số hiệu ứng hiển thị đặc biệt như xuống dòng, sang trang,...
- Cú pháp:

printf(xâu_định_dạng, [danh_sách_tham_số]);

- Xâu_định_dạng: Là xâu dùng để qui định cách thức hiển thị dữ liệu ra màn hình máy tính.
- danh sách tham số: Danh sách các biến sẽ được hiển thị giá trị lên màn hình theo cách thức được qui định trong xâu_định_dạng.

-

2.3.1. Hàm printf (tiếp)

- Trong xâu_định_dạng chứa:
 - Các kí tự thông thường: Được hiển thị ra màn hình.
 - Các nhóm kí tự định dạng: Xác định quy cách hiển thị các tham số trong phần danh_sách_tham_số.
 - Các kí tự điều khiển: Dùng để tạo các hiệu ứng hiển thị đặc biệt như xuống dòng ('\n') hay sang trang ('\f')...

12

```
Hàm printf - ví dụ:

Chương trình sau
#include <conio.h>
#include <stdio.h>
void main()
{ int a = 5;
    float x = 1.234;
    printf("Hien thi mot so nguyen %d và mot so thuc %f",a,x);
    getch();
}

Sẽ cho ra kết quả:
    Hien thi mot so nguyen 5 va mot so thuc 1.234000
```



Hàm printf

- Trong ví dụ trên:
 - "Hien thi mot so nguyen %d và mot so thuc %f" là xâu_định_dạng
 - a,x là danh_sách_tham_số
 - %d dùng để báo cho máy biết rằng cần phải hiển thị tham số kiểu nguyên (biến a)
 - %f dùng để báo cho máy cần hiển thị tham số tương ứng (biến x) theo định dạng số thực

14



Hàm printf

- Nhóm kí tự định dạng thứ k trong xâu dịnh dạng dùng để xác định quy cách hiển thị tham số thứ k trong danh sách tham số.
- Số lượng tham số trong Danh sách tham số bằng số lượng nhóm các kí tự định dạng trong xâu dịnh dạng.
 Trong ví dụ trên là 2.
- Mỗi nhóm kí tự định dạng chỉ dùng cho một kiểu dữ liệu
 Ví dụ: %d dùng cho kiểu nguyên
 %f dùng cho kiểu thực
- Nếu giữa nhóm kí tự định dạng và tham số tương ứng không phù hợp với nhau thì sẽ hiển thị ra kết quả không như ý.

15



Nhóm ký tự định dạng

Một số nhóm kí tự định dạng phổ biến:

Nhóm kí tự định dạng	Áp dụng cho kiểu dữ liệu	Ghí chú	
%d	int, long, char	Hiền thị tham số tương ứng dưới dạng số nguyên có dấu hệ đếm thập phân	
%i	int, long, char	Hiển thị tham số tương ứng dưới dạng số nguyên có dấu hệ đếm thập phân	
%0	int, long, char	Hiển thị tham số tương ứng dưới dạng số nguyên không dấu trong hệ đểm cơ số 8.	

16



Nhóm ký tự định dạng (tiếp)

%u	int, long, char	Hiển thị tham số tương ứng đười dạng số nguyên không dấu.
%x	int, long, char	Hiển thị tham số tương ứng đười dạng số nguyên hệ đếm 16 (không có 0x đứng trước), sử dụng các chữ cái a b c d e f
\$X	int, long, char	Hiển thị tham số tương ứng đười đạng số nguyên hệ đếm 16 (không có 0x đứng trước), sử dụng các chữ cái A B C D E F
вe	float, double	Hiển thị tham số tương ứng dưới dạng số thực dấu phẩy động
%f	float, double	Hiển thị tham số tương ứng dưới dạng số thực dấu phẩy tính

17



Nhóm ký tự định dạng (tiếp)

₩g	float, double	Hiển thị tham số tương ứng số thực dưới dạng ngắn gọn hơn trong 2 dạng dấu phẩy tính và dấu phẩy động
%C	int, long, char	Hiển thị tham số tương ứng dưới dạng kí tự
85	char * (xâu kí tự)	Hiển thị tham số tương ứng đười dạng xâu kí tự



Bổ sung về định dạng

- C cho phép đưa thêm một số thuộc tính định dạng dữ liệu khác vào trong xâu định dạng như:
 - Độ rộng để hiển thị (độ rộng tối thiểu)
 - Căn lè trái
 - Căn lè phải.

19



Độ rộng hiển thị

- Đối với số nguyên hoặc ký tự hoặc xâu ký tự:
 - Có dạng %md, với m là số nguyên không âm
 - Ví dụ: nếu số a = 1234

I ênh:

printf("%5d",a);//danh 5 cho de hien thi a
printf("\n%5d",34);

Cho ra kết quả:

0123400034

(I: kí hiệu cho dấu cách đơn (space))

20



Độ rộng hiển thị (tiếp)

Đối với số nguyên hoặc ký tự hoặc xâu ký tự:
 Ví du:

printf("\n%3d %15s %3c", 1, "nguyen van a", 'g'); printf("\n%3d %15s %3c", 2, "tran van b", 'k');

Két quả:

001000nguyen van a00g 00200000tran van b00k

21



Độ rộng hiển thị (tiếp)

• Đối với số thực: m, n là 2 số nguyên không âm

%m.nf

Báo rằng cần dành:

- m vị trí để hiển thị số thực
- n vị trí trong m vị trí đó để hiển thị phần thập phân.

22



Độ rộng hiển thị (tiếp)

- Định dạng với dữ liệu là số thực (tiếp):
 - Ví dụ:

printf("\n%f",12.345);
printf("\n%.2f",12.345);
printf("\n%8.2f",12.345);

Két quả:

12.345000 12.35 00012.35

23



Độ rộng hiển thị (tiếp)

Khi số chỗ cần để hiển thị nội dung dữ liệu lớn hơn trong định dạng:

Tự động cung cấp thêm chỗ mới để hiển thị chứ không cắt bớt nội dung của dữ liệu.

■ Ví du: a=1000

printf("So a la: %1d", a);

Két quả:

So a la: 1000



Căn lề phải, căn lề trái

Căn lè phải:

Khi hiển thị dữ liệu, mặc định C căn lè phải

Căn lề trái:

Nếu muốn căn lề trái khi hiển thị dữ liệu ta chỉ cần thêm dấu trừ - vào ngay sau dấu %.

25



Căn lề phải, căn lề trái (tiếp)

Ví du:

printf("\n%-3d %-15s %4.2f %-3c",
 9, "nguyen van a", 8.5, 'g');
printf("\n%-3d %-15s %4.2f %-3c",
 10, "nguyen ha", 6.75, 'k');

Két quả:

9 nguyen van a 8.50g10 nguyen ha 6.75k

26



2.3.2. Hàm scanf

- Muc đích:
 - Hàm scanf() dùng để nhập dữ liệu từ bàn phím
- Cú pháp:

scanf(xâu_dinh_dang,[danh_sách_dia_chi]);

Ví dụ:

scanf("%d%f", &a, &b);

- Địa chỉ của một biến được viết bằng cách đặt dấu & trước tên biến.
 - Ví dụ:

Các biến có tên là a, x, ten_bien

Thì địa chỉ của chúng lần lượt sẽ là: $\&a, \&x, \&ten_bien$

27



Hàm scanf (tiếp)

- Xâu_định_dạng:
 - Gồm các ký tự được quy định cho từng loại dữ liệu được nhập vào.
 - Ví dụ: Với dữ liệu định nhập vào là kiểu nguyên thì xâu định dạng là: %d
- Danh_sách_địa_chỉ:
 - Bao gồm các địa chỉ của các biến, các địa chỉ này được phân tách nhau bởi dấu phẩy (,)
 - Danh_sách_địa_chỉ phải phù hợp với các nhóm kí tự định dạng trong xâu_định_dạng về:
 - Số lượng
 - Kiểu dữ liệu
 - Thứ tự

28



Nhóm ký tự định dạng

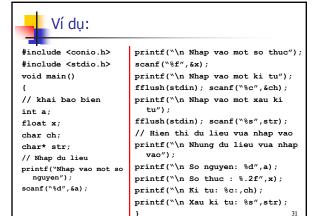
Nhóm kí tự định dạng	Ghi chú	
₹d	Định khuôn dạng dữ liệu nhập vào đượi dạng số nguyên kiểu int	
80	Định khuôn dạng dữ liệu nhập vào đười dạng số nguyên kiểu int hệ cơ số 8	
8x	Định khuôn dạng dữ liệu nhập vào đười dạng số nguyên kiểu int hệ cơ số 16	

29



Nhóm ký tự định dạng (tiếp)

8€	Định khuôn dạng dữ liệu nhập vào dưới dạng kí tự kiểu char
%S	Định khuôn dạng dữ liệu nhập vào dưới dạng xâu kí tự
%f	Định khuôn dạng dữ liệu nhập vào dưới dạng số thực kiểu float
%ld	Định khuôn dạng dữ liệu nhập vào dưới dạng số nguyên kiểu long
%lf	Định khuôn dạng dữ liệu nhập vào dưới dạng số thực kiểu double





Kết quả:

Nhap vao mot so nguyen: 2007 Nhap vao mot so thuc: 18.1625

Nhap vao mot ki tu: b

Nhap vao mot xau ki tu: ngon ngu lap trinh C

Nhung du lieu vua nhap vao

So nguyen: 2007 So thuc: 18.16 Ki tu: b

Xau ki tu: ngon

32



Một số quy tắc cần lưu ý

- Quy tắc 1: Khi đọc số
 - Hàm scanf() quan niệm rằng mọi kí tự số, dấu chấm ('.') đều là kí tự hợp lệ.
 - Khi gặp các dấu phân cách như tab, xuống dòng hay dấu cách (space bar) thì scanf() sẽ hiểu là kết thúc nhập dữ liệu cho một số
- Quy tắc 2: Khi đọc ký tự:

Hàm scanf() cho rằng mọi kí tự có trong bộ đệm của thiết bị vào chuẩn đều là hợp lệ, kể cả các kí tự tab, xuống dòng hay dấu cách.

33



Một số quy tắc cần lưu ý (tiếp)

- Quy tắc 3: Khi đọc xâu kí tự:
 - Hàm scanf() nếu gặp các kí tự dấu trắng, dấu tab hay dấu xuống dòng thì nó sẽ hiểu là kết thúc nhập dữ liệu cho một xâu kí tự.
 - Trước khi nhập dữ liệu kí tự hay xâu kí tự ta nên dùng lệnh fflush(stdin) để xóa bộ đêm.

34



2.4. Các lệnh vào ra khác

■ Hàm gets():

Dùng để nhập vào từ bàn phím một xâu kí tự bao gồm cả dấu cách, điều mà hàm scanf() không làm được.

Cú pháp :

gets (xâu_kí_tự);

Ví dụ:

```
char* str;
puts("Nhap vao mot xau ki tu:");
fflush(stdin); gets(str);
```

35



Các lệnh vào ra khác (tiếp)

Hàm puts():

Hiển thị ra màn hình nội dung xâu_kí_tự và sau đó đưa con trỏ xuống dòng mới.

Cú pháp:

puts (xâu_kí_t\u00a4);

Ví du:

puts("Nhap vao xau ki tu:");

 Tương đương với lệnh: printf("%s\n","Nhap vao xau ki tu:").



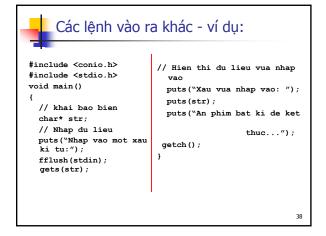
Các lệnh vào ra khác (tiếp)

- Hàm getch(): thường dùng để chờ người sử dụng ấn một phím bất kì rồi sẽ kết thúc chương trình.
- Cú pháp

getch();

 Để sử dụng các hàm gets(), puts(), getch() ta cần khai báo tệp tiêu đề conio.h.

37





Các lệnh vào ra khác - kết quả:

Nhap vao mot xau ki tu:
ngon ngu lap trinh C
Xau vua nhap vao:
ngon ngu lap trinh C
An phim bat ki de ket thuc ...

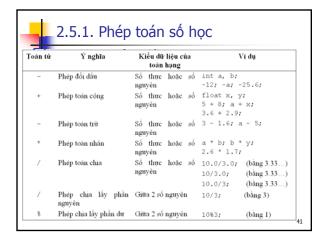
39



2.5. Các phép toán trong C

- Bao gồm:
 - Nhóm các phép toán số học
 - Nhóm các phép toán thao tác trên bit
 - Nhóm các phép toán quan hệ
 - Nhóm các phép toán logic.
 - Ngoài ra C còn cung cấp một số phép toán khác nữa như phép gán, phép lấy địa chỉ...

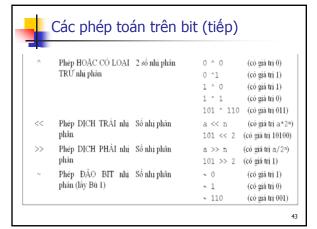
40

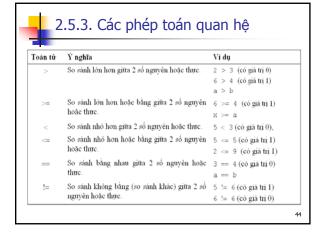


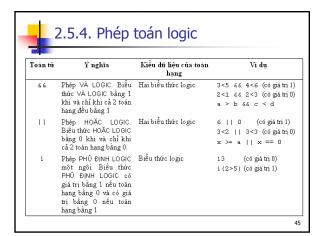


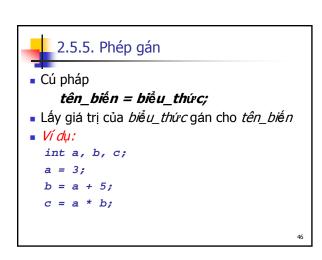
2.5.2. Các phép toán trên bit

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	'	'í dụ
6	Phép VÀ nhị phân	2 số nhị phân	0 & 0	(có giá trị 0)
			0 & 1	(có giá trị 0)
			1 & 0	(có giá trị 0)
			1 & 1	(có giá trị 1)
			101 & 110	(có giá trị 100)
1	Phép HOÁC nhị phân	2 số nhị phân	0 0	(có giá trị 0)
			0 1	(có giá trị 0)
			1 0	(có giá trị 0)
			1 1	(có giá trị 1)
			101 110	(có giá trị 111)











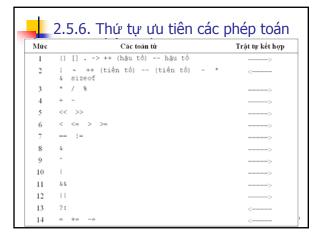
- Biểu thức gán là biểu thức nên nó cũng có giá trị.
- Giá trị của biểu thức gán bằng giá trị của biểu_thức:
 - ightarrow Có thể gán giá trị của biểu thức gán cho một biến khác hoặc sử dụng như một biểu thức bình thường
- Ví dụ:

```
int a, b, c;
a = b = 2007;
c = (a = 20) * (b = 30);
```

47

Phép gán (tiếp)

- Phép toán gán thu gọn:
 - $\mathbf{x} = \mathbf{x} + \mathbf{y}$; giống như $\mathbf{x} += \mathbf{y}$;
- Dạng lệnh gán thu gọn này còn áp dụng được với các phép toán khác: +, -, *, /, %, >>, <<, &, |, ^





Nguyên tắc xác định trật tự thực hiện các phép toán

- Biểu thức con trong ngoặc được tính toán trước các phép toán khác
- Phép toán một ngôi đứng bên trái toán hạng được kết hợp với toán hạng đi liền nó.
- Nếu toán hạng đứng cạnh hai toán tử thì có 2 khả năng là
 - Nếu hai toán tử có độ ưu tiên khác nhau thì toán tử nào có độ ưu tiên cao hơn sẽ kết hợp với toán hạng
 - Nếu hai toán tử cùng độ ưu tiên thì dựa vào trật tự kết hợp của các toán tử để xác định toán tử được kết hợp với toán hạng.

50



2.6. Biểu thức trong C

- 2.6.1. Các loai biểu thức
- 2.6.2. Sử dụng biểu thức

51

53



2.6.1. Các loại biểu thức

- Biểu thức số học:
 - Là biểu thức mà giá trị của nó là cái đại lượng số học (số nguyên, số thực).
 - Các toán tử là các phép toán số học (cộng, trừ, nhân, chia...), các toán hạng là các đại lượng số học (số, biến, hằng).
 - Ví du:
 - a, b, c là các biến thuộc một kiểu dữ liệu số nào đó.
 - · 3 * 3.7
 - 8 + 6/3
 - a + b c...

52



Các loại biểu thức (tiếp)

- Biểu thức logic:
 - Là biểu thức mà giá trị của nó là các giá trị logic, tức là một trong hai giá trị: Đúng (TRUE) hoặc Sai (FALSE).
 - Giá trị nguyên khác 0: Đúng (TRUE),
 - Giá trị 0: Sai (FALSE).
 - Các phép toán logic gồm có
 - AND: VÀ logic, kí hiệu là &&
 - OR: HOẶC logic, kí hiệu là 11
 - NOT: PHỦ ĐỊNH, kí hiệu là !

4

Các loại biểu thức (tiếp)

- Biểu thức quan hệ:
 - Là những biểu thức trong đó có sử dụng các toán tử quan hệ so sánh như lớn hơn, nhỏ hơn, bằng nhau, khác nhau...
 - Chỉ có thể nhận giá trị là một trong 2 giá trị Đúng (TRUE) hoặc Sai (FALSE)
 - → Biểu thức quan hệ là một trường hợp riêng của biểu thức logic.



Các loại biểu thức (tiếp)

- Ví du về biểu thức quan hê:
- // có giá trị logic là sai, FALSE
- // có giá trị logic là đúng, TRUE
- // có giá trị logic là đúng, TRUE
- // giả sử a, b là 2 biến kiểu int
- a+1 > a
- // có giá trị đúng, TRUE



Các loại biểu thức (tiếp)

- Ví dụ về biểu thức logic:
- (5 > 7) && (9!=10)
- // có giá trị logic là sai, FALSE
- // có giá trị logic là đúng, TRUE
- (5 > 7) | | (9!=10)
- // có giá trị logic là đúng, TRUE
- // có giá trị logic là sai, FALSE
- // có giá trị logic là đúng, TRUE
- // phủ định của 3, có giá trị logic là sai, FALSE

// phủ định của 0, có giá trị logic là đúng, TRUE

- // Có giá trị sai, FALSE. Giả sử a, b là 2 biến kiểu



2.6.2. Sử dụng biểu thức

- Làm vế phải của lệnh gán.
- Làm toán hạng trong các biểu thức khác.
- Làm tham số thực trong lời gọi hàm.
- Làm chỉ số trong các cấu trúc lặp for, while, do while.
- Làm biểu thức kiểm tra trong các cấu trúc rẽ nhánh if, switch.



2.7. Một số toán tử đặc trưng

- 2.7.1. Các phép toán tăng giảm một đơn vị
- 2.7.2. Phép toán lấy địa chỉ biến
- 2.7.3. Phép toán chuyển đổi kiểu bắt buộc
- 2.7.4. Biểu thức điều kiện



2.7.1. Các phép toán tăng giảm một đơn vị

- Tăng hoặc giảm một đơn vị cho biến:
 - <tên biến> = <tên biến> + 1;
 - → <tên biến>++
 - <tên biến> = <tên biến> 1;
 - → <tên biến>--
 - Ví du:

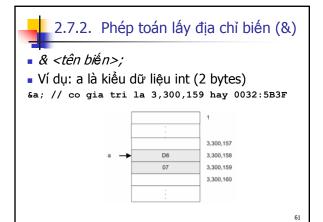
```
int a = 5;
float x = 10;
a ++; // turong durong với <math>a = a + 1;
x --; // turong durong với <math>x = x - 1;
```



Các phép toán tăng giảm một đơn vị (tiếp)

- Tiền tố: Thay đổi giá trị của biến trước khi sử
- Hậu tố: Tính toán giá trị của biểu thức bằng giá trị ban đầu của biến, sau đó mới thay đổi giá trị của biến
- Ví du:

```
int a, b, c;
        // a bang 3
a = 3;
b = a++;// Dang hau to
        // b bằng 3; a bằng 4
c = ++b;// Dang tien to
        // b bằng 4, c bằng 4;
```





2.7.3. Phép toán chuyển đổi kiểu bắt buộc

- (<kiểu dữ liệu mới>) <biểu thức>;
- Chương trình dịch sẽ tự động chuyển đổi kiểu
 - Số nguyên int → long int
 - Số long int → float...
- Ngược lại
 - Số nguyên long int 50,000 không phải là một số nguyên kiểu int vì phạm vi biểu diễn của kiểu int là từ (-32,768 đển 32,767).
 - → Phải ép kiểu

62



Phép toán chuyển đổi kiểu bắt buộc (tiếp)

vi du:
 #include <stdio.h>
 #include <conio.h>
void main()
{
 long int li; int i; float f; clrscr();
 li = 0x12345678; f = 123.456;
 i = (int) li;
 printf("\n li = %ld",li);
 printf("\n i = %d",i); i = (int) f;
 printf("\n f = %f",f);
 printf("\n i = %d",i);
 getch();
}



Phép toán chuyển đổi kiểu bắt buộc (tiếp)

- Két quả
 - li = 1193046
 - i = 13398
 - f = 123.456001
 - i = 123
- C hỗ trợ chuyển kiểu tự động trong những trường hợp sau
- char int long int float
 double long double

64



2.7.4. Biểu thức điều kiện

- Cú pháp
 - biểu_thức_1 ? biểu_thức_2 : biểu_thức_3
 - Giá trị của biểu thức điều kiện
 - Giá trị của biểu_thức_2 nếu biểu_thức_1 có giá trị khác 0 (tương ứng với giá trị logic ĐÚNG),
 - Ngược lại: Giá trị của biểu_thức_3 nếu biểu_thức_1 có giá trị bằng 0 (tương ứng với giá trị logic SAI).
- Ví dụ:

```
float x, y, z; // khai báo biến x = 3.8; y = 2.6; // gán giá trị cho các biến x, y z = (x < y)? x : y; // z số có giá trị bằng giá trị // nhỏ nhất trong 2 số x và y
```



63

Bài tập

- Bài 1
 - Viết chương trình tính chu vi và diện tích hình tròn theo giá trị bán kính r nhập từ bàn phím.
- Bài 2
 - Nhập phần thực x và phần ảo y của một số phức, sau đó in ra theo mẫu: (x,y) với các yêu cầu số in ra có 2 chữ số sau dấu chấm thập phân.



■ Bài 3

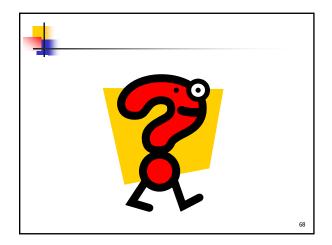
 Tính giá trị của biểu thức sau và đưa ra giá trị của biến a, b, c sau khi thực hiện phép toán:

 với a,b,c là các số nguyên nhập vào từ bàn phím

■ Bài 4

 Dùng biểu thức điều kiện ? : để đưa ra số lớn nhất trong 3 số thực nhập vào từ bàn phím







Lời hay ý đẹp

"Tình bạn chân thực là cây sồi và dây thường xuân.Đứng cùng đứng,ngã cùng ngã"



Lời giải bài tập chương 2: Kiểu dữ liệu và biểu thức trong C

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Bài 1

- Viết chương trình tính chu vi và diện tích hình tròn theo giá trị bán kính r nhập từ bàn phím.
- Công thức toán học:
 - Chu vi đường tròn: cv = 2 x x r
 - Diện tích hình tròn: dt = x r²
- Lời giải

2

Bài 1 (tiếp)

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    float r, cv, dt; // Khai bao 3 bien thuc
    printf("\n Nhap vao ban kinh r = ");
    scanf("%f",&r);
    cv = 2*M_PI*r; // Tinh chu vi
    dt = M_PI*r*r; // Tinh dien tich
    printf("\nChu vi = %10.2f\nDien tich = %10.2f",cv, dt);
    getch();
}
```



Bài 2

Nhập phần thực x và phần ảo y của một số phức, sau đó in ra theo mẫu: (x,y) với các yêu cầu số in ra có 2 chữ số sau dấu chấm thập phân.

4

Bài 2 (tiếp)

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float x, y; // Khai bao 2 bien thuc
    printf("\nNhap phan thuc = "); scanf("%f",&x);
    printf("\nNhap phan ao = "); scanf("%f",&y);
    printf("\n(%0.2f,%0.2f)",x,y);
    getch();
}
```



Bài 3

- Tính giá trị của biểu thức sau và đưa ra giá trị của biến a, b, c sau khi thực hiện phép toán:
 - a++ + ++a
 - --a b-- * ++c
- với a,b,c là các số nguyên nhập vào từ bàn phím

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a, b, c, z;
    printf("Nhap vao a = ");scanf("%d",&a);
    printf("Nhap vao b = ");scanf("%d",&b);
    printf("Nhap vao c = ");scanf("%d",&c);
    z = a++ + ++a;
    printf("NBieu thuc 1 = %d, a = %d",z,a);
    z = --a - b-- * ++c;
    printf("NBieu thuc 2 = %d, a = %d, b = %d, c = %d",z,a,b,c);
    getch();
}
```

```
Bài 3 - kết quả

CC(Dev-Cpp)Project3.exe

Nhap vao a = 4

Nhap vao b = 5

Nhap vao c = 6

Bieu thuc 1 = 10, a = 6

Bieu thuc 2 = -30, a = 5, b = 4, c = 7

Press any key to continue . . . _
```

Bài 4

 Dùng biểu thức điều kiện ? : để đưa ra số lớn nhất trong 3 số thực nhập vào từ bàn phím

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float a, b, c, z;
    printf("\nNhap vao so a = ");scanf("%f",&a);
    printf("\nNhap vao so b = ");scanf("%f",&b);
    printf("\nNhap vao so c = ");scanf("%f",&c);
    z = (a>b)?a:b;
    z = (z>c)?z:c;
    printf("\nSo lon nhat = %0.2f\n",z);
    getch();
}
```

```
Bài 4 - Kết quả

C:\Dev-Cpp\Project3.exe

Nhap vao so a = 3.45

Nhap vao so b = 6.32

Nhap vao so c = 1.2

So lon nhat = 6.32

Press any key to continue . . .
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
float a, b, c, z;
printf("\nNhap vao so a = ");scanf("%f",&a);
printf("\nNhap vao so b = ");scanf("%f",&b);
printf("\nNhap vao so c = ");scanf("%f",&c);
z = (a>b)?a:b;
z = (z>c)?z:c;
printf("\nSo lon nhat = %0.2f\n",z);
getch();
}
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
  float a, b, c, z;
  printf("\nNhap vao so a = ");scanf("%f",&a);
  printf("\nNhap vao so b = ");scanf("%f",&b);
  printf("\nNhap vao so c = ");scanf("%f",&c);
  z = (a>b)?(a>c?a:c):(b>c?b:c);
  printf("\nSo lon nhat = %0.2f\n",z);
  getch();
}
```

```
Bài 4 - kết quả (tiếp)

C\Dev-Cpp\Project3.exe

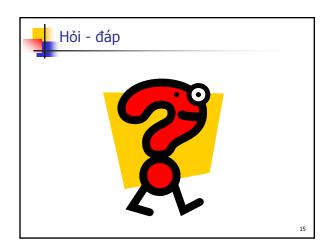
Nhap vao so a = 4.5

Nhap vao so b = 8.9

Nhap vao so c = 7.6

So lon nhat = 8.98

Press any key to continue . . . _
```





Nguyễn Hồng Phương

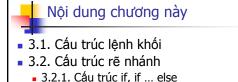
Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



■ 3.2.2. Cấu trúc lựa chọn switch

3.3. Cấu trúc lặp

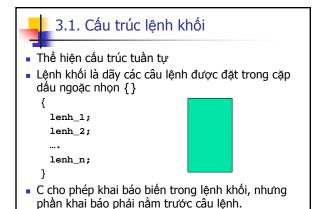
■ 3.3.1. Vòng lặp for

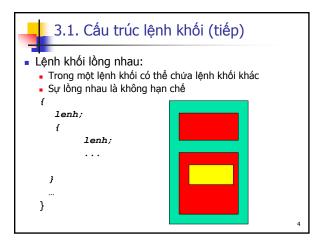
■ 3.3.2. Vòng lặp while

3.4. Các lệnh thay đổi cấu trúc lập trình

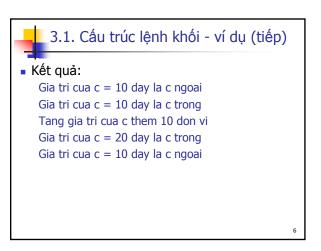
■ 3.4.1. continue

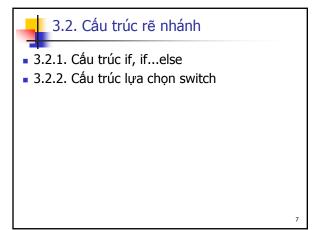
■ 3.4.2. break





```
3.1. Cấu trúc lệnh khối - ví du
#include <comio.h>
#include <stdio.h>
void main()
//Noi dung cua ham main() cung
la mot khoi lenh
                                            printf("\n Gia tri cua c = %d
  day la c trong",c);
{
                                             printf("\n Tang gia tri cua c
   them 10 don vi");
   c = c + 10;
   // khai bao bien
  int c:
                                                printf("\n Gia tri cua c
= %d day la c trong",c);
  c = 10;
  printf("Gia tri cua c = %d
day la c ngoai",c);
                                               printf("\n Gia tri cua c =
%d day la c ngoai",c);
getch();
// bat dau mot khoi lenh khac
                                             }// ket thuc khoi lenh cua ham
  main()
      int c;
      c = 10;
```

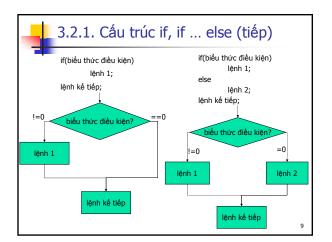


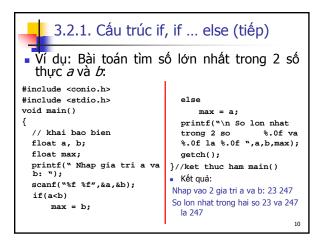


```
3.2.1. Cấu trúc if, if ... else

• Cú pháp cấu trúc if
    if (bieu_thuc_dieu_kien)
        lenh;

• Cú pháp cấu trúc if ... else
    if (bieu_thuc_dieu_kien)
        lenh_1;
    else
        lenh_2;
```



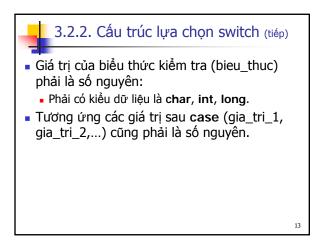


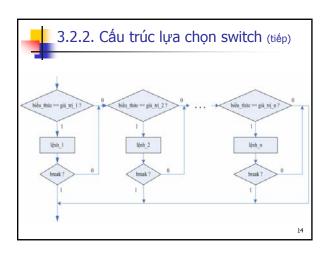
3.2.1. Cấu trúc if, if ... else - Luyện tập

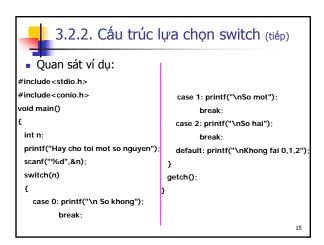
- Lập chương trình tìm số nhỏ nhất trong ba số thực nhập vào từ bàn phím.
- Viết chương trình nhập vào một số nguyên từ bàn phím, nếu là số chẵn thì hiển thị "Ban vua nhap so chan", nếu là số lẻ thì hiển thị "Ban vua nhap so le".
- Nhập vào 3 số thực, kiểm tra xem nó có tạo thành 3 cạnh của tam giác không?

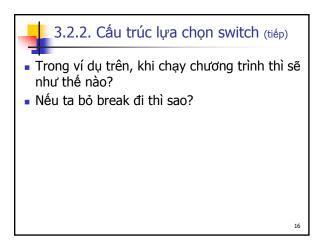
3.2.2. Cấu trúc lựa chọn switch

• Cú pháp cấu trúc switch
switch (bieu_thuc)
{
 case gia_tri_1: lenh_1; [break];
 case gia_tri_2: lenh_2; [break];
 ...
 case gia_tri_n: lenh_n; [break];
 [default: lenh_n+1; [break];]
}

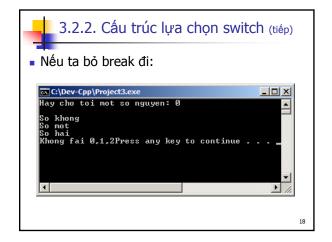














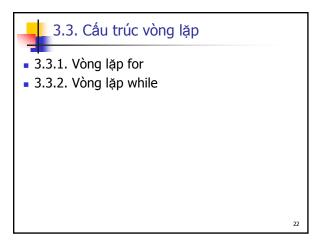
3.2.2. Cấu trúc lựa chọn switch-Luyện tập

- Trong một năm các tháng có 30 ngày là 4, 6, 9, 11 còn các tháng có 31 ngày là 1, 3, 5, 7, 8, 10, 12. Riêng tháng hai có thể có 28 hoặc 29 ngày.
- Hãy viết chương trình nhập vào 1 tháng, sau đó đưa ra kết luận tháng đó có bao nhiều ngày.

19

```
3.2.2. Cấu trúc lưa chon switch (tiếp)
#include <conio.h>
#include <stdio.h>
                                            case 4:
                                                case 6
int main ()
                                                case 9:
                                                case 11:
int thang; clrscr();
                                                  printf("\n Thang %d
printf("\n Nhap vao thang trong nam
                                            co 30 ngay ",thang);
break;
scanf("%d",&thang);
switch(thang)
                                                case 2:
                                                 printf ("\ Thang 2 co
                                            28 hoac 29 ngay");
  case 1:
                                                 break:
  case 3:
                                                default :
                                                 printf("\n Khong co
 case 5:
                                            thang %d", thang);
  case 8:
                                                 break:
                                              getch();
   printf("\n Thang %d co 31 ngay
                                              return 0;
                                                                      20
```

3.2.2. Cấu trúc lựa chọn switch (tiếp) #include <conio.h> #include <stdio.h> int main () case 4: case 6: int thang; clrscr(); case 9: printf("\n Nhap vao thang trong case 11: printf("\n Thang %d co 30 ngay ",thang); nam "); scanf("%d",&thang); default : while(thang < 1 | | thang > 12); printf("\n Thang %d co 31 ngay ", thang); break; switch(thang) printf ("\ Thang 2 co 28 hoac 29 getch(); return 0: 21



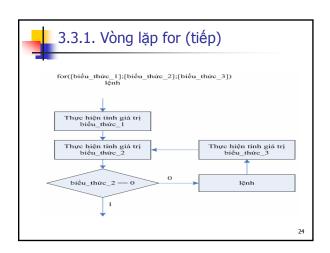
```
3.3.1. Vòng lặp for

Mục đích

Dùng để thực hiện lặp đi lặp lại một công việc nào đó với số lần lập xác định.

Cú pháp:
for (bieu_thuc_1; bieu_thuc_2; bieu_thuc_3)
{
   day_cac_lenh;
}

Trong đó:
   bieu_thuc_1: Khởi tạo giá trị ban đầu cho vòng lặp
   bieu_thuc_2: Điều kiện tiếp tục vòng lặp
   bieu_thuc_3: Thực hiện bước tăng của vòng lặp
   Chú ý các biểu thức 1, 2, 3 có thể có hoặc không
```



```
3.3.1. Vòng lặp for - ví dụ

• Đưa ra màn hình các số nguyên lẻ nhỏ hơn 100

#include <stdio.h>
#include <conio.h>
void main()
{
    int i;
    for(i = 1;i<100;i++)//so vong lap la 99
    {
        if(i%2 == 1) printf("%5d",i);
        if((i+1)%20 ==0) printf("\n");
    }
    getch();
}
```

```
3.3.1. Vòng lặp for - ví dụ (tiếp)

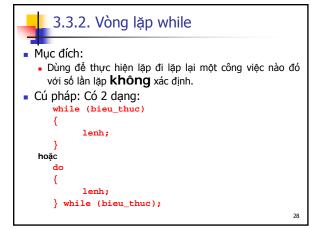
• Đưa ra màn hình các số nguyên lẻ nhỏ hơn 100

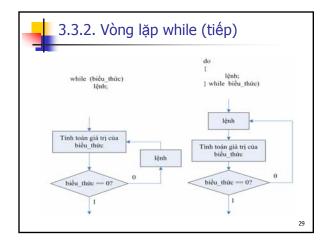
#include <stdio.h>
#include <conio.h>
void main()
{
    int i;
    for(i = 1;i<100;i+=2)//so vong lap la 50
    {
        printf("%5d",i);
        if((i+1)%20 ==0) printf("\n");
    }
    getch();
}
```

```
3.3.1. Vòng lặp for - ví dụ (tiếp)

• Kết quả:

1  3  5  7  9 11 13 15 17 19
21 23 25 27 29 31 33 35 37 39
41 43 45 47 49 51 53 55 57 59
61 63 65 67 69 71 73 75 77 79
81 83 85 87 89 91 93 95 97 99
```







3.3.2. Vòng lặp while (tiếp)



3.3.2. Vòng lặp while (tiếp)

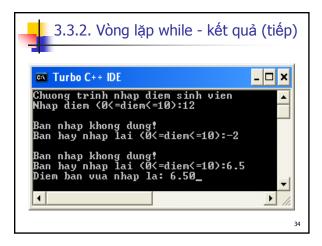
- Ví dụ: Nhập vào điểm của một sinh viên, nếu điểm đó không [0, 10] thì thông báo cho người dùng nhập lại.
- · Cách làm:
 - Nếu dùng lệnh if → Chỉ kiểm tra được 1 lần
 - Không dùng for được vì chưa biết trước số lần lặn
 - → Sử dụng vòng lặp while

31

```
#include <stdio.h>
#include <conio.h>
void main()

{
float diem; clrscr();
    printf("Chuong trinh nhap diem sinh vien\n");
    printf("Nhap diem (0<=diem<=10):");
    scanf("%f",&diem);
    while (diem < 0 | | diem > 10)
    {
        printf("\nBan nhap khong dung!\n");
        printf("\nBan nhap khong dung!\n");
        printf("\nBan nhap khong dung!\n");
        printf("\nDiem ban vua nhap la: %.2f", diem);
        getch();
    }
```

```
#include <stdio.h>
#include <conio.h>
#include <conio.h>
void main()
{
    float diem; clrscr();
    printf("Chuong trinh nhap diem sinh vien\n");
    do
    {
        printf("Nhap diem (0<=diem<=10):");
        scanf("%f",&diem);
        if (diem < 0 | | diem > 10)
            printf("\nBan nhap khong dung!\n");
    }
    while (diem < 0 | | diem > 10);
    printf("\nDiem ban vua nhap la: %.2f", diem);
    getch();
}
```





3.4. Các lệnh thay đổi cấu trúc lập trình

- continue
- break

3.4. Các lệnh thay đổi cấu trúc lập trình

■ Đối với các lệnh lặp:

■ while, do{...} while, hoặc for

■ Thay đổi việc thực hiện lệnh trong vòng lặp

→ C cung cấp 2 lệnh:

■ continue;

■ break;



3.4. Các lệnh thay đổi cấu trúc lập trình (tiếp)

continue

- Bỏ qua việc thực hiện các câu lệnh nằm sau lệnh continue trong thân vòng lặp.
- Chuyển sang thực hiện một vòng lặp mới

break

 Thoát khỏi vòng lặp ngay cả khi biểu thức điều kiện của vòng lặp vẫn còn được thỏa mãn.

37

```
3.4. Các lệnh thay đổi cấu trúc lập trình (tiếp)

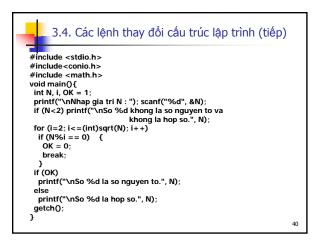
• Ví dụ:
#include <stdio.h>
#include <conio.h>
void main()
{
    int i;
    for(i = 1;i<=10;i++)
    {
        if(i == 5) continue;
        printf(%5d",i);
        if(i==7) break;
    }
    getch();
}
```



3.4. Các lệnh thay đổi cấu trúc lập trình (tiếp)

Ví dụ: Nhập vào 1 số nguyên. Kết luận số đó là số nguyên tố hay là hợp số?

39

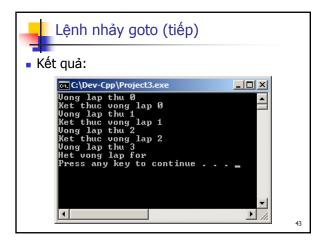




Lệnh nhảy goto

- Lệnh này sẽ chuyển điều khiển chương trình tới một nơi nào đó có gán một nhãn để đánh dấu.
- Nói chung, người ta khuyến cáo không nên dùng lệnh nhảy goto.
- Quan sát ví dụ sau:

```
#include<stdio.h>
#include<conio.h>
void main()
{
for(int i=0;i<=10;i++)
{
    printf("Vong lap thu %d \n",i);
    if(i==3) goto thoat; // nhày tới chỗ có nhân thoat
    printf("Ket thuc vong lap %d\n",i);
}
thoat: printf("Het vong lap for\n");
getch();
}
```





Bài tập

- Bài 1: Lập chương trình giải phương trình bậc hai: ax² + bx + c = 0 với a, b, c nhập vào từ bàn phím.
- Bài 2: Viết chương trình nhập một số nguyên n từ bàn phím và đưa ra n! (theo 3 cách)
- Bài 3: Viết chương trình nhập một số nguyên dương n từ bàn phím, đưa ra tổng các số từ 1 đến n (theo 3 cách)

44



Bài tập (tiếp)

- Bài 4: Viết chương trình nhập một số nguyên dương n từ bàn phím, đưa ra tổng các số chẵn từ 1 đến n (theo 3 cách)
- Bài 5: Viết chương trình kiểm tra một số nguyên nhập từ bàn phím có phải là số nguyên tố hay không?
- Bài 6: Lâp trình tính tổng
 - $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$
 - với n là số nguyên dương nhập từ bàn phím

Bài tập (tiếp)

- Bài 7: Viết chương trình nhập vào 2 số nguyên dương từ bàn phím và đưa ra ước số chung lớn nhất của chúng.
- Bài 8: Viết chương trình nhập vào 2 số nguyên từ bàn phím và đưa ra bội số chung nhỏ nhất của chúng.

46





Lời hay ý đẹp

"Phẩm cách chân chính của người là ở trong cách họ sống chứ không ở cái họ có"

Blackie



Chương 4: Con trỏ và mảng

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1

Nội dung chương này

- 4.1. Con trỏ và địa chỉ
 - 4.1.1. Tổng quan về con trỏ
 - 4.1.2. Các phép toán làm việc với con trỏ
- 4.2. Mång
 - 4.2.1. Khái niệm mảng
 - 4.2.2. Khai báo và sử dụng mảng
 - 4.2.3. Các thao tác cơ bản làm việc trên mảng
- 4.3. Sử dụng con trỏ làm việc với mảng

2



Nội dung trình bày

4.1. Con trỏ và địa chỉ

- 4.1.1. Tổng quan về con trỏ
- 4.1.2. Các phép toán làm việc với con trỏ
- 4.2. Mång
 - 4.2.1. Khái niệm mảng
 - 4.2.2. Khai báo và sử dụng mảng
 - 4.2.3. Các thao tác cơ bản làm việc trên mảng
- 4.3. Sử dụng con trỏ làm việc với mảng



4.1. Con trỏ và địa chỉ

- 4.1.1.Tổng quan về con trỏ
- 4.1.2. Các phép toán làm việc với con trỏ

4





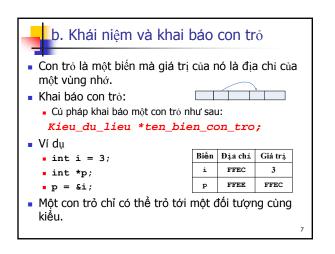
4.1.1. Tổng quan về con trỏ

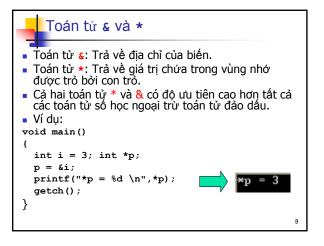
- a. Địa chỉ và giá trị của một biến
 - Bộ nhớ như một dãy các byte nhớ.
 - Các byte nhớ được xác định một cách duy nhất qua một địa chi.
 - Biến được lưu trong bộ nhớ.
 - Khi khai báo một biến
 - Chương trình dịch sẽ cấp phát cho biến đó một số ô nhớ liên tiếp đủ để chứa nội dung của biến. Ví dụ một biến số nguyên (int) được cấp phát 2 byte.
 - Địa chi của một biến chính là địa chi của byte đầu tiên trong số đó.

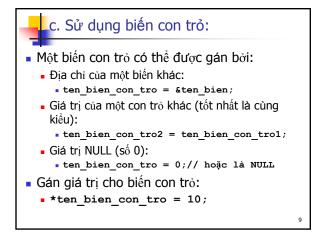
Địa chỉ và giá trị của một biến (tiếp)

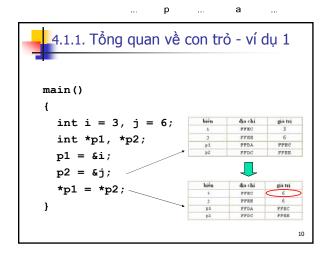
- Một biến luôn có hai đặc tính:
 - Địa chỉ của biến.
 - Giá trị của biến.
- Ví du:
 - int i, j;
 - = i = 3;
 - •j = i + 1;

Biến	Địa chỉ	Giá trị
i	FFEC	3
j	FFEE	4









```
d. Con trỏ void
void *ten_bien_con_tro;
Con trỏ đặc biệt, không có kiểu,
Có thể nhận giá trị là địa chỉ của một biến thuộc bất kỳ kiểu dữ liệu nào.
Ví dụ:

void *p, *q;
int x = 21;
float y = 34.34;
p = &x; q = &y;
```



4.1.2. Các phép toán làm việc với con trỏ

- Cộng/trừ con trỏ với một số nguyên (int, long)
 → Kết quả là một con trỏ cùng kiểu
 - ptr--; //ptr trỏ đến vị trí của phần tử đứng trước nó.
- Trừ hai con trỏ cho nhau
 - Kết quả là một số nguyên
 - Kết quả này nói lên khoảng cách (số phần từ thuộc kiểu dữ liệu của con trò) ở giữa hai con trò.
- Các phép toán: Cộng, nhân, chia, lấy số dư trên con trở là không hợp lệ.
- Ví dụ: (p2 trỏ đến số nguyên nằm ngay sau x trong bộ nhớ)

```
int x, *p1, *p2;
p1= &x;
p2= p1+1;
```

13



- 4.1. Con trỏ và địa chỉ
 - 4.1.1. Tổng quan về con trỏ
 - 4.1.2. Các phép toán làm việc với con trỏ

4.2. Mảng

- 4.2.1. Khái niệm mảng
- 4.2.2. Khai báo và sử dụng mảng
- 4.2.3. Các thao tác cơ bản làm việc trên mảng
- 4.3. Sử dụng con trỏ làm việc với mảng

4



4.2. Mảng

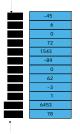
- 4.2.1. Khái niệm mảng
- 4.2.2. Khai báo và sử dụng mảng
- 4.2.3. Các thao tác cơ bản làm việc trên mảng

15



4.2.1. Khái niệm mảng

- Mảng là một tập hợp hữu hạn các phần tử có cùng kiểu dữ liệu được lưu trữ liên tiếp nhau trong bộ nhớ.
- Các phần tử trong mảng có cùng tên (và cũng là tên mảng) nhưng phân biệt với nhau ở chỉ số cho biết vị trí của chúng trong mảng



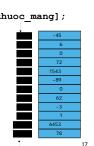
16



4.2.2. Khai báo và sử dụng mảng

a. Khai báo:

- Cú pháp:
 - Kieu_du_lieu ten_mang[kich_thuoc_mang];
- Ví dụ:
 - char c[12];
 - Phần tử đầu tiên có chỉ số 0
 - n phần tử của mảng có tên c:
 - c[0], c[1]...c[n 1]
 - int mang_nguyen[4];
 - float mang_thuc[6];





4.2.2. Khai báo và sử dụng mảng (tiếp)

a. Khai báo (tiếp):

- Mảng nhiều chiều:
 - Mỗi phần tử của mảng cũng là một mảng khác
 → Giống vector trong toán học.
 - Ví dụ:
 - Mảng 2 chiều: int a[6][5];
 - Mång 3 chiều: int b[3][4][5];



4.2.2. Khai báo và sử dụng mảng (tiếp)

- b. Sử dụng mảng:
 - Truy cập vào 1 phần tử của mảng thông qua tên mảng và chỉ số của phần tử đó.
 - Cú pháp: ten mang[chi so cua phan tu]
 - Ví dụ 1: int mang_nguyen[3];
 - mang nguyen[0]: Phần tử thứ 1.
 - mang nguyen[1]: Phần tử thứ 2.
 - mang nguyen[2]: Phần tử thứ 3.

19



4.2.2. Khai báo và sử dụng mảng (tiếp)

- b. Sử dụng mảng (tiếp):
 - Ví dụ 2: int a[6][5];
 - a[0] là phần tử đầu tiên của mảng, là 1 mảng
 - → Phần tử đầu tiên của mảng a[0] là a[0][0],...
 - ...
 - a[2][3] sẽ là phần tử thứ 4 của phần tử thứ 3 của
 - \rightarrow a[i][j] sẽ là phần tử thứ j+1 của a[i], mà phần tử a[i] lại là phần tử thứ i+1 của a.

20



4.2.3. Các thao tác cơ bản làm việc trên mảng

a. Nhập dữ liệu cho mảng:

- Nhập dữ liệu cho từng phần tử của mảng
- Ví dụ 1:

```
float a[10]; int i;
scanf("%f",&a[1]);
a[2] = a[1] + 5;
```

Ví du 2:

VI UU Z;
int b[10];
int i;
//Nhap gia tri tu ban phim cho tat ca cac phan tu mang b
for(i = 0; i < 10; i++)
//

printf("\n Nhap gia tri cho b[%d]", i);
scanf("%d", &b[i]);
}

Nhập dữ liệu cho mảng (tiếp)

 Trường hợp không biết mảng sẽ có bao nhiêu phần tử mà chỉ biết số phần tử tối đa có thể có của mảng. Ví du:

```
int a[100];//Khai bao mang, so phan tu toi da la 100
int n; // Bien luu giu so phan tu thuc su cua mang
int i;
printf("\n Cho biet so phan tu cua mang: ");
scanf("%d",&n);
for(i = 0; i < n; i++)
{
    printf("\n a[%d] = ", i);
    scanf("%d",&a[i]);
}</pre>
```

22

24



Nhập dữ liệu cho mảng (tiếp)

 Mảng có thể được khởi tạo giá trị ngay khi khai báo

```
int a[4] = {4, 9, 22, 16};
float b[3] = {40.5, 20.1, 100};
char c[5] = {'h', 'e', 'l', 'l', 'o'};
```

→ Câu lệnh thứ nhất có tác dụng tương đương với 4 lệnh gán:

```
a[0] = 4; a[1] = 9; a[2] = 22; a[3] = 16;
```

23

21

b. Xuất dữ liệu chứa trong mảng

```
#include <stdio.h>
#include <conio.h>
#define KT 6
void main()
{
   int a[KT];
   int i, k;
   //Nhap gia tri cho cac phan tu mang a tu ban phim
   for(i = 0; i < KT; i++)
   {
      printf("\n a[%d] = ", i);
      scanf("%d", &a[i]);
   }
   printf("\nBat dau hien thi gia tri cac phan tu\n");
   printf("\n a[3] = %d", a[3]);
   //Hien thi gia tri tat ca cac phan tu
   //Moi phan tu tren 1 dong
   for(i = 0; i < KT; i++)
      printf("\n%d", a[i]);</pre>
```

```
Kết quả

a[0] = -2
a[1] = 4
a[2] = 1
a[3] = -11
a[4] = 3
a[5] = 8

Bat dau hien thi gia tri cac phan tu
a[3] = -11
-2
4
1
1
1
3
8
-2
4
Cho biet gia tri cua k = 2
1
-2
4
1
3
8
26
```

c. Tìm các phần tử có giá trị lớn nhất, nhỏ nhất

- #define KT 100
- int a[KT];
- $\mathbf{max} = \mathbf{a}[0];$
- Dùng vòng lặp for so sánh max với phần tử có chỉ số từ 1, 2,..., n-1 của mảng a.

27

c. Tîm các phần tử có giá trị lớn nhất, nhỏ nhất

int a[100]; int i, n; int max;

printf("\n Cho biet so phan tu cua mang: ");

scanf("%d", &n);

for(i = 0; i < n; i++)

{
 printf("\n a[%d] = ",i);
 scanf("%d", &a[i]);

}

max = a[0];// Ban dau gia su phan tu lon nhat la a[0]

//Lan luot so sanh voi cac phan tu con lai trong mang
for(i = 1; i < n; i++)

if(max < a[i]) // gap phan tu co gia tri lon hon

max = a[i];// coi phan tu nay la phan tu lon nhat

 $printf("\n Phan tu lon nhat trong mang la: %d", max);$

28



4.2.3. Các thao tác cơ bản làm việc trên mảng (tiếp)

- Tìm kiếm trên mảng
- Sắp xếp mảng

29

Nội dung trình bày

- 4.1. Con trỏ và địa chỉ
 - 4.1.1. Tổng quan về con trỏ
 - 4.1.2. Các phép toán làm việc với con trỏ
- 4.2. Mång
 - 4.2.1. Khái niệm mảng
 - 4.2.2. Khai báo và sử dụng mảng
 - 4.2.3. Các thao tác cơ bản làm việc trên mảng
- 4.3. Sử dụng con trỏ làm việc với mảng



4.3.1. Con trỏ và mảng 1 chiều

Xét câu lệnh:

int a[10];

• Khai báo này hệ thống sẽ cấp cho mảng a 10 khối gồm 10 khối nhớ liên tiếp: a[0], a[1], a[2], a[3],, a[9](trong trường hợp này mỗi khối nhớ 2 byte).

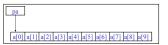
a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7] a[8] a[9]

31

4

4.3.1. Con trỏ và mảng 1 chiều (tiếp)

- Có một điều cần nói ở đây là tên mảng là một hằng địa chỉ, nó chính là địa chỉ của phần tử đầu tiên của mảng; điều này có nghĩa là:
- a tương đương với &a[0]
- Néu pa là một con trỏ kiểu nguyên, int *pa;
- Khi đó phép gán pa = &a[0];
- Đem pa trỏ đến phần tử thứ nhất (có chỉ số là 0) của a; nghĩa là pa chứa địa chỉ của a[0].



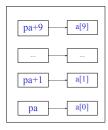
32

34



4.3.1. Con trỏ và mảng 1 chiều (tiếp)

 pa +i là địa chỉ của a[i] và *(pa+i) là nội dung của a[i]



33

#define N 5 int tab[N] = {1, 2, 6, 0, 7}; main() {

```
main()
{
   int i; int *p;
   p = tab;
   for (i = 0; i < N; i++)
   {
      printf(" %d \n",*p);
      p++;
   }
}</pre>
```

4

4.3.1. Con trỏ và mảng 1 chiều (tiếp)

- Tóm lại, một biểu thức chứa một mảng và một chỉ số tương đương với một cách viết khác sử dụng một con trỏ cùng một độ lệch.
- Cổ sự khác nhau giữa tên của mảng và con trỏ. Một con trỏ là một biến và vì thế, các câu lệnh pa=a và pa++ là hợp lệ. Nhưng một tên mảng không phải là một con trỏ, do đó các câu lệnh kiểu như a=pa;a++ là không hợp lệ. (Về thực chất, tên màng là một hằng con trỏ do đó chúng ta không thể thay đổi giá trị của nó được).
- Để minh hoạ cho các ý tưởng vừa trình bày ta xét một ví dụ sau: viết chương trình thực hiện các công việc sau:
 - Đọc từ bàn phím các phần tử của một mảng.
 - Tính tổng của chúng

35

4.3.1. Con trỏ và mảng 1 chiều (tiếp) #include <stdio.h> #include <stdio.h> main() main() float a[5],s; float a[5],s; for(i=0:i<5:i++) for(i=0:i<5:i++) printf("\na[%d] printf("\na[%d] = ",i); = ",i); scanf("%f",a+i); scanf("%f",&a[i]); s=0: for(i=0;i<5;i++) s=0; for(i=0;i<5;i++) s+=a[i]; s+=a[i]; printf("\n tong la: printf("\n tong la: %10.2f",s); %10.2f",s);

```
4.3.1. Con trỏ và mảng 1 chiều (tiếp)
#include <stdio.h>
                                          #include <stdio.h>
main()
                                          main()
  float a[5],s, *p;
                                            float a[5],s,*p;
  int i; p=a;
for(i=0;i<5;i++)
                                            int i; p=a;
for(i=0;i<5;i++)
                printf("\na[%d]
                                                           printf("\na[%d]
= ",i);
                                          = ",i);
                                                           scanf("%f",p+i);
        scanf("%f",&p[i]);
                                                   s=0;
                                            for(i=0;i<5;i++)

s+=*(p+i);

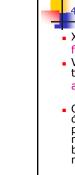
printf("\n tong la: %10.2f",s);
  for(i=0;i<5;i++)
s+=p[i];
  printf("\n tong la: %10.2f",s);
                                                                             37
```

4.3.2. Con trỏ và mảng nhiều chiều

- Phép toán lấy địa chỉ
- Phép cộng địa chỉ trong mảng hai chiều
- · Con trỏ và mảng hai chiều

38

4.3.2. Con trỏ.... - Phép toán lấy địa chỉ Phép toán lấy địa chỉ nói chung không được đối với các thành phân của màng nhiều chiều (trừ trường hợp màng hai chiều các số nguyển). Xét chương trình sau với ý định nhập số liệu cho ma trận thực. #include <stdio.h> main() { float a[10][20]; int i,j,n,m; printf("Nhap vao kich thuoc ma tran m,n="); scanf("%dšd", &m, &n); /*Chuong trình chậy dúng cho dến dẩy*/ for (i=0;j<m;i++) for (j=0;j<m;i++) { printf("a[%d][%d] = ",i,j); scanf("%f", &a[i][j]); } }





- Xét khai báo
- float a[2][3];
- Với khai báo này hệ thống cấp sáu phần tử liên tiếp trong bộ nhớ theo thứ tự sau:
 - a[0][0]a[0][1] a[0][2] a[1][0] a[1][1] a[1][2]
- Cũng như mảng một chiều tên của mảng hai chiều được hiểu như địa chỉ của phần tử đầu tiên của nó; phép cộng địa chỉ phải hiểu như sau: C cho rằng mảng hai chiều là mảng của mảng; như vậy với khai báo trên thi a là mảng mà mỗi phần tử của nó là một dãy gồm 3 số thực. Vì vậy
 - a trỏ tới đầu hàng thứ nhất (phần tử a[0][0]
 - a+1 trỏ tới đầu hàng thứ hai (phần tử a[1][0]

```
4.3.2. Con trỏ.... - Con trỏ và mảng 2 chiều

• Để truy xuất vào các phần tử của mảng hai chiều ta vẫn có thể dùng con trỏ theo cách sau: float *p, a[2][3]; p=(float*)a;

• Khi đó p trỏ tới a[0][0] p+1 trỏ tới a[0][1] p+2 trỏ tới a[0][2] p +3 trỏ tới a[1][0] p +4 trỏ tới a[1][1] p +5 trỏ tới a[1][2]
```



Con trỏ và mảng 2 chiều

- Các bạn để ý rằng, a là một hằng con trỏ trỏ đến các dòng của một ma trân hai chiều, vì vậy
 - a trỏ đến dòng thứ nhất a+1 trỏ đến dòng thứ hai
- Để tính toán được địa chỉ của phần tử ở dòng i cột j chúng ta phải dùng phép chuyển đổi kiểu bất buộc đối với a: (float *)a, đẩy là con trỏ trỏ đến thành phần a[0][0] của ma trận. Và vì vậy thành phần a[i][j] sẽ có địa chỉ là (float *a) +i*n+j (n là số cột).
- một cách tổng quát, nếu mảng có kiểu type và có kích thước các chiếu tương ứng là n1,n2,...,nk (Giả sử mảng a có k chiều). Địa chỉ cuả thành phần a[0]..[0] (k chỉ số 0) là (type *)a, và địa chỉ của a[i1][i2]...[ik] được tính như sau

$$(type^*)a + \sum_{j=1}^{k-1} i_j \prod_{l=j+1}^{k} n_l + i_k$$

43

4.3.3. Mảng các con trỏ

 Trước tiên chúng ta có thể khai báo một mảng các con trỏ bằng câu lệnh sau:

```
type *pointer_array[size];
• ví dụ câu lệnh,
   char *ma[10];
```

Sẽ khai báo một mảng 10 con trỏ char có thể được dùng để khai báo một mảng để lưu trữ địa chỉ của mười chuỗi ký tự nào đó.

Nếu các con trỏ được chuẩn bị để chỉ đến một biến nào đó đã có, thì như vậy, chúng ta có thể truy xuất được các biến này thông qua một mảng mà không cần đến vị trí thực sự của các biến đó có liên tiếp hay không.

44



4.3.3. Mảng các con trỏ

- Ví du:
 - Xem xét một mảng các con trỏ ptr_array được gán các địa chỉ của các biến int có giá trị và vị trí bất kỳ.
 - Chúng ta sẽ dùng một hàm để sắp xếp lại các địa chỉ này trong mảng để sao cho các địa chỉ của các số bé được xếp trước địa chỉ của các số lớn hơn. Lúc đó dù chúng ta không làm thay đổi vị trí hoặc thay đổi các giá trị của các biến nhưng mảng vẫn có vẻ như là một mảng chỉ đến các giá trị đã sắp xếp có thứ tự.

45



```
#include <stdio.h>
main() {
  int i,j,*x;
  int d=10,e=3,f=7;
  int a=12,b=2,c=6;
  int *ptr_array[6];
  /*Gán các thành phần của mảng*/
  ptr_array[0]=&a;
  ptr_array[1]=&b;
  ptr_array[2]=&c;
  ptr_array[3]=&d;
  ptr_array[4]=&e;
  ptr_array[5]=&f;
```

16

4.3.3. Mảng các con trỏ - ví dụ (tiếp) /*sắp xếp lại dãy số*/ for(i=0;i<5;i++) for(j=i+1;j<6;j++) if(*ptr_array[i]>*ptr_array[j]) { x=ptr_array[i]=ptr_array[j]; ptr_array[j]=x; } /*In kết quả sau khi sắp xếp*/ for(i=0;i<6;i++) printf(" %d \n",*ptr_array[i]); getch(); }

4.3.3. Mảng các con trỏ - ví dụ (tiếp) • Kết quả chạy chương trình

12



4.3.3. Mảng các con trỏ - nhận xét

 việc sử dụng một mảng các con trỏ có nhiều ý niệm gần giống như sử dụng một mảng hai chiều. Ví dụ, nếu các biến n và m được khai báo là:

int m[10][9]; int *n[10];

thì cách viết để truy xuất được các phần tử của các mảng này có thể tương tự nhau, chẳng hạn: m[6][5] và n[6][5] đều cho ta một kết quả là một số int.

Khai báo:

char slist[10][100];
char *plist[10];

đều khiến cho C hiểu rằng slist[i] và plist[i] là các con trỏ chỉ đến một đối tượng là char, hoặc là mảng char

49



4.3.3. Mảng các con trỏ - nhận xét

- Tuy vậy, giữa mảng nhiều chiều và mảng các con trỏ cũng tôn tại nhiều điểm khác nhau:
 - Mảng nhiều chiều thực sự là mảng có khai báo, do đó có chỗ đầy đủ cho tất cả các phần tử của nó. Còn mảng các con trổ chỉ mới có chỗ cho các biến con trổ mà thôi.
- Bên cạnh đó, việc sử dụng mảng các con trỏ có hai ưu điểm, đó là
 - Việc truy xuất đến các phần tử là truy xuất gián tiếp thông qua các con trỏ và như vậy, vị trí của các máng con này có thể là bất kỳ, và chúng có thể là những máng đã có bằng cách xin cấp phát chỗ động hay bằng khai báo biến mảng bình thường, tùy ý.
 - Các mảng con của nó được chỉ đến bởi các con trỏ, có thể có độ dài tùy ý, hoặc có thể không có (nếu con trỏ đó không được chuẩn bị, hoặc được gán bằng NULL).

50



4.3.4. Con trỏ trỏ đến con trỏ

■ Ví dụ

Khi đó *pp sẽ chỉ đến con trỏ đầu tiên của mảng, con trỏ này lại chỉ đến chuỗi "January" như đã khởi đầu cho mảng.

 Nếu tăng pp lên 1 thì khi đó pp sẽ chỉ đến phần từ kế tiếp của mảng có giá trị là con tró đến xâu ký tự "February".v.v.

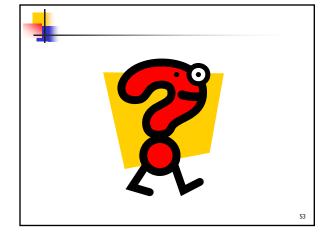
51



Bài tập

- Bài 1: Nhập vào một dãy số thực, sắp xếp tăng dần (sử dung mảng một chiều).
- Bài 2: Nhập vào một dãy gồm 6 số nguyên, sắp xếp giảm dần (không sử dụng mảng).

52





Lời hay ý đẹp

"Con người ta có ba điều lầm lỗi dễ mắc phải là: chưa đến lượt đã vội nói, điều đáng nói lại không nói và không nhìn vẻ mặt người khác mà đã nói"

Khổng Tử



Chương 5 Xâu ký tự

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Nội dung

- 5.1. Khái niệm xâu ký tự
- 5.2. Khai báo và sử dụng xâu
 - 5.2.1. Khai báo xâu ký tự
 - 5.2.2. Truy cập vào một phần tử của xâu
- 5.3. Một số hàm làm việc với ký tự và xâu ký tự trong C.
 - 5.3.1. Các hàm xử lý ký tự
 - 5.3.2. Các hàm xử lý xâu ký tự
 - 5.3.3. Một số hàm xử lý xâu ký tự khác
 - 5.3.4. Con trỏ và xâu ký tự

2



Nôi dung

⇒ 5.1. Khái niệm xâu ký tự

- 5.2. Khai báo và sử dụng xâu
 - 5.2.1. Khai báo xâu ký tự
 - 5.2.2. Truy cập vào một phần tử của xâu
- 5.3. Một số hàm làm việc với ký tự và xâu ký tự trong C.
 - 5.3.1. Các hàm xử lý ký tự
 - 5.3.2. Các hàm xử lý xâu ký tự
 - 5.3.3. Một số hàm xử lý xâu ký tự khác
 - 5.3.4. Con trỏ và xâu ký tự



5.1. Khái niệm xâu ký tự

- Là một dãy các kí tự viết liên tiếp nhau.
- Xâu rỗng: Xâu không gồm kí tự nào cả.
- Độ dài xâu: Số kí tự có trong xâu.
- Ví du:
 - "Tin hoc" là một xâu kí tự gồm 7 kí tự: 'T', 'i',
 'n', dấu cách (' '), 'h', 'o', và 'c'.
- Lưu trữ xâu ký tự:
 - Ký tự kết thúc xâu: NULL hoặc '\0'.



.



5.1. Khái niệm xâu ký tự (tiếp)

- Cần phân biệt giữa ký tự và xâu bao gồm một ký tự. Ví dụ 'A' là ký tự A được mã hóa bằng 1 byte, trong khi "A" là một xâu ký tự chứa ký tự A xâu này được mã hóa bằng 2 bytes cho ký tự A và ký tự "\0'.
- Trong C không tồn tại các phép toán so sánh, gán nội dung của xâu này cho xâu khác. Để thực hiện các công việc này C cung cấp cho người lập trình một thư viện các hàm chuẩn, được khai báo trong tệp header có tên là string.h.
- Để sử dụng các hàm thao tác xâu, trên đầu chương trình cần phải có dòng khai báo #include <string.h>



Nôi dung

5.1. Khái niệm xâu ký tự

5.2. Khai báo và sử dụng xâu

- 5.2.1. Khai báo xâu ký tự
- 5.2.2. Truy cập vào một phần tử của xâu
- 5.3. Một số hàm làm việc với ký tự và xâu ký tự trong C.
 - 5.3.1. Các hàm xử lý ký tự
 - 5.3.2. Các hàm xử lý xâu ký tự
 - 5.3.3. Một số hàm xử lý xâu ký tự khác
 - 5.3.4. Con trỏ và xâu ký tự



5.2.1. Khai báo xâu ký tự

- Cú pháp:
 - char ten xau[so ky tu toi da];
- Ví du:
 - char ho_va_ten[20];



5.2.2. Truy câp vào một phần tử của xâu

- Cú pháp:
 - ten_xau[chi_so_ky_tu_can_truy_cap]
- - char que_quan[10] = "Ha noi";

Phần tử thứ	Chi số của phần tử	Tên của phần tử	Nội dung lưu trữ
1	0	que_quan[0]	'H'
2	1	que_quan[1]	`a'
3	2	que_quan[2]	٠,
4	3	que_quan[3]	'N'
5	4	que quan[4]	`o'
6	5	que quan[5]	'i'
7	6	que quan[6]	'\0'
8	7	que quan[7]	
9	8	que_quan[8]	
10	9	que_quan[9]	



Nôi dung

- 5.1. Khái niệm xâu ký tự
- 5.2. Khai báo và sử dụng xâu
 - 5.2.1. Khai báo xâu ký tự
 - 5.2.2. Truy cập vào một phần tử của xâu
- 5.3. Một số hàm làm việc với ký tự và xâu ký tự trong C.
 - 5.3.1. Các hàm xử lý ký tự
 - 5.3.2. Các hàm xử lý xâu ký tự
 - 5.3.3. Một số hàm xử lý xâu ký tự khác
 - 5.3.4. Con trỏ và xâu ký tự



5.3.1. Các hàm xử lý ký tự

- Để sử dụng các hàm này ta khai báo tệp tiêu đề #include<ctype.h>
- int toupper(int ch)
 - Chuyển một kí tự chữ cái thường (các kí tự 'a', 'b', ..., 'z') thành kí tự chữ cái hoa tương ứng ('A', 'B', ..., 'Z').
- int tolower(int ch)
 - Chuyển một kí tự chữ cái hoa ('A', 'B', ..., 'Z') thành kí tự chữ cái thường tương ứng ('a', 'b', ...'z').
- int isalpha(int ch)
 - Kiểm tra một kí tự có phải là chữ cái hay không ('a', 'b',
 - Hàm trả về giá trị khác không nếu đúng là chữ cái, trả về giá trị 0 nếu ngược lại.

10



5.3.1. Các hàm xử lý ký tự (tiếp)

- int isdigit(int ch)
 - Kiểm tra một kí tự có phải là chữ số hay không ('0', '1', ...'9').
 - Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược
- int islower(int ch)
 - Kiểm tra một kí tự có phải là chữ cái thường hay không ('a', 'b',
 - Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược
- int isupper(int ch)
 - Kiểm tra một kí tự có phải là chữ cái hoa hay không ('A', 'B', ...'Z').
 - Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược



5.3.1. Các hàm xử lý ký tự (tiếp)

- int iscntrl(int ch)
 - Kiểm tra một kí tự có phải là kí tự điều khiển hay không (là các kí tự không hiến thị được và có mã ASCII từ 0 đến 31).
 - Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nếu ngược lại.
- int isspace(int ch)
 - Niễm tra một kí tự có phải là dấu cách (space, mã ASCII là 32), kí tự xuống dòng ('\n', mã ASCII là 10), kí tự về đầu dòng ('\r', mã ASCII là 13), dấu tab ngang ('\t', mã ASCII là 9) hay dấu tab dọc ('\v', mã ASCII là 11) hay không.
 - Hàm trả về giá trị khác không nếu đúng, trả về giá trị 0 nêu ngược lại.



5.3.2. Các hàm xử lý xâu ký tự

- Vào ra dữ liệu
 - Vào ra dữ liệu trên xâu kí tự tức là nhập dữ liệu cho xâu và hiển thị dữ liệu chứa trong xâu.
 - Khai báo thư viện <stdio.h>.
 - Để nhập dữ liệu cho xâu ta có thể sử dụng 2 hàm scanf() hoặc gets()
 - scanf("%s", ten_xau);
 - gets(ten_xau);
 - Để hiển thị nội dung xâu ta có thể dùng 2 hàm printf() hoăc puts()
 - printf("%s", ten xau);
 - puts(ten_xau);

13



5.3.2. Các hàm xử lý xâu ký tự

 Ví dụ: Nhập vào một chuỗi và hiển thị trên màn hình chuỗi vừa nhập.

```
#include<conio.h>
#include<stdio.h>
#include<string.h>
int main()
   char ten[12];
   printf("Nhap chuoi: "); gets(ten);
printf("Chuoi vua nhap: "); puts(ten);
   getch();
   return 0;
```



5.3.3. Một số hàm xử lý xâu ký tự khác

- Khai báo tệp tiêu đề #include<string.h>
- int strlen(char* tên xâu);
 - Trả về độ dài (số kí tự có trong xâu) của xâu kí tự tên_xâu.
- char* strcpy(char* xâu_đích, char* xâu_nquồn)
 - Sao chép nội dung xâu_nguồn và ghi lên xâu_đích.
- char* strncpy(char* xâu_đích, char* xâu_nguồn, int
 - Tương tự strcpy() nhưng ngừng sao chép sau n ký tự. Trong trường hợp không có đủ số ký tự trong xâu nguồn thì hàm sẽ điền thêm các ký tự trắng vào xâu đích.



5.3.3. Một số hàm xử lý xâu ký tự khác

- int strcmp(char* xâu_thứ_nhất, char* xâu_thứ_hai);
 - Trả về
 - giá trị < 0 nếu xâu_thứ_nhất nhỏ hơn xâu_thứ_hai
 - giá trị 0 nếu xâu_thứ_nhất bằng xâu_thứ_hai
 - giá trị > 0 nếu xâu_-thứ _nhất lớn hơn xâu_thứ_hai
- int strncmp(char* xâu_thứ_nhất, char* xâu_thứ_hai);
 - Tương tự như hàm strcmp nhưng giới hạn việc so sánh với n ký tự đầu tiên của hai xâu.
- int stricmp(char* xâu_thứ_nhất, char* xâu_thứ_hai);
 - Tương tự như strcmp() nhưng không phân biệt chữ hoa hay chữ
- int strnicmp(char* xâu_thứ_nhất, char* xâu_thứ_hai);
 - Tương tự như hàm stricmp nhưng giới hạn việc so sánh với n ký tự đầu tiên của hai xâu.



5.3.3. Một số hàm xử lý xâu ký tự khác

- char* strchr(char* str, int ch);
 - Tìm kiếm vị trí của kí tự ch trong xâu str.
 - Nếu có kí tự ch trong str thì hàm strchr() trả về con trỏ trỏ tới kí tự ch đầu tiên trong str, ngược lại nó sẽ trả về
- char* strrchr(char* str, int ch);
 - Tượng tự như strchr() nhưng việc tìm kiếm bắt đầu từ cuối xâu str.
- char* strstr(char* str1, char* str2);
 - Tìm kiếm vị trí của xâu con str2 trong xâu str1.
 - Nếu str2 là xâu con của str1 thì hàm strstr() trả về con trỏ trỏ tới kí tự đầu tiên của xâu con str2 đầu tiên trong str1, ngược lại nó sẽ trả về con trỏ NULL.



5.3.3. Một số hàm xử lý xâu ký tự khác

- char* strcat(char* xâu_đích, char* xâu_nguồn);
 - Ghép nối xâu_nguồn vào ngay sau xâu_đích.
 - Kết quả trả về của hàm strcat() là xâu mới ghép nối từ 2 xâu xâu_nguồn và xâu_đích.
- char* strncat(char* xâu_đích, char* xâu_nguồn, int n);
 - Tương tự streat nhưng chỉ giới hạn với n ký tự đầu tiên của xâu_nguồn.
- char* strlwr(char* xâu)
- Chuyển đổi các chữ in hoa trong xâu sang chữ thường.
- char* strupr(char* xâu)
 - Chuyển đổi các chữ thường trong xâu sang chữ in hoa.



5.3.3. Một số hàm xử lý xâu ký tự khác

- void strset(char* s, char c)
 - Khởi đầu giá trị tất cả các ký tự của xâu s bằng ký tự c
- void strnset(char* s, char c, int n)
 - Khởi đầu giá trị cho n ký tự đầu tiên của xâu s bằng ký tư c
- int atoi(char* str)
 - Chuyển một xâu kí tự là biểu diễn của một số nguyên thành số nguyên tương ứng.
 - Nếu chuyển đổi thành công, hàm atoi() trả về giá trị số nguyên chuyển đổi được, ngược lại trả về giá trị 0.

19



5.3.4. Con trỏ và xâu ký tự

- Như đã biết xâu ký tự là một dãy các ký tự đặt trong hai dấu nhay kép, ví dụ: "Dai Hoc Bach Khoa".
- Cũng giống như mảng, xâu ký tự là một hằng địa chỉ biểu thị địa chỉ đầu của mảng ký tự chứa xâu (Mảng này chứa các ký tự của xâu công thêm ký tự NIJI I).
- Vì vậy, nếu chúng ta khai báo char *pc; là một con tró kiểu char thì phép gán pc="Dai Hoc Bach Khoa" là hợp lệ. Sau khi thực hiện câu lệnh này, trong con trỏ pc có địa chỉ đầu của xâu "Dai Hoc Bach Khoa".

20



5.3.4. Con trỏ và xâu ký tự

 Giữa con trỏ kiểu char và mảng kiểu char có sự khác biệt. Xét ví du sau:

```
char ch[20];
char *pc;
pc = "Dai Hoc Bach Khoa";/*HOp lê*/
ch = "Dai Hoc Bach Khoa";/*Không hOp lê*/
```

câu lệnh gán đầu tiên là hợp lệ vì pc là con trỏ kiểu char, còn câu lệnh gán thứ hai không hợp lệ vì ch được khai báo là một mảng ký tự, do đó <u>ch</u> <u>là một hằng</u>, chúng ta không thể gán giá trị cho hằng được.

21



5.3.4. Con trỏ và xâu ký tự

→ Có thể thao tác trên tất cả các ký tự của xâu thông qua một con trỏ.

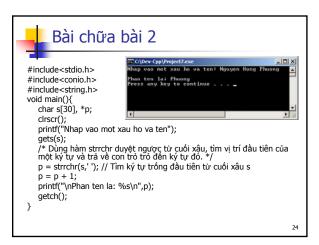
```
#include <stdio.h>
main()
{
   int i;
   char *str;
   str = "Lap trinh C that thu vi";
   for (i = 0; *str != '\0'; i++)
        str++;
   printf("so cac ky tu = %d\n",i);
}
```

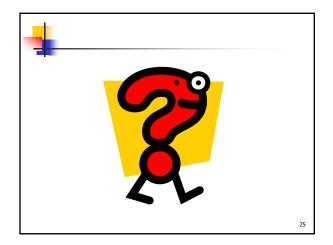
22

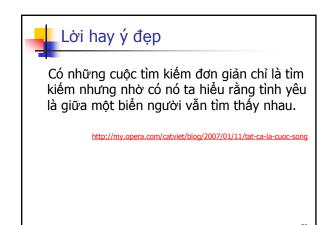


Bài tập

- Bài 1: Nhập một xâu kí tự từ bàn phím gồm các từ, ví dụ "Thu do Ha Noi". Lập chương trình để bỏ bót các dấu trống giữa các từ sao cho các từ chỉ cách nhau ít nhất một dấu trống.
- Bài 2: Viết chương trình nhập vào từ bàn phím họ và tên của một người, sau đó in phần tên ra màn hình. Ví dụ: "Tran Hung Dao" thì in ra "Dao".
- Bài 3: Nhập vào một câu, kết thúc bằng dấu chấm. In ra câu đó có bao nhiêu từ.









Chương 6 Hàm

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn
Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Nội dung chương này

- 6.1. Giới thiệu
- 6.2. Module chương trình trong C
- 6.3. Các hàm toán học
- 6.4. Các hàm người dùng định nghĩa
- 6.5. Nguyên mẫu hàm
- 6.6. Các tệp header
- 6.7. Truyền tham số cho hàm
- 6.8. Phạm vi biến
- 6.9. Đệ quy
- 6.10. Hàm main có đối dòng lệnh

2



6.1. Giới thiêu

- Có những đoạn chương trình được thực hiện lặp đi lặp lại nhiều lần, tuy dữ liệu có khác nhưng bản chất các công việc lại giống nhau.
- Viết gộp những đoạn chương trình đó lại thành một chương trình con mà khi cần chỉ việc truyền dữ liệu cho nó?
- Tư tưởng đó cũng dẫn chúng ta tới việc chia một chương trình lớn thành nhiều phần nhỏ rồi giải quyết từng phần; sau đó sẽ ráp nổi chúng lại là sẽ hoàn tắt một chương trình lớn. Các chương trình nhỏ này trong C chính là các hàm (function).
- Như vậy một chương trình sẽ là một tập hợp các định nghĩa hàm riêng biệt.

7

6.2. Module chương trình trong C

Các module trong C được gọi là hàm.



Hàm chuẩn

Hàm tự viết

(có trong thư viện) (Hàm người dùng định nghĩa)

Hàm: viết một lần, dùng lại ở nhiều nơi.

4



6.2. Module chương trình trong C (tiếp)

- Các hàm được gọi để thực hiện bằng các lời gọi hàm.
- Các lời gọi hàm xác định tên của hàm và cung cấp các thông tin (hay còn gọi là các tham số) mà hàm được gọi theo đúng trình tư khi khai báo hàm đó.



6.3. Các hàm toán học trong C

- Khai báo tệp tiêu đề #include<math.h>
- Khi dùng chỉ việc viết lời gọi hàm
- Ví dụ: viết hàm tính và in ra căn bậc 2 của 900.0 là : printf("%.2f", sqrt(900.0));
 - sqrt là hàm khai căn bậc 2
 - số 900.0 là tham số của hàm sqrt
 - hàm sqrt nhận tham số kiểu double và trả về giá trị kiểu double
 - Câu lệnh sẽ in ra 30.00

6



6.3. Các hàm toán học trong C (tiếp)

- Các tham số của hàm có thể là các hằng, biến hay các biểu thức
- Ví dụ nếu c1 = 13.0, d = 3.0 thì câu lệnh printf("%.2f", sqrt(c1 + d * 4.0)); sẽ tính và in ra căn bậc 2 của 13.0 + 3.0 * 4.0 = 25.0 là 5.00

6.3.	Các hàm toán	học trong C (tiế
Hàm	Mô tả	Ví dụ
sqrt(x)	Căn bậc 2 của x	sqrt(9.00) bằng 3.0
exp(x)	hàm mũ e ^x	exp(1.0) bằng 2.718282
log(x)	logarithm tự nhiên (cơ số e) của x	log(2.718282) bằng 1.0
log10(x)	logarithm thập phân (cơ số 10) của x	log10(1.0) bằng 0.0 log10(10.0) bằng 1.0
fabs(x)	giá trị tuyệt đối của x	nếu x 0 thì fabs(x) bằng x nếu x<0 thì fabs(x) bằng -x
ceil(x)	làm tròn thành số nguyên nhỏ nhất lớn hơn x	ceil(9.2) bằng 10.0 ceil(-9.8) bằng -9.0



6.3. Các hàm toán học trong C (tiếp)

_		
floor(x)	làm tròn thành số	floor(0.2) bằng 0.0
noor(x)	nguyên lớn nhất nhỏ hơn x	floor(-9.8) bằng -10.0
pow(x,y)	x mũ y (x ^y)	pow(2,7) bằng 128
fmod(x,y)	phần dư của phép chia x cho y	fmod(13.657,2.333) bằng 1.992
sin(x)	sin của x (x theo radian)	sin(0.0) bằng 0.0
cos(x)	cos của x (x theo radian)	cos(0.0) bằng 1.0
tan(x)	tan của x (x theo radian)	tan(0.0) bằng 0.0



6.4. Hàm người dùng định nghĩa

- Xét bài toán: tính bình phương của 10 số tự nhiên từ 1 đến 10;
- Ta sẽ tổ chức chương trình thành 2 hàm hàm main() và một hàm nữa mà ta đặt tên là square(); nhiệm vụ của hàm square() là tính bình phương của một số nguyên; còn hàm main() là sẽ gọi hàm square() để thực hiện nhiệm vụ được đề ra.

10

f.4. Hàm người dùng định nghĩa (tiếp) #include<stdio.h> #include<conio.h> int square(int) /* Hàm nguyên mẫu */ void main() { int x; for (x = 1; x <= 10; x++) printf("%d ", square(x)); printf("\n"); getch(); } int square(int y) /* Định nghĩa hàm square */ { return y * y; } Kết quả 1 4 9 16 25 36 49 64 81 100</pre>



6.4. Hàm người dùng định nghĩa (tiếp)

Khuôn dạng một hàm:

```
kiểu tên-hàm (danh-sách-tham-số)
{
    Các khai báo
    ...
    Các câu lệnh
    [return [bieu_thuc];]
}
```



6.4. Hàm người dùng định nghĩa (tiếp)

- Giải thích:
- Ở đây kiểu là kiểu dữ liệu của kết quả của hàm, tênhàm là một cái tên do người lập trình tự đặt;
- Danh-sách-tham-số gồm danh sách các tham số hàm nhận được khi nó được gọi và được ngăn cách nhau bởi dấu phẩy.
- Các khai báo và các câu lệnh trong ngoặc là phần thân của hàm. Khai báo và cài đặt một hàm lại không được nằm trong một hàm khác trong bất kỳ tình huống nào.

13



6.4. Hàm người dùng định nghĩa (tiếp)

- Hàm có thể có giá trị trả về hoặc không có giá trị trả về.
 - Nếu có giá trị trả về, trong thân hàm có ít nhất một lệnh return.
 - Nếu không có giá trị trả về cần khai báo cho hàm đó có kiểu trả về là void.

14



6.4. Hàm người dùng định nghĩa (tiếp)

Ví dụ:

15



6.4. Hàm người dùng định nghĩa Ví du thứ ba

 Xây dựng hàm maximum trả về số nguyên lớn nhất trong ba số nguyên.

16



6.4. Hàm người dùng định nghĩa Ví du thứ ba (tiếp)

```
/* Định nghĩa hàm maximum */
int maximum(int x, int y, int z)
{
  int max = x;
  if (y > max)
  max = y;
  if (z > max)
  max = z;
  return max;
```

Két quả

Vào 3 số nguyên: 22 85 17 Số lớn nhất là: 85 Vào 3 số nguyên: 85 22 17 Số lớn nhất là: 85

17



6.5. Nguyên mẫu hàm (Prototype)

- Một trong những đặc trưng quan trọng của ANSI C là hàm nguyên mẫu (khai báo hàm). Hàm nguyên mẫu thông báo cho trình biên dịch biết kiểu dữ liệu hàm trả lại, số lượng, kiểu và trình tự của các tham số truyền cho hàm. Trình biên dịch dùng hàm nguyên mẫu để kiểm tra các lời gọi hàm.
- Quan sát lại ví dụ đầu tiên:

```
#include<stdio.h>
#include<conio.h>
int square(int) * Hàm nguyên mẫu */
void main()
{
  int x;
  for (x = 1; x <= 10; x++)
  printf( "%d ", square(x));
  printf("\n");
  getch();
}
int square(int y) /* Định nghĩa hàm square */
{
  return y * y;
}
```



6.6. Các tệp header

- Mỗi một thư viện chuẩn tương ứng có một tệp header chứa các khai báo của tất cả các hàm trong thư viện này cùng với các định nghĩa các kiểu dữ liệu khác nhau, các hàng dùng trong các hàm đó
- Lập trình viên có thể tạo các tệp header riêng của mình. Các tệp header này cũng thường có kiểu .h và có thể dùng chỉ thị tiền xử lý #include để thêm vào chương trình.

20



6.7. Truyền tham số cho hàm

- Có hai cách để truyền các tham số cho hàm:
 - Khi các tham số được truyền theo trị, một bản sao giá trị của tham số được tạo ra và truyền cho hàm. Vì vậy mọi sự thay đổi trong hàm trên bán sao sẽ không ảnh hưởng đến giá trị ban đầu của biến nằm trong hàm gọi.
 - Khi một tham số được truyền theo con trở, hàm gọi sẽ truyền trực tiếp tham số đó cho hàm bị gọi thay để đổi giá trị nguyên thuỷ của biến truyền vào tham số.
- Trong C, tắt cả các tham số được truyền đều là truyền theo trị.
- Ta có thể mô phỏng cách truyền tham số theo tham chiếu bằng cách truyền địa chỉ của các biến vào tham số.



6.7. Truyền tham số cho hàm - ví dụ

- Viết hàm hoán đổi hai số nguyên
- Hãy quan sát hai cách sau

22

```
6.7. Truyền tham số cho hàm - ví dụ
  Cách 1:
#include<stdio.h>
void swap(int, int);
                        /* Hàm nguyên mẫu */
void main(){
  int x, y;
x = 10; y = 20;
  printf("Ban dầu x = %d, y = %d", x, y);
  swap(x, y);
  printf("Sau đó x = %d, y = %d", x, y);
                  void swap( int x, int y) /* Định nghĩa hàm swap */
  int temp;
  temp = x;
                             Liệu có thực sự
  y = temp;
                                                     23
```

```
6.7. Truyền tham số cho hàm - ví du
 Cách 2:
include <stdio.h>
void swap( int* , int* );
void main(){
                              /* Hàm nguyên mẫu */
 int x, y;
  printf("Ban dầu x = %d, y = %d", x, y);
  swap(&x, &y);
 printf("Sau dó x = %d, y = %d", x, y);
void swap( int* x, int* y){
                              /* Định nghĩa hàm swap */
 int temp;
  temp = *x;
                            Truyền địa chỉ của biến
  *x = *y;
                            vào vị trí các tham số
  *y = temp;
                                                       24
```



6.7. Truyền tham số cho hàm-nhận xét

- Nếu đối của hàm là con trỏ kiểu int, float, double,... thì tham số thực sự tương ứng trong lời gọi hàm phải là địa chỉ của biến hoặc địa chỉ của phần tử mảng có kiểu int, float, double... Khi đó, địa chỉ của biến được truyền cho đối con trỏ tương ứng. Do đã biết được địa chỉ của biến nên có thể gán cho biến các giá trị mới.
- Các đối của hàm có hai loại: các đối chứa dữ liệu đã biết (gọi là đối vào), các đối chứa kết quả mới nhận được (đối ra). Các đối ra phải khai là các đối con trỏ.

25



6.7. Truyền tham số cho hàm

- Hàm có đối là mảng một chiều: Hai cách khai báo hàm:
 - void Tên_hàm(float *pa,...): pa là một con trỏ.
 - void Tên_hàm(float pa[],...): pa coi như một mảng hình thức.
 - Nếu tham số thực là tên mảng a một chiều kiểu float thì ta gọi Tên_hàm(a,...).
 - Khi hàm bắt đầu làm việc thì giá trị của a (địa chỉ phần tử a[0]) được truyền cho pa. Do đó, muốn truy nhập tới phần tử a[i] trong thân hàm ta dùng *(pa+i) hoặc pa[i].
 - Với các kiểu char, int, double, cấu trúc,...ta cũng làm tương tự như trên.

26



6.7. Truyền tham số cho hàm

- Hàm có đối là mảng 2 chiều: Giả sử a là mảng 2 chiều float a[20][30]. Ta có thể khai báo theo hai cách sau:
 - Cách 1:
 - void Tên_hàm(float (*pa)[50], int m, int n)
 - void Tên_hàm(float pa[][50], int m, int n)
 - pa là con trỏ kiểu float[50] (chứa địa chỉ đầu của vùng nhơ dành cho 50 số thực, tức là vùng nhớ 200 byte), m là số hàng và n là số côt
 - Trong thân hàm, để truy nhập tới phần tử hàng i cột j, ta dùng kí hiệu pa[i][j].
 - Theo cách này, ta chỉ làm việc với màng 2 chiều có tối đa 50 cột. Lời gọi hàm Tên_hàm(a, m, n)

27



6.7. Truyền tham số cho hàm

- Hàm có đối là mảng 2 chiều
 - Cách 2:
 - float *pa; biểu thị địa chỉ đầu của mảng a
 - int N; biểu thị số cột của mảng a
 - Khai báo: Tên_hàm(float *pa, int N,...)
 - Lời gọi: Tên_hàm(a, 30,...)
 - Trong thân hàm, để truy nhập tới phần tử a[i][j] ta dùng công thức *(pa + i*N + j)
 - Theo cách này, mảng 2 chiều quy về mảng 1 chiều, hàm có thể dùng cho bất kỳ mảng 2 chiều nào.

28



6.8. Pham vi biến

- Biến địa phương (Local Variable):
 - Là các biến được khai báo trong lệnh khối hoặc trong thân chương trình con.
- Biến toàn cục (Global Variable):
 - Là biến được khai báo trong chương trình chính.
 - Vị trí khai báo của biến toàn cục là sau phần khai báo tệp tiêu đề và khai báo hàm nguyên mẫu

29

#include <stdio.h> void main() { int a = 1; printf("\n a = %d",a); { int a = 2; printf("\n a = %d",a); } printf("\n a = %d",a); } int a = 3; printf("\n a = %d",a); } int a = 3; printf("\n a = %d",a); } 30

```
#include <stdio.h>
#include <conio.h>
int a, b, c;
int tich()

{
    printf("\n Gia tri cac bien tong the a, b, c: ");
    printf(" a = %-5d b = %-5d c = %-5d", a, b, c);
    return a*b*c;
}

void main()

{
    clrscr();
    printf("\n Nhap gia tri cho 3 so nguyen a, b, c: 2 3 4
    Gia tri cac bien tong the a, b, c: 2 3 4
    Gia tri cac bien tong the a, b, c: 2 3 4
    Gia tri cac bien tong the a, b, c: 2 3 4
    Gia tri cac bien tong the a, b, c: 3 so a 2 4
    include 3 so la 24

clrscr();
    printf("\n Nhap gia tri cho 3 so nguyen a, b, c: ");
    scanf("%d %d %d", &a, &b, &c);
    printf("\n Tich cua 3 so la %d", tich());
    getch();
}
```



Biến register

- Thanh ghi có tốc độ truy nhập nhanh hơn so với các loại bộ nhớ khác (RAM, bộ nhớ ngoài).
- Nếu một biến thường xuyên sử dụng được lưu vào trong thanh ghi thì tốc độ thực hiện của chương trình sẽ được tăng lên.
- Để làm điều này ta đặt từ khóa register trước khai báo của biến đó.
 - Ví dụ: register int a;
- Số lượng và kích thước các thanh ghi có hạn → Số lượng biến khai báo register sẽ không nhiều và thường chỉ áp dụng với những biến có kích thước nhỏ như kiểu char, int.

32

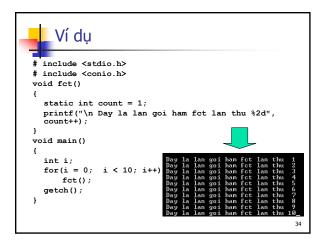


Biến static

- Một biến cục bộ khi ra khỏi phạm vi của biến đó thì bộ nhớ dành để lưu trữ biến đó sẽ được giải phóng.
- Nếu cần lưu giá trị của các biến cục bộ này, cần khai báo biến với từ khóa static.
 - Ví du: static int a;
- Biến static là biến tĩnh, nghĩa là nó sẽ được cấp phát một vùng nhớ thường xuyên từ lúc khai báo và chỉ giải phóng khi chương trình chính kết thúc.

33

35





6.9. Đệ quy

- Trong C cho phép trong thân một hàm có thể gọi ngay đến chính nó, cơ chế này gọi là đệ qui.
- Cổ thể định nghĩa hàm đệ qui là hàm sẽ gọi đến chính nó trực tiếp hay gián tiếp thông qua các hàm khác.
- Cách tiến hành giải một bài toán đệ qui nhìn chung có những điểm chung sau.
 - Có trường hợp cơ sở
 - Nó gọi đển chính dạng của nó nhưng có sự chuyển biển.
 - Sự chuyển biến ấy đến một lúc nào đó phải kết thúc.

6.9. Đệ quy - ví dụ 1

- Ta xem xét ví dụ tính n giai thừa (n!).
 - Theo định nghĩa n! bằng tích của tất cả các số tự nhiên từ 1 đến n: n! = n * (n-1) ... 2 * 1
 - Ví dụ: 5! = 5 * 4 * 3 * 2 * 1 = 120, 0! được định nghĩa bằng 1.
 - Để tính n! có thể dùng vòng lặp như sau:

```
fact = 1;
for (i = n; i >= 1; i--)
fact *= i;
```

 Tuy nhiên cũng có thể định nghĩa đệ qui hàm giai thừa như sau :

nếu n>0 thì n! = n * (n-1)!nếu n=0 thì n! = 0! = 1

```
/* Tính giai thừa theo phương pháp dệ qui */
#include <stdio.h>
long fact(long); /* Hàm nguyên mẫu */
main()
{
  int i;
  for (i=0; i<=10; i=i+2)
  printf("%2d! = %ld\n", i, fact(i));
}
/* Định nghĩa dệ qui hàm factorial */
long fact(long n)
{
  if (n == 0) return 1;
  else return (n*fact(n-1));
}
```

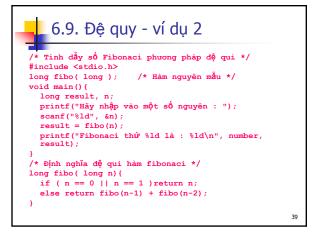


6.9. Đệ quy - ví dụ 2

- Một ví dụ thứ hai dùng đệ qui là tính dẫy số Fibonaci
 - Dãy số Fibonaci gồm những số
 0, 1, 1, 2, 3, 5, 8, 13, 21 ...
 - Bắt đầu từ hai số 0 và 1, tiếp sau đó các số
 Fibonaci sau bằng tổng của 2 số Fibonaci trước nó.
 - Dãy Fibonaci có thể định nghĩa đệ qui như sau:
 fibonaci(0) = 0; fibonaci(1) = 1;

```
fibonaci(n) = fibonaci(n-1) + fibonaci(n-2) n>1
```

38





So sánh đệ quy và lặp

- So sánh:
 - Lập trình đệ quy sử dụng cấu trúc lựa chọn, còn lặp sử dụng cấu trúc lặp.
 - Cà hai phương pháp đều phải kiểm tra khi nào thì kết thúc.
 Phương pháp lặp kết thúc khi điều kiện để tiếp tục vòng lặp sai, còn phương pháp đệ quy kết thúc khi đến trường hợp cơ sở
 - Tuy nhiên đệ quy tồi hơn, nó liên tục đưa ra các lời gọi hàm làm tốn thời gian xử lý của bộ vi xử lý và không gian nhớ.
- "Công nghệ phần mềm" là quan trọng nhưng hiệu năng của chương trình cũng quan trọng và thật không may hai mục tiêu đó lại thường loại trừ lẫn nhau.
- Vậy khi module hoá chương trình phải khôn ngoan cân đối giữa hai mục tiêu là Công nghệ phần mềm và Hiệu năng

40



6.10. Hàm main có đối dòng lệnh

- Giả sử ta có chương trình echo.exe và chạy nó theo dòng lệnh:
 C:\>echo Hanoi 2008
- Khi đó, "Hanoi" và "2008" là các tham số dòng lênh
- Hàm main có 2 tham số:
 - argc (argument count): số lượng đối số dòng lệnh
 - argv (argument vector): con trỏ trỏ tới mảng xâu kí tự, mảng này chứa các đối dòng lệnh, mỗi đối là một xâu
- Khi gõ vào từ dòng lệnh, các tham số phải cách nhau bởi dấu cách. Chương trình xử lý các tham số dòng lệnh như các dữ liệu vào.

Mã nguồn chương trình echo.c



- Bài 1: Viết hàm tìm ước số chung lớn nhất uscln(int x, int y) và bội số chung nhỏ nhất bscnn(int x, int y) của hai số.
- Bài 2: Lập hàm giải phương trình bậc 2 ptb2(float a, float b, float c, float *x1, float *x2), trong đó a,b,c là các hệ số của phương trình. Hàm nhận giá trị 0 nếu phương trình vô nghiệm, giá trị 1 nếu phương trình có nghiệm kép (chứa trong *x1), giá trị 2 nếu phương trình có 2 nghiệm (chứa trong *x1 và *x2), giá trị 3 nếu phương trình có vô số nghiệm.

Bài tập

Bài 3: Lập hàm tính giai thừa và dùng hàm này tính tổng:

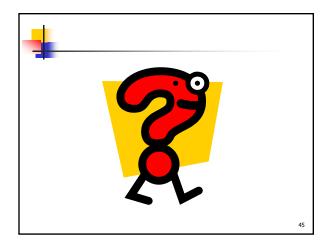
1! + 2! +...+N! (N nhập từ bàn phím và <=10)

Bài 4: Đa thức cấp n được tính theo công thức:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x^1 + a_0 x^0$$

 Lập hàm đa thức f(float a[], int n, float x) để tính giá trị của đa thức.

44





Lời hay ý đẹp

Có những ước mơ sẽ vẫn chỉ là ước mơ dù cho ta có nỗ lực đến đâu nhưng nhờ có nó ta mạnh mẽ hơn, yêu cuộc sống hơn và biết cố gắng từng ngày.

Có những lời hứa cũng vẫn chỉ là lời hứa dù ta có mãi chờ đợi bởi người hứa đã không còn nhớ, nhưng nhờ có nó ta biết hi vọng và mong chờ.

http://my.opera.com/catviet/blog/2007/01/11/tat-ca-la-cuoc-song



Chương 7 Cấu trúc (struct)

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội

1



Nội dung

7.1. Khái niệm cấu trúc

- 7.2. Khai báo và sử dụng cấu trúc
- 7.3. Xử lí dữ liêu cấu trúc
- 7.4. Mảng cấu trúc
- 7.5. Bài tập

2



7.1. Khái niệm cấu trúc

- Kiểu dữ liệu cấu trúc (struct) là kiểu dữ liệu phức hợp bao gồm nhiều thành phần, mỗi thành phần có thể thuộc những kiểu dữ liêu khác nhau.
- Các thành phần dữ liệu trong cấu trúc được gọi là các trường dữ liệu (field).



7.1. Khái niệm cấu trúc (tiếp)

- Ví du: Cấu trúc mô tả sinh viên:
 - Cần lưu giữ các thông tin liên quan đến sinh viên như họ tên, tuổi, kết quả học tập...
 - Mỗi thông tin thành phần lại có kiểu dữ liệu khác nhau như họ tên có kiểu dữ liệu là xâu kí tự, tuổi có kiểu dữ liệu là số nguyên, kết quả học tập có kiểu dữ liệu là số thực.

4



Nội dung

■ 7.1. Khái niệm cấu trúc

7.2. Khai báo và sử dụng cấu trúc

- 7.3. Xử lí dữ liêu cấu trúc
- 7.4. Mảng cấu trúc
- 7.5. Bài tập

5



7.2. Khai báo kiểu dữ liêu cấu trúc



7.2.2. Khai báo biến cấu trúc

Cú pháp:

struct ten_cau_truc ten_bien_cau_truc;

Ví du:

```
struct sinh_vien a, b, c;
```



7.2.3. Khai báo kết hợp

 Có thể kết hợp vừa khai báo kiểu dữ liệu cấu trúc vừa khai báo biến cấu trúc:

```
struct [ten_cau_truc]
{
    //khai_bao_cac_truong;
} ten_bien_cau_truc;

Ví dụ:
struct thi_sinh
{
    char SBD[10]; // so bao danh
    char ho_va_ten[30];
    struct diem_thi ket_qua;
} thi_sinh_1, thi_sinh_2;
```

۰



7.2.4. Định nghĩa kiểu dữ liệu cấu trúc với typedef

 Đặt tên mới cho kiểu dữ liệu cấu trúc cú pháp:

typedef struct tên_cũ ten_moi;

 Hoặc ta có thể đặt lại tên cho cấu trúc ngay khi khai báo bằng cú pháp

```
typedef struct [tên_cũ]
{
   //khai_bao_cac_truong;
}danh_sach_cac_tên_mới;
```

9



7.2.4. Định nghĩa kiểu dữ liệu cấu trúc với typedef (tiếp)

- Lưu ý: Được phép đặt tên_mới trùng với tên_cũ.
- Vidu:
 struct point_3D
 {
 float x, y, z;
 } P;
 struct point_3D M;
 typedef struct point_3D p_3D;
 p_3D N;

10



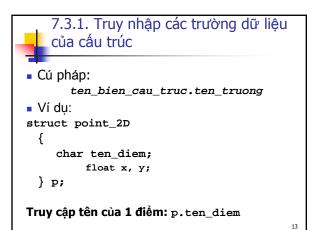
Nội dung

- 7.1. Khái niệm cấu trúc
- 7.2. Khai báo và sử dụng cấu trúc
- 7.3. Xử lí dữ liêu cấu trúc
- 7.4. Mảng cấu trúc
- 7.5. Bài tập



7.3. Xử lý dữ liệu cấu trúc

- Truy nhập các trường dữ liệu của cấu trúc
- Phép gán giữa các biến cấu trúc
- Con trỏ cấu trúc





7.3.2. Phép gán giữa các biến cấu trúc

- Giả sử ta có 2 biến cấu trúc là a và b có
 Cùng kiểu dữ liệu là một cấu trúc nào đó.
- Phép gán cấu trúc có cú pháp tương tự như phép gán thông thường:

```
bien_cau_truc_1 = bien_cau_truc_2;
```

14



7.3.3. Con trỏ cấu trúc

Cú pháp:

struct ten_cau_truc *ten_bien_con_tro_ctruc;

 Có 2 cách truy nhập vào trường dữ liệu của cấu trúc từ biến con trỏ cấu trúc là

 $(\verb|*ten_bien_con_tro_ctruc|).ten_truong_du_lieu$

Hoặc:

 ${\tt ten_bien_con_tro_ctruc\text{-}}{\tt >ten_truong_du_lieu}$

15



Nội dung

- 7.1. Khái niệm cấu trúc
- 7.2. Khai báo và sử dụng cấu trúc
- 7.3. Xử lí dữ liệu cấu trúc
- 7.4. Mảng cấu trúc
- 7.5. Bài tập

16



7.4. Mảng cấu trúc

```
Cú pháp:
    struct ten_cau_truc ten_mang_ctruc[so_ptu];

Ví dụ:
    struct sinh_vien
    {
        char ho_ten[20];
        float diem_thi;
    };

struct sinh_vien lop_Tin5[80];
```

```
Ví dụ
```

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
   struct sinh_vien
   {
      char ma_sv[10];
      char ho_ten[20];
      float diem_thi;
   };
   struct sinh_vien sv[3];
   int i;
   clrscr();
```

```
for(i=0;i<3;i++)
{
  char str[20];
  float diem;
  printf("\nNhap thong tin cho sinh vien %d",i+1);
  printf("\nNhap thong tin cho sinh vien %d",i+1);
  printf("\nMa so sinh vien:");
  fflush(stdin); gets(str);
  strcpy(sv[i].ma_sv,str);
  printf("\n Ho va ten: ");
  fflush(stdin); gets(str);
  strcpy(sv[i].ho_ten,str);
  printf("\n Diem thi: ");
  scanf("%f",&diem);
  sv[i].diem_thi = diem;
}</pre>
```

```
Ví dụ (tiếp)

printf("\n Thong tin ve cac sinh
  vien");
  for(i=0;i<3;i++)
  {
    printf("\n Sinh vien thu %d ",i+1);
    printf("%-10s %-20s %-3.1f",
    sv[i].ma_sv, sv[i].ho_ten,
    sv[i].diem_thi);
  }
  getch();
}</pre>
```



Ví dụ - Kết quả

Nhap thong tin cho sinh vien thu 1
Ma so sinh vien: SV0032
Ho va ten: Nguyen Thanh Binh
Diem thi: 8.5
Nhap thong tin cho sinh vien thu 2
Ma so sinh vien: SV0002
Ho va ten: Pham Hong Phuc
Diem thi: 9
Nhap thong tin cho sinh vien thu 3
Ma so sinh vien: SV0046
Ho va ten: Le Minh Hoa
Diem thi: 10
Thong tin ve cac sinh vien
Sinh vien thu 1: SV0032
Nguyen Thanh Binh
Sinh vien thu 2: SV0002
Pham Hong Phuc
9.0
Sinh vien thu 3: SV0046
Le Minh Hoa
10.0

21



Bài tập

- Viết 1 chương trình thực hiện những công việc sau:
 - Yêu cầu người dùng nhập vào một số nguyên dương n với 5 n 20 (có kiểm tra tính hợp lệ của giá trị được nhập vào, nếu giá trị n nhập vào không thỏa mãn điều kiện thì yêu cầu nhập lại)
 - Yêu cầu người dùng nhập vào thông tin của n sinh viên gồm những mục sau
 - Họ và tên: có kiểu dữ liệu là xâu kí tự gồm không quá 30 kí tự
 - Lớp: xâu kí tự có độ dài không quá 5 kí tự
 - Điểm thi Tin đại cương: là một số nguyên có giá trị từ 0 đến 10
 - Đưa ra màn hình danh sách các sinh viên cùng thông tin của họ mà người dùng vừa nhập vào.

22



Bài tập

- Yêu cầu người dùng nhập vào từ bàn phím một số thực. Đưa ra màn hình danh sách các sinh viên có điểm thi Tin đại cương nhỏ hơn giá trị số thực vừa nhập vào.
- Đưa ra màn hình danh sách sinh viên được sắp xếp theo chiều giảm dần của điểm thi Tin đại cương.
- Đưa ra màn hình danh sách sinh viên với họ và tên được sắp xếp theo chiều của bảng chữ cái.



Hỏi - đáp



24





Chương 8 Têp (FILE)

Nguyễn Hồng Phương

Email: phuongnh-fit@mail.hut.edu.vn Website: http://is.hut.edu.vn/~phuongnh

Bộ môn Các hệ thống thông tin

Viện Công nghệ thông tin và Truyền thông

Đại học Bách Khoa Hà Nội



Nôi dung

- 8.1. Giới thiêu
- 8.2. Kiểu xuất nhập nhị phân và văn bản
- 8.3. Các hàm thao tác cấp 2
- 8.4. Đóng/mở tệp, xóa vùng đệm, kiểm tra lỗi
- 8.5. Nhập xuất ký tư
- 8.6. Các hàm nhập xuất theo kiểu văn bản
- 8.7. Tệp văn bản và các thiết bị chuẩn
- 8.8. Các hàm nhập xuất theo kiểu nhị phân
- 8.9. Nhập xuất ngẫu nhiên, di chuyển con trỏ chỉ



8.1. Giới thiệu

- Môt têp tin đơn giản chỉ là một dãy các byte (mỗi byte có giá trị từ 0 đến 255) ghi trên đĩa. Số byte của dãy chính là độ dài của tệp.
- Chương này trình bày các thao tác trên tệp như tạo một tệp mới, ghi dữ liệu từ bộ nhớ lên tệp, đọc dữ liệu từ tệp vào bộ nhớ,...
- Trong C, các thao tác trên tệp được thực hiện nhờ các hàm thự viện. Các hàm này được chia thành 2 nhóm: cấp 1 và cấp 2.
- Mỗi hàm (cấp 1 hay cấp 2) đều có thể truy xuất theo cả hai kiểu nhị phân và văn bản.



8.1. Giới thiệu

- Các hàm cấp 1:
 - thực hiện việc đọc/ghi như DOS
 - Không có dịch vụ xuất nhập riêng cho từng kiểu dữ liệu mà chỉ có dịch vụ đọc/ghi một dãy các byte. Ví dụ: để ghi 1 số thực lên đĩa, ta dùng dịch vụ ghi 4 byte; để ghi 10 số nguyên lên đĩa, ta dùng dịch vụ ghi 20 byte.
 - Mỗi tệp có một số hiệu (handle). Các hàm cấp 1 làm việc với tệp thông qua số hiệu tệp này.



8.1. Giới thiêu

- Các hàm cấp 2:
 - được xây dựng từ các hàm cấp 1 nên dễ sử dụng và có nhiều khả

 - nang nơn.

 có dịch vụ truy xuất cho từng kiểu dữ liệu. Ví dụ: hàm xuất nhập ký tự, chuỗi, số nguyên, số thực, cấu trúc,...

 C tự động cung cấp một vùng đệm. Mỗi lần đọc/ghi thường tiến hành trên vùng đệm chứ không hẫn trên tệp. Khi ghi dữ liệu thì dữ liệu được đưa vào vùng đệm, khi nào vùng đệm đầy thì dữ liệu ở vùng đệm mội được đấy lên đĩa. Khi đọc, thông tin được lấy ra từ vùng đệm, khi nào vùng đệm trống thì máy mới lấy dữ liệu từ đĩa dưa vào vùng đệm giảm số lần nhập xuất trên đĩa, nằng cạo tốc đô lầm việc. nâng cao tốc độ làm việc.
 - làm việc với tệp thông qua một biến con trỏ tệp.

8.2. Kiểu nhập xuất nhị phân và văn bản

- 8.2.1. Kiểu nhị phân
 - Bảo toàn dữ liệu: trong quá trình xuất nhập, dữ liệu không bị biến đổi
 - Mã kết thúc tệp: trong khi đọc, nếu gặp cuối tệp thì ta nhận được mã kết thúc tệp EOF (giá trị là -1) và hàm feof cho giá trị khác 0. Tại sao lại chọn giá trị -1? Lý do rất đơn giản: chưa gặp cuối têp thì sẽ đọc được một byte có giá tri từ 0 đến 255. Giá tri -1 sẽ không trùng với bất kỳ byte nào.



8.2. Kiểu nhập xuất nhị phân và văn bản

- 8.2.2. Kiểu văn bản:
 - chỉ khác kiểu nhị phân khi xử lý ký tự chuyển dòng (mã 10) và ký tự mã 26
 - Mã chuyển dòng:
 - khi ghi, một ký tự LF (mã 10) được chuyển thành 2 ký tự CR (mã 13) và LF.
 - (ma 13) va LF.

 khi đọc, 2 ký tự liên tiếp CR và LF trên tệp chỉ cho ta một ký tự LF.
 - Ví dụ: xét hàm fputc(10,fp); nếu tệp fp mở theo kiểu nhị phân thì hàm sẽ ghi lên tệp một ký tự mã 10; nhưng nếu fp mở theo kiểu văn bản thì hàm ghi lên tệp hai mã là 13 và 10.
 - Mã két thúc tệp: khi đọc, nếu gặp ký tự có mã 26 hoặc cuối tệp thì ta nhận được mã kết thúc tệp EOF (số -1) và hàm feof(fp) cho giá trị khác 0.

7



8.2.3. Ví dụ minh họa 1

- Chương trình sau tạo 2 tệp có tên là vb và np. Trong chương trình dùng các hàm:
 - fopen để mở tệp
 - fputc để ghi một ký tự lên tệp
 - fclose để đóng tệp

8



8.2.3. Ví dụ minh họa 1 (tiếp)

#include<stdio.h>
void main(){

FILE *fvb, *fnp; //Khai báo 2 biến con trỏ tệp

fvb = fopen("vb","wt"); //Mở tệp vb để ghi theo kiểu văn bản

fnp = fopen("np","wb"); //Mở tệp np để ghi theo kiểu văn bản

// Ghi các ký tự lên tệp fvb

fputc('A',rvb); fputc(26,rvb); fputc(10,rvb); fputc('B',rvb);

// Ghi các ký tự lên tệp fnp

fputc('A',fnp); fputc(26,fnp); fputc(10,fnp); fputc('B',fnp);

fclose(fvb);

fclose(fnp);

9



8.2.3. Ví dụ minh họa 1 (tiếp)

- Két quả:
 - Tệp vb có các ký tự ứng với các mã: 65 26 13 10 66
 - Tệp np có các ký tự ứng với các mã: 65 26 10 66
- Chú ý:
 - nếu dùng kiểu văn bản để đọc tệp vb hay tệp np thì ta chỉ nhận được một ký tự đầu (mã 65) vì khi gặp ký tự thứ hai (mã 26) thì ta nhận được mã kết thúc tệp.
 - muốn đọc tất cả các ký tự của tệp, ta cần dùng hàm fgetc theo kiểu nhị phân.

10



8.2.4. Ví du minh hoa 2

Xét chương trình sau:

#include<stdio.h>
void main(){
 FILE *f; // Khai báo biến con trỏ tệp
 f = fopen("sl","wt"); //Mở tệp sl để ghi theo kiểu văn bản
 // Ghi 3 dòng lên tệp f
 fprintf(f,"%2d\n%2d\n%2d",56,7,8);
 fclose(f); // Đóng tệp
}

11



8.2.4. Ví dụ minh họa 2 (tiếp)

Hàm fprintf() đưa kết quả ra tệp theo cách như hàm printf() đưa ra màn hình. Vì tệp f mở theo kiểu văn bản nên ký tự xuống dòng '\n' được ghi thành 2 mã 13 và 10. Kết quả là 10 ký tự ứng với các mã sau được ghi lên tệp:

53 54 13 10 32 55 13 10 32 56

trong đó: 53 là mã của chữ số 5, 54 là mã của chữ số 6, 13 là CR, 10 là LF, 32 là mã của khoảng trống, 55 là mã của chữ số 7, 56 là mã của chữ số 8.



8.2.4. Ví dụ minh họa 2 (tiếp)

- Nếu dùng trình soạn thảo văn bản (ví dụ notepad) để mở tệp trên thì ta sẽ nhìn thấy các số 56, 7, 8 trên 3 dòng khác nhau.
- Nếu mở tệp sI theo kiểu nhị phân bằng cách dùng câu lệnh:

f = fopen("sl","wb"); thì tệp sI sẽ gồm 8 mã sau: 53 54 10 32 55 10 32 56



8.3. Các hàm cấp 2

- Các hàm dùng chung cho cả 2 kiểu:

 fopen dùng để mở tệp.
 fclose dùng để đóng tệp.

 - fcloseall dùng để đóng tất cả các tệp đang mở.
 - fflush dùng để làm sạch vùng đệm của tệp.
 - fflushall dùng để làm sạch vùng đệm của các tệp đang mở.

 - feof cho biết đã gặp cuối tệp hay chưa. rewind dùng để chuyển con trỏ chỉ vị về đầu tệp.
 - fseek dùng để di chuyển con trở chỉ vị đến bất kỳ vị trí trên tệp (hàm này chỉ nên dùng cho kiểu nhị phân). ftell cho biết vị trí hiện tại của con trỏ chỉ vị.

 - ferror cho biết có lỗi (khác 0) hay không lỗi (=0).
 - perror thông báo lỗi trên màn hình.
 - unlink và remove dùng để loại tệp trên đĩa



8.3. Các hàm cấp 2 (tiếp)

- Các hàm xuất nhập ký tự: dùng cho cả 2 kiểu
 - putc và fputc dùng để ghi ký tự lên tệp
 - getc và fgetc dùng để đọc ký tự từ tệp
- Các hàm xuất nhập theo kiểu văn bản:
 - fprintf dùng để ghi dữ liệu theo khuôn dạng lên tệp
 - fscanf dùng để đọc dữ liệu từ tệp theo khuôn dạng
 - fputs dùng để ghi một chuỗi ký tự lên tệp
 - fgets dùng để đọc một dãy ký tự từ tệp
- Các hàm xuất nhập theo kiểu nhị phân:
 - putw dùng để ghi một số nguyên (2 byte) lên tệp
 - getw dùng để đọc một số nguyên (2 byte) từ tệp
 - fwrite dùng để ghi một số mẩu tin lên tệp
 - fread dùng để đọc một số mẩu tin từ tệp



8.4. Đóng mở tệp, xóa vùng đệm và kiểm tra lỗi

- Dùng chung cho cả 2 kiểu nhị phân và văn
- 8.4.1. Hàm fopen: Mở tệp
 - Dạng hàm:

FILE *fopen(const char *tên_tệp, const char *kiểu)

- Các đối:
 - Đối thứ nhất là tên tệp (có thể có đường dẫn đầy
 - Đối thứ hai là kiểu truy nhập, có các giá trị sau:



8.4.1. Hàm fopen: Mở tệp

Kiểu	Ý nghĩa	
"r", "rt"	Mở 1 tệp để đọc theo kiểu văn bản. Tệp cần tồn tại nếu không sẽ có lỗi.	
"w", "wt"	Mở 1 tệp mới để ghi theo kiểu văn bản. Nếu tệp đã tồn tại, nó sẽ bị xóa.	
"a", "at"	Mở 1 tệp để ghi bổ sung theo kiểu văn bản. Nếu tệp chưa tồn tại thì tạo tệp mới.	
"rb"	Mở 1 tệp để đọc theo kiểu nhị phân. Tệp cần tồn tại néu không sẽ có lỗi.	
"wb"	Mở 1 tệp mới để ghi theo kiểu nhị phân. Nếu tệp đã tồn tại, nó sẽ bị xóa.	
"ab"	Mở 1 tệp để ghi bổ sung theo kiểu nhị phân. Nếu tệp chưa tồn tại thì tạo tệp mới.	



8.4.1. Hàm fopen: Mở tệp (tiếp)

Kiểu	Ý nghĩa	
"r+", "r+t"	Mở 1 tệp để đọc/ghi theo kiểu văn bản. Tệp cần tồn tại nếu không sẽ có lỗi.	
"w+", "w+t"	Mở 1 tệp mới để đọc/ghi theo kiểu văn bản. Nếu tệp đã tồn tại, nó sẽ bị xóa.	
"a+", "a+t"	Mở 1 tệp để đọc/ghi bổ sung theo kiểu văn bản. Nếu tệp chưa tồn tại thì tạo tệp mới.	
"r+b"	Mở 1 tệp để đọc/ghi theo kiểu nhị phân. Tệp cần tồn tại nết không sẽ có lỗi.	
"w+b"	Mở 1 tệp mới để đọc/ghi theo kiểu nhị phân. Nếu tệp đã tồr tại, nó sẽ bị xóa.	
"a+b"	Mở 1 tệp để đọc/ghi bổ sung theo kiểu nhị phân. Nếu tệp chưa tồn tại thì tạo tệp mới.	



8.4.1. Hàm fopen: Mở tệp (tiếp)

- Công dụng: hàm dùng để mở tệp. Nếu thành công, hàm trả về con trỏ kiểu FILE ứng với tệp vừa mở. Các hàm cấp 2 sẽ làm việc với tệp thông qua con trỏ này. Nếu có lỗi hàm trả về giá trị NULL.
- Chú ý: Trong các kiểu đọc/ghi, cần làm sạch vùng đệm trước khi chuyển từ đọc sang ghi hoặc ngược lại. Dùng các hàm fflush và di chuyển đầu từ.

19



8.4.2. Hàm fclose: đóng tệp

- Dang hàm: int fclose(FILE *f);
- Đối: f là con trỏ tương ứng với tệp cần đóng.
- Công dụng: hàm dùng để đóng tệp. Nội dung đóng tệp gồm:
 - đẩy dữ liệu còn trong vùng đệm lên đĩa (khi đang ghi)
 - xóa vùng đệm (khi đang đọc)
 - giải phóng biến f để nó có thể dùng cho tệp khác. Nếu thành công, hàm cho giá trị 0, trái lại hàm cho EOF.

20



8.4.3. Hàm fcloseall: đóng các tệp đang mở

- Dang hàm: int fcloseall(void);
- Công dụng: hàm dùng để đóng tất cả các tệp đang mở. Nếu thành công, hàm cho giá trị nguyên bằng số tệp đóng được, trái lai hàm cho EOF.

21



8.4.4. Hàm fflush: làm sạch vùng đệm

- Dang hàm: int fflush(FILE *f);
- Đối: f là con trỏ tệp
- Công dụng: hàm làm sạch vùng đệm của tệp f. Nếu thành công hàm cho giá trị 0, trái lai hàm cho EOF.

22



8.4.5. Hàm fflushall: làm sạch vùng đệm

- Dang hàm: int fflushall(void);
- Công dụng: hàm dùng làm sạch vùng đệm của các tệp đang mở. Nếu thành công hàm cho giá trị nguyên bằng số tệp đang mở, trái lai hàm cho EOF.

22



8.4.6. Hàm feof: kiểm tra cuối têp

- Dạng hàm int feof(FILE *f);
- Đối: f là con trỏ tệp
- Công dụng: hàm dùng để kiểm tra cuối tệp. Hàm cho giá trị khác 0 nếu gặp cuối tệp khi đọc, trái lại hàm cho giá trị 0.



8.4.7. Hàm ferror: kiểm tra lỗi

- Dang hàm: int ferror(FILE *f);
- Đối: f là con trỏ têp
- Công dụng: hàm dùng để kiểm tra lỗi thao tác trên tệp f. Hàm cho giá trị 0 nếu không lỗi, trái lại hàm cho giá trị khác 0.

__



8.4.8. Hàm perror: thông báo lỗi hệ thống

- Dang hàm: void perror(const char *s);
- Đối: s là con trỏ trỏ tới một chuỗi ký tự
- Công dụng: hàm in chuỗi s và thông báo lỗi

26



8.4.9. Hàm unlink: xóa tệp

- Dang hàm: int unlink(const char *tên têp);
- Đối: là tên tệp cần xóa
- Công dụng: hàm dùng để xóa 1 tệp trên đĩa. Nếu thành công, hàm cho giá trị 0, trái lại hàm cho giá trị EOF.

27



8.4.10. Hàm remove: xóa tệp

- Dang hàm: remove(const char *tên têp);
- Đối: là tên tệp cần xóa.
- Công dụng: hàm dùng để xóa một tệp trên đĩa. Nó là hàm macro gọi tới unlink.

28



8.4.11. Ví dụ: mở 1 tệp và kiểm tra lỗi

FILE *fp;

/*Mở tệp so_lieu để đọc theo kiểu nhị phân. Nếu thành công, con trỏ tệp so_lieu gán cho biến fp*/

fp = fopen("so_lieu","rb");

// Kiểm tra lỗi

if(fp==NULL) perror("Lỗi khi mở tệp so_lieu");

8.5. Nhập xuất ký tự

- Dùng được cả trong kiểu nhị phân và văn bản nhưng tác dụng khác nhau.
- 8.5.1. Hàm putc và fputc
 - Dạng hàm:
 - int putc(int ch, FILE *fp);
 int fputc(int ch, FILE *fp);
 - Đối: ch là một giá tị nguyên, fp là con trỏ tệp.
 - Công dụng: hàm ghi lễn tệp fp một ký tự có mã bằng: m = ch%256, trong đó ch được xem là số nguyên không dấu. Nếu thành công hàm cho mã ký tự được ghi, trái lại hàm cho EOF



8.5.1. Hàm putc và fputc (tiếp)

- Ví dụ: câu lệnh putc(-1,fp); sẽ ghi lên tệp fp mã 255 vì dạng không dấu của -1 là 65535.
- Ghi chú:
 - Hai hàm trên có ý nghĩa như nhau.
 - Trong kiểu văn bản, nếu m =10 thì hàm sẽ ghi lên tệp hai mã 13 và 10.

31



8.5.2. Hàm getc và fgetc

- Dạng hàm:
 - int getc(FILE *fp);int fgetc(FILE *fp);
- Đối: fp là con trỏ tệp
- Công dụng: hàm đọc 1 ký tự từ tệp fp. Nếu thành công hàm cho mã đọc được (có giá trị từ 0 đến 255). Nếu gặp cuối tệp hay có lỗi hàm cho EOF
- Ghi chú:
 - hai hàm trên có ý nghĩa như nhau
 - trong kiểu văn bản, hàm đọc một lượt cả hai mã 13, 10 và trả về giá trị 10; khi gặp mã 26 thì hàm không trả về 26 mà trả về EOF

32



8.5.3. Ví dụ

■ Chương trình sao tệp chế độ nhị phân và dùng hàm fgetc, fputc #include"stdiib.h"
void main(\{
 int c;
 char tt[14], t2[14];
 FILE *f1, *f2;
 printf("\nTEP NGUÔN:"); gets(t1);
 printf("\nTEP NGUÔN:"); gets(t2);
 f1=fopen(t1,"rb");
 if(f1==NULL){
 printf("\nTEP %s không tòn tại",t1); getch(); exit(1);
 }
 f2=fopen(t2,"wb");
 while((c=fgetc(f1))!=EOF) fputc(c,f2);
 fclose(f1); fclose(f2);

33



8.5.3. Ví dụ (tiếp)

- Chương trình trên thực hiện sao tệp theo thuật toán sau:
 - bước 1: đọc 1 ký tự của tệp f1, kết quả đặt vào biến c
 bước 2: nếu c bằng EOF thì kết thúc; nếu c khác EOF thì ghi c vào tệp f2 rồi quay trở lại bước 1.
- Nhận xét 1: nếu trong chương trình trên, ta thay bằng kiểu văn bản thì chỉ các byte đứng trước mã 26 đầu tiên của tệp f1 được sao sang tệp f2.
- Nhận xét 2: nếu dùng hàm feof và thuật toán:
 - bước 1: nếu feof(f1) khác 0 thì kết thúc, trái lại chuyển xuống bước 2.
 - bước 2: đọc 1 ký tự từ tệp f1, ghi lên tệp f2 thì ta có đoạn chương trình:

24



8.5.3. Ví dụ (tiếp)

while(!feof(f1)) fputc(fgetc(f1),f2);

- Đoạn chương trình này lại chưa thật đúng! Tệp f2 sẽ dài hơn tệp f1 đúng một byte có giá trị 255.
- Lý do: giả sử tệp f1 có đúng một ký tự mã 65, khi đó thuật toán sẽ diễn ra như sau:
 - bước 1: đầu từ đang trỏ vào ký tự A nên feof(f) = 0, chuyển xuống bước 2.
 - bước 2: đọc ký tự A của f1 và ghi lên f2, trở lại bước 1.
 - bước 1: đầu đọc đặt ở cuối tệp f1 nhưng chưa có thao tác đọc nên feof(f1) vẫn bằng 0, chuyên xuống bước 2.
 - bước 2: đọc một ký tự của f1. Khi đó nhận được -1. Ghi -1 lên f2 thì mã 255 sẽ được ghi. Ngoài ra, do khi đọc từ f1 gặp phải cuối tệp nên lúc này feof(f1) khác 0. Đến đây thuật toán kết thúc.

25



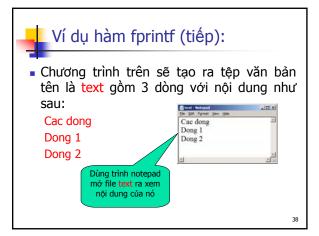
8.6. Các hàm nhập xuất theo kiểu văn bản

- 8.6.1. Hàm fprintf: ghi dữ liệu theo khuôn dạng
 - Dạng hàm:

int fprintf(FILE *f, const char *dk,...);

- Đối:
 - f là con trỏ tệp
 - dk chứa địa chỉ của chuỗi điều khiển
 - ... là danh sách các đối mà giá trị của chúng cần ghi lên tệp.
- Công dụng: giá trị các đối được ghi lên tệp f theo khuôn dạng xác định trong chuỗi dk. Nếu thành công hàm trả về một giá trị nguyên bằng số byte ghi lên tệp, nếu có lỗi thì trả về EOF.
- Nhận xét: Hàm làm việc giống hàm printf.

```
#include<stdio.h>
void main(){
   FILE *f;
   int i;
   f=fopen("text","wt");
   fprintf(f,"Cac dong");
   for(i=1;i<=2;i++) fprintf(f,"\nDong%2d",i);
   fclose(f);
}</pre>
```





8.6.2. Hàm fscanf: đọc dữ liệu từ tệp theo khuôn dạng

- Dạng hàm:
- int fscanf(FILE *f,const char *dk,...);
- Đối
 - f là con trỏ tệp
 - dk chứa địa chỉ của chuỗi điều khiển
 - ... là danh sách các đối sẽ chứa kết quả đọc được từ tệp
- Công dụng: đọc dữ liệu từ tệp f, biến đổi theo khuôn dạng trong dk và lưu kết quả vào các đối.
 Hàm trả về một giá trị bằng số trường được đọc.
- Nhận xét: Hàm làm việc giống hàm scanf.

39



Ví dụ 1 về hàm fscanf

 Giả sử có tệp văn bản "da_giac.sl" chứa thông tin về một đa giác. Tệp gồm n+1 dòng với nội dung như sau:

 Dòng 1:
 n
 (số đỉnh)

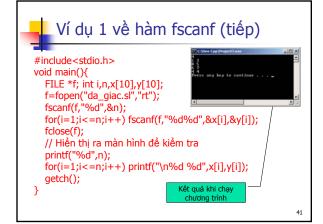
 Dòng 2:
 x1 y1 (tọa độ đỉnh 1)

 Dòng 3:
 x2 y2 (tọa độ đỉnh 2)

Dòng n+1: xn yn (tọa độ đỉnh n)

 Chương trình sau sẽ đọc số đỉnh và tọa độ các đỉnh từ tệp "da_giac.sl"

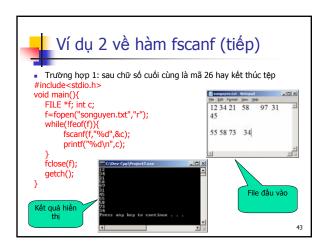
40

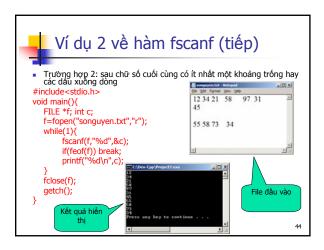


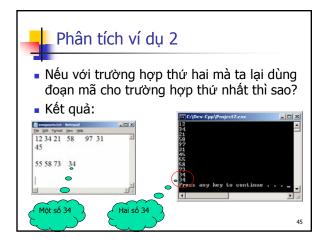


Ví du 2 về hàm fscanf

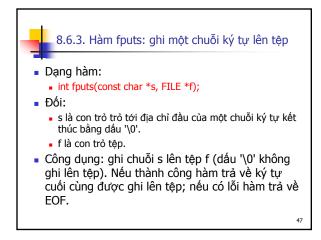
- Giả sử có một dãy số nguyên ghi trên tệp văn bản "songuyen.txt". Giữa hai số nguyên có ít nhất một khoảng trống hay các dấu xuống dòng. Yêu cầu đọc và in ra màn hình dãy số nói trên.
- Ta phân biệt 2 trường hợp:
 - Sau chữ số cuối cùng là mã 26 hay cuối tệp
 - Sau chữ số cuối cùng có ít nhất một khoảng trống hay các dấu xuống dòng.

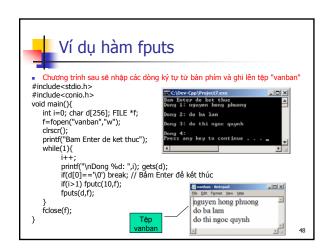














8.6.4. Hàm fgets: đọc một dãy ký tự từ tệp

- Dang hàm:
- *fgets(char *s, int n, FILE *f);
- s là con trỏ trỏ tới vùng nhớ đủ lớn để chứa chuỗi ký tự sẽ đọc từ tệp.
- n là số nguyên xác định độ dài cực đại của dãy cần đọc.
 f là con trỏ tệp.
- Cộng dụng: đốc 1 đãy ký tự từ tệp f chứa vào vùng nhớ s. Việc đọc kết thúc khi:

 - t thốc khi. hoặc đã đọc n-1 ký tự hoặc gập dấu xuống dòng (cặp mã 13 10). Khi đó mã 10 được đưa vào xâu kết quả.
 - hoặc kết thúc tệp.
- Xâu kết quả sẽ được bổ sung thêm dấu hiệu kết thúc chuỗi '\0'. Khi thành công hàm trả về địa chỉ vùng nhận kết quả; khi có lỗi hoặc gặp cuối tệp, hàm cho giá trị NULL.





8.7. Tệp văn bản và các thiết bị chuẩn

 Có thể dùng các hàm nhập xuất văn bản trên các thiết bị chuẩn. C đã định nghĩa các tệp tin và con trỏ tệp ứng với các thiết bị chuẩn như sau:

Tệp	Con trỏ	Thiết bị
in	stdin	Thiết bị vào chuẩn (bàn phím)
out	stdout	Thiết bị ra chuẩn (màn hình)
err	stderr	Thiết bị lỗi chuẩn (màn hình)
prn	stdprn	Thiết bị in chuẩn (máy in)

 Khi chương trình C bắt đầu làm việc thì các tệp này được tự động mở, vì vậy có thể dùng các con trỏ nêu trên để nhập xuất trên các thiết bị chuấn

8.7. ví du

```
#include<stdio.h>
#include < conio.h >
void main(){
  char ht[25]; float diem; int ns;
  printf("\nHo ten: ");fgets(ht,25,stdin);
  printf("\nDiem va nam sinh");
  fscanf(stdin,"%f%d",&diem,&ns);
  fputs(ht,stderr);
  fprintf(stdout,"Diem %f nam sinh %d",diem, ns);
```

52



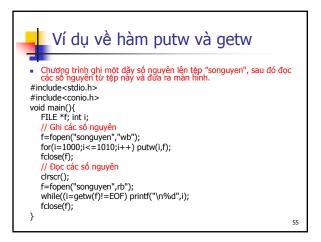
8.8. Các hàm nhập xuất theo kiểu nhị phân

- 8.8.1. Hàm putw: ghi một số nguyên
 - Dang hàm: int putw(int n, FILE *f);
 - Đối:
 - n là giá trị nguyên
 - f là con trỏ tệp
 - Công dụng: ghi giá trị n lên tệp f dưới dạng 2 byte. Nếu thành công hàm trả về số nguyên được ghi; nếu có lỗi hàm trả về EOF.

53

8.8.2. Hàm getw: đọc một số nguyên

- Dang hàm: int getw(FILE * f);
- Đối: f là con trỏ tệp.
- Công dung: đọc một số nguyên 2 byte từ tệp f. Nếu thành công, hàm trả về số nguyên đọc được; nếu có lỗi hoặc gặp cuối tệp, hàm trả về EOF.





8.8.3. Hàm fwrite: ghi các mẫu tin lên tệp

- Dạng hàm:
 - int fwrite(void *ptr, int size, int n, FILE *f);
- Đối:
 - ptr là con trỏ trỏ tới vùng nhớ chứa dữ liệu cần ghi.
 - size là kích thước của mẫu tin theo byte.
 - n là số mẫu tin cần ghi.
 - f là con trỏ tệp.

f2=fopen(t2,"wb");

 Công dụng: ghi n mẫu tin kích thước size byte từ vùng nhớ ptr lên tệp f. Hàm trả về giá trị bằng số mẫu tin thực sự được ghi.

56



8.8.4. Hàm fread: đọc các mẫu tin từ tệp tin

- Dang hàm:
 - int fread(void *ptr, int size, int n, FILE *f);
- Đối:
 - ptr là con trỏ trỏ tới vùng nhớ sẽ chứa dữ liệu đọc được từ tệp tin.
 - size là kích thước của mẫu tin theo byte.
 - n là số mẫu tin cần đọc.
 - f là con trỏ tệp.
- Công dụng: đọc n mẫu tin kích thước size byte từ tệp f chứa vào vùng nhớ ptr. Hàm trả về một giá trị bằng số mẫu tin thực sự đọc được.

57



58

4

Ví dụ về fwrite, fread

```
Ví du 2: ghi và đọc một dãy n phần tử số thực
#include<stdio.h>
#include<conio.h>
void main(){
    FILE *f; float a[20],b[20]; int i,n;
    // Nhập số phần tử n
    do{
        printf("Nhap so phan tu n= ");scanf("%d",&n);
    }while((n<1)||(n>20));
    // Nhập vào n phần tử thực
    for(i=0;i<n;i++){
        printf("\na[%d]= ",i); scanf("%f",&a[i]);
    }</pre>
```



Ví dụ về fwrite, fread

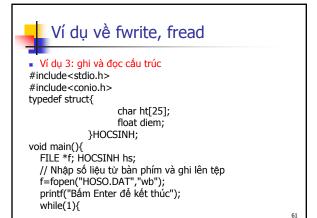
while((n=fread(c,1,1000,f1))>0) fwrite(c,1,n,f2); fclose(f1); fclose(f2);

```
f=fopen("mangsolieu","wb");

// Ghi n phần tử thực của mảng a vào file f
fwrite(a,sizeof(float),n,f);
fclose(f);
f=fopen("mangsolieu","rb");

// Đọc n phần tử thực từ file f đưa vào mảng b
fread(b,sizeof(float),n,f);

// Hiển thị ra màn hình
for(i=0;i<n;i++)
    printf("\nb[%d]=%f",i,b[i]);
fclose(f);
getch();
}
```







8.9. Nhập xuất ngẫu nhiên và các hàm di chuyển con trỏ chỉ vi

- Mỗi tệp khi đang mở có một con trỏ chỉ vị dùng để xác định vị trí đọc/qhi trên tệp.
- Khi mở tệp tin để đọc/ghi, con trỏ chỉ vị luôn ở đầu tệp tin. Nhưng nếu mở theo chế độ "a" thì con trỏ chỉ vị ở cuối tệp để ghi thêm dữ liệu vào têp.
- Việc xuất nhập dữ liệu được thực hiện từ vị trí hiện tại của con trỏ chỉ vị và sau khi hoàn thành thì con trỏ này dịch chuyển đi một số byte bằng số byte đã đọc hay ghi.
- Việc xuất nhập được tiến hành tuần tự từ đầu đến cuối tệp tin.

63



8.9.1. Hàm rewind: chuyển con trỏ chỉ vị về đầu têp

- Dang hàm: void rewind(FILE *f);
- Đối: f là con trỏ tệp.
- Công dụng: chuyển con trỏ chỉ vị của tệp f về đầu tệp. Khi đó, việc nhập xuất trên tệp f được thực hiện từ đầu tệp.

64



8.9.2. Hàm fseek: di chuyển con trỏ chỉ vị đến vị trí mong muốn

- Dạng hàm:
 - int fseek(FILE *f, long sb, int xp);
- Đối:
 - f là con trỏ tệp.
 - sb là số byte cần di chuyển.
 - xp cho biết vị trí xuất phát mà việc dịch chuyển được bắt đầu từ đấy. xp cps thể nhận các giá trị sau:
 - SEEK_SET hay 0: xuất phát từ đầu tệp.
 - SEEK_CUR hay 1: xuất phát từ vị trí hiện tại của con trỏ chỉ vị.
 - SEEK_END hay 2: xuất phát từ cuối tệp.

c E



8.9.2. Hàm fseek (tiếp)

- Công dụng: hàm di chuyển con trỏ chỉ vị của tệp f từ vị trí xác định bởi xp qua một số byte bằng giá trị tuyệt đối của sb. Chiều di chuyển về cuối tệp nếu sb dương, trái lại di chuyển về đầu tệp. Khi thành công, hàm trả về giá trị 0; nếu có lỗi hàm trả về giá trị khác 0.
- Chú ý: Không nên dùng fseek trên kiểu văn bản.



8.9.3. Hàm ftell: cho biết vị trí hiện tại của con trỏ chỉ vi

- Dang hàm: long ftell(FILE *f);
- Đối: f là con trỏ tệp.
- Công dụng: khi thành công, hàm cho biết vị trí hiện tại của con trỏ chỉ vị (byte thứ mấy trên tệp f). Số thứ tự byte được tính từ 0. Khi có lỗi, hàm trả về -1L.

67





Bài tập

Bài 1:

Viết chương trình:

- Nhập tử bàn phím N số thực lưu vào một mảng (N 100 và N được nhập từ bàn phím).
- Sau đó ghi ra một file văn bản có tên là "float.txt" theo quy cách: dòng đầu tiên lưu số lượng các số thực, các dòng tiếp theo lưu các số thực, mỗi số lưu trên một dòng.
- Đọc lại tệp văn bản đó và lưu các số thực đọc được vào một màng.
- Sắp xếp các số thực trong mảng theo thứ tự tăng dần và ghi ra một tệp văn bản khác có tên là "floatsx.txt" theo quy cách giống như tệp "float.txt".

69



Bài tập

- Bài 2: Viết chương trình ghép nối nội dung 2 file:
 - Nhập vào từ bàn phím 2 xâu kí tự là đường dẫn của file nguồn và file đích
 - Ghép nội dung của file nguồn vào cuối file đích.



