



Giới thiệu tổng quan về lập trình

Nhập môn lập trình

Trình bày: ...; Email: ...@fit.hcmus.edu.vn

Nội dung

- Khái niệm về chương trình máy tính
- Các ngôn ngữ lập trình
- Các khái niệm cơ bản về lập trình
- Các vấn đề tìm hiểu mở rộng kiến thức nghề nghiệp
- Thuật ngữ và bài đọc thêm tiếng Anh



Khái niệm về chương trình máy tính



Chương trình mã máy

- Khái niệm
 - CPU của máy tính được thiết kế để có thể thực hiện được các chương trình mã máy (machine code program) đã được hệ điều hành (HĐH) nạp vào RAM của máy tính.
 - Chương trình mã máy thường phải tương thích với từng họ máy cụ thể, bao gồm tập hợp các chỉ thị được viết bằng các lệnh CPU của họ máy đó, được lưu trên đĩa dưới dạng một tập tin mã thực thi (executable program file) của HĐH cụ thể.



Chương trình mã máy

- Qui trình thực hiện
 - B1. Người sử dụng (người dùng cuối – end user) ra lệnh thực hiện (chạy) chương trình.
 - B2. HĐH nhận được lệnh sẽ thực hiện:
 - Tìm và nạp tập tin mã thực thi của chương trình (nằm trên đĩa) vào RAM của máy tính.
 - Bộ đếm lệnh của CPU (CPU program counter) được trỏ đến lệnh đầu tiên của chương trình (còn gọi là ngõ vào chương trình – program entry point)



Chương trình mã máy

- Qui trình thực hiện (tiếp theo)
 - B3. CPU thực hiện từng chỉ thị một trong RAM cho đến khi gặp lệnh kết thúc:
 - Chép lệnh mã máy hiện hành vào thanh ghi lệnh.
 - Tăng bộ đếm lệnh (để trỏ đến lệnh kế tiếp).
 - Thi hành lệnh mã máy.
 - B4. Kết thúc thực hiện chương trình, HĐH chờ nhận lệnh mới.



Chương trình mã máy

- Đặc điểm
 - Mỗi chỉ thị của chương trình là một lệnh mã máy (một dãy các byte chỉ phù hợp với qui ước tập lệnh của một loại CPU nào đó)
 - Được cấu trúc hóa theo qui ước của HĐH.
 - Được chạy trên một họ CPU và HĐH cụ thể.
 - Nội dung rất khó hiểu đối với người dùng máy tính, chỉ có CPU thích hợp với hiểu rõ và thi hành được.



Chương trình mã máy

- Nhận xét
 - Khó có thể sản xuất ra phần mềm bằng cách viết trực tiếp các chương trình mã máy.
 - Nếu có làm được theo cách này thì:
 - Giá cả sẽ rất đắt do quá khó, tốn quá nhiều thời gian và công sức.
 - Khả năng dùng lại rất giới hạn do không thể bán cho người dùng trên họ máy tính khác hay người dùng sử dụng hệ điều hành khác.



Chương trình nguồn

- Khái niệm
 - Việc viết các chương trình mã máy rất cực và kém hiệu quả ngay cả đối với các lập trình viên chuyên nghiệp vì vậy giải pháp khởi đầu là sử dụng các NNLT cấp thấp như hợp ngữ hoặc các NNLT cấp cao (sẽ bàn ở phần sau).
 - Chương trình viết bằng NNLT được gọi là chương trình nguồn (source code program) hay mã nguồn (source code).



Dịch chương trình

- Chương trình nguồn được dịch sang chương trình mã máy bằng cách chương trình dịch:
 - Trình hợp dịch (assembler) để dịch các chương trình hợp ngữ.
 - Trình thông dịch (interpreter) và trình biên dịch (compiler) để dịch các chương trình cấp cao.



Các ngôn ngữ lập trình



Khái niệm

- Ngôn ngữ lập trình (programming language) là ngôn ngữ được lập trình viên sử dụng để viết chương trình cho máy tính.
- Khi một chương trình được viết bằng một NNLT nào đó thì các chỉ thị, câu lệnh trong chương trình phải tuân theo các qui tắc, các luật do NNLT đó qui định.



NNLT cấp thấp

- Đặc điểm
 - Là NNLT phụ thuộc vào từng họ máy cụ thể, vì vậy không có tính tương thích.
 - Dễ viết, đọc, sửa hơn chương trình mã máy.
 - Ưu điểm là tận dụng và khai thác được tính năng của mỗi họ máy cụ thể, nhờ vậy chương trình có thể chạy nhanh hơn.



NNLT cấp cao

- Đặc điểm
 - Được đề xuất để khắc phục các hạn chế của NNLT cấp thấp.
 - Dễ dùng và dễ diễn đạt được các ý tưởng trừu tượng.
 - Có tính tương thích cao (khi thay đổi dạng máy tính thì chỉ cần sửa chương trình rất ít hoặc thậm chí không cần sửa mà vẫn đảm bảo chạy đúng).

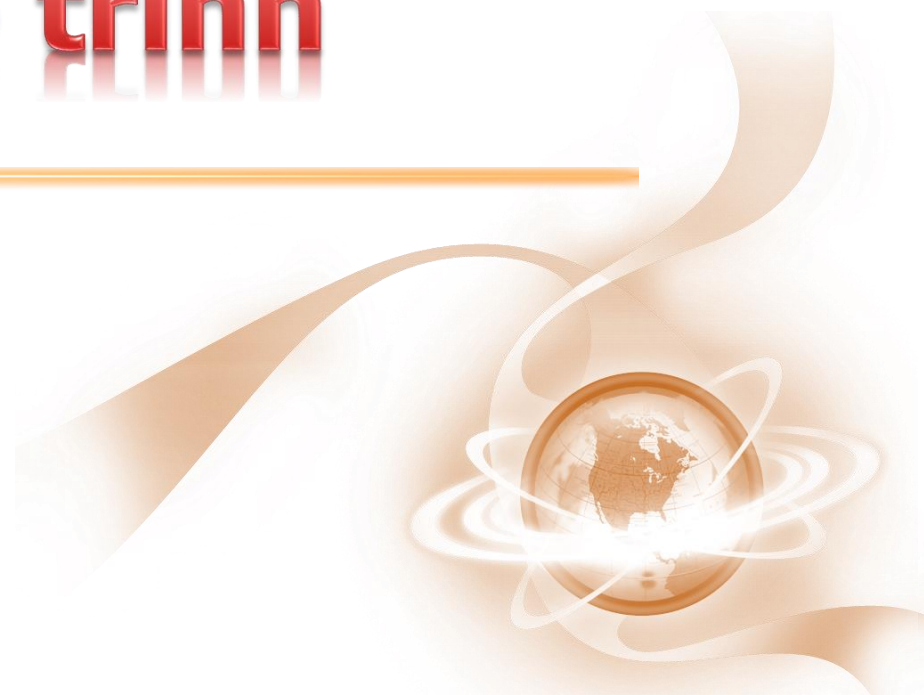


Một vài NNLT thông dụng

- NNLT cấp thấp
 - Hợp ngữ (assembly language)
- NNLT cấp cao
 - C/C++
 - COBOL
 - FORTRAN
 - Java, C#
 - PHP, Ruby, Perl
 - Ada, BASIC, Visual Basic (VB), Lisp, Pascal, ...



Các khái niệm cơ bản về lập trình



Khái niệm cơ bản

- Một chương trình (program) là một dãy các chỉ thị (instruction) điều khiển sự hoạt động của máy tính nhằm giải quyết một công việc nào đó.
- Người viết chương trình (còn gọi là lập trình viên hay thảo chương viên – programmer) là những người tạo lập ra những chương trình máy tính.



Chương trình minh họa

- Hai chương trình đơn giản sau chỉ in ra một dòng chữ có nội dung là "Hello everybody!" bằng NNLT Java và C.

	Chương trình Java	Chương trình C
1	<code>// Hello.java</code>	<code>/* Hello.c */</code>
2	<code>import java.util.*;</code>	<code>#include <stdio.h></code>
3	<code>public class Hello {</code>	
4	<code> public static void main(String argv[])</code>	<code>void main(void)</code>
5	<code>{</code>	<code>{</code>
6	<code> System.out.print("Hello everybody!");</code>	<code> printf("Hello everybody!");</code>
7	<code>}</code>	<code>}</code>
8	<code>}</code>	



Viết – dịch – chạy chương trình

- Đối với các NNLT cấp cao truyền thống (trước thế hệ của Java và C#), quá trình viết, dịch và chạy chương trình gồm các công đoạn như sau:
 - B1. Soạn chương trình nguồn và lưu lên đĩa.
 - B2. Dịch chương trình nguồn nhờ trình biên dịch.
 - B3. Nối kết các tập tin mã trung gian tạo ra ở B2.
 - B4. Chạy chương trình ngôn ngữ máy tạo ra ở B3.



Các vấn đề tìm hiểu mở rộng kiến thức nghề nghiệp



Công nghệ lập trình hiện đại

- Hạn chế của các chương trình cấp cao truyền thống là trình biên dịch của chúng phát sinh trực tiếp mã thực thi phụ thuộc vào mã máy tính của một họ máy tính và hệ điều hành cụ thể nên không thể mang đi sử dụng ở các hệ điều hành khác.



Công nghệ lập trình hiện đại

- NNLT hiện đại như Java hay C# trình biên dịch không dịch trực tiếp mã nguồn thành mã thực thi mà được thiết kế để có thể dịch thành mã thực thi trừu tượng (abstract executable code) độc lập máy và hệ điều hành.



Công nghệ lập trình hiện đại

- Do máy tính thật không thể hiểu được mã trừu tượng nên những chương trình dạng mã thực thi trừu tượng chỉ chạy được khi có sẵn máy ảo hỗ trợ cho việc thi hành loại mã thực thi đó.
 - Chương trình nguồn Java (tập tin *.java) được dịch thành mã thực thi không phụ thuộc máy tính (tập tin *.class) có thể chạy được trên bất kỳ máy tính nào đã cài đặt máy ảo Java (Java Virtual Machine – JVM)



Công nghệ lập trình hiện đại

- Trong các năm gần đây, các ứng dụng chạy trên web phát triển rất mạnh.
 - Chạy trên internet thông qua một trình duyệt web.
 - Được viết bằng các ngôn ngữ như PHP, ASP.NET, JSP, Java Script, VB Script... có tính tương thích cao, hoạt động trên bất kỳ máy tính nào có internet



Môi trường lập trình

- Toàn bộ qui trình biên dịch được thực một cách dễ dàng và thuận tiện nhờ vào công cụ gọi là môi trường phát triển phần mềm (Integrated Development Environment – IDE)
 - Soạn thảo chương trình.
 - Quản lý hệ thống tập tin mã nguồn.
 - Quản lý hệ thống các phiên bản của mã nguồn.
 - Kiểm tra lỗi cú pháp, biên dịch, liên kết chương trình.
 - Chạy từng dòng lệnh (debug) để tìm lỗi.



Môi trường lập trình

- Một số IDE thông dụng:
 - Eclipse: hỗ trợ nhiều ngôn ngữ.
 - C++ Visual Studio: ngôn ngữ C++.
 - C# Visual Studio: ngôn ngữ C#.
 - Visual Café: ngôn ngữ Java.
 - J Builder: ngôn ngữ Java.



© J-Builder Sdn Bhd 2009. All Rights Reserved.



Thuật ngữ và bài đọc thêm tiếng Anh



Thuật ngữ tiếng Anh

- ***abstract executable code***: mã trừu tượng
- ***assembler***: trình hợp dịch
- ***assembly language***: hợp ngữ
- ***compiler***: trình biên dịch
- ***data type***: kiểu dữ liệu
- ***debug***: chạy chương trình theo từng dòng lệnh để tìm lỗi
- ***executable program file***: một tập tin mã thực thi
- ***end user(s)***: người sử dụng, người dùng cuối
- ***IDE***: viết tắt của "*Integrated Development Environment*", môi trường phát triển chương trình tích hợp
- ***instruction***: chỉ thị
- ***interpreter***: trình thông dịch
- ***link***: nối kết các mã trung gian
- ***linker*** (hay ***link program***): chương trình liên kết mã trung gian



Thuật ngữ tiếng Anh

- **machine code program**: chương trình mã máy
- **object code**: mã đối tượng, một loại mã trung gian chưa phải là mã máy thật sự
- **program entry point**: ngõ vào chương trình
- **program**: chương trình
- **programmer**: người viết chương trình, lập trình viên, từ cũ: “thảo chương viên”
- **programming language**: ngôn ngữ lập trình
- **low-level programming language**: ngôn ngữ lập trình cấp thấp
- **high-level programming language**: ngôn ngữ lập trình cấp cao
- **programming**: lập trình
- **source code program**: chương trình nguồn
- **source code**: mã nguồn
- **syntax error**: lỗi cú pháp
- **text editor**: trình soạn thảo văn bản (có thể dùng để soạn mã nguồn)



Bài đọc thêm tiếng Anh

- **Thinking in C**, Bruce Eckel, E-book, 2006.
- **Theory and Problems of Fundamentals of Computing with C++**, John R. Hubbard, Schaum's Outlines Series, McGraw-Hill, 1998.



