TRANSLATION UNIT 13 + UNIT 14

**UNIT 13:**

THIẾT KẾ TRANG WEB

HTML và trình soạn thảo web

Mã đằng sau các trang web là HTML (Ngôn ngữ đánh dấu siêu văn bản) bao gồm các lệnh gọi thẻ. Thẻ được đặt xung quanh các đoạn văn bản để cho trình duyệt web biết cách hiển thị văn bản hoặc hình ảnh. Bạn có thể xem mã HTML nguồn của trang web bằng cách chọn tùy chọn Nguồn trang trong trình duyệt web của mình. Nhưng bạn không cần phải học HTML để xây dựng trang web của riêng mình. Thay vào đó, bạn có thể sử dụng trình xử lý văn bản có khả năng thiết kế web hoặc trình soạn thảo web chuyên dụng như Macromedia Dreamweaver hoặc Microsoft FrontPage. Trình soạn thảo web thân thiện với người dùng và WYSIWYG (Những gì bạn thấy là những gì bạn nhận được). Các nút và mục menu khác nhau cho phép bạn thiết kế trang mà không cần viết HTML.

Các tệp HTML có cấu trúc cơ bản sau:

Một số mã nguồn HTML cơ bản Tệp HTML hiển thị dưới dạng trang web

Các thành phần của trang web

Có một số thành phần khác nhau mà bạn có thể sử dụng trên trang web:

- Văn bản – được hiển thị bằng nhiều phông chữ và kích cỡ khác nhau. Hầu hết các tệp văn bản đều có hai định dạng: HTML hoặc PDF (định dạng tài liệu di động có thể xem bằng Acrobat Reader).

- Nền - màu sắc và hoa văn cơ bản của trang web.

- Bảng - có cột và hàng, dùng để định vị hình ảnh và văn bản trên một trang.

- Khung - các vùng hình chữ nhật cho phép hiển thị các trang khác nhau trong cùng một cửa sổ trình duyệt.

- Cascading Style Sheets (CSS) - một cơ chế để thêm kiểu vào tài liệu web, Bạn có thể sử dụng mã HTML để chỉ định phông chữ, kiểu văn bản và màu nền. Tuy nhiên, ngày nay, CSS phổ biến hơn. Điều này giúp dễ dàng áp dụng các thay đổi về cách trình bày trên toàn bộ trang web.

- Đồ họa, clip nghệ thuật, biểu tượng, mẫu nền, hình nền và hình ảnh trong suốt - các định dạng phổ biến là .jpg (nhóm chuyên gia nhiếp ảnh chung), lý tưởng cho hình ảnh có nhiều màu sắc, .gif (định dạng trao đổi đồ họa) lý tưởng cho hình ảnh có ít màu hơn và .png (Đồ họa mạng di động) hỗ trợ 16 triệu màu.

- Siêu liên kết – văn bản hoặc hình ảnh được tô sáng (nút, bản đồ hình ảnh, v.v.) đóng vai trò là liên kết đến các trang khác. Nếu bạn muốn chia sẻ thông tin với mọi người, bạn có thể sử dụng nguồn cấp RSS và cung cấp cho người đọc liên kết đến nguồn cấp. RSS cho phép người đăng ký nhận thông tin cập nhật về blog, tin tức, podcast, v.v. Trước khi phát trực tiếp, bạn nên kiểm tra xem tất cả các liên kết có hoạt động không.

Âm thanh, video và hoạt hình

Nhiều trang web hiện nay kết hợp các tệp âm thanh và nếu bạn đang thiết kế một trang web, bạn có thể chèn các bài hát, podcast, v.v. Các định dạng âm thanh phổ biến nhất là: .wav (định dạng âm thanh sóng Windows).ra (tệp âm thanh thực) và .mp3 (Lớp âm thanh MPEG-1-3).

Video chuyển động đầy đủ được lưu trữ ở các định dạng sau: .avi (âm thanh video xen kẽ), .mov (phim QuickTime) và .mpg (nhóm chuyên gia về hình ảnh chuyển động).

Nếu bạn muốn đưa một cái gì đó đặc biệt vào các trang web của mình, bạn có thể sử dụng Adobe Flash để đưa vào hoạt ảnh tương tác và âm thanh phát trực tuyến. Ngoài ra, bạn có thể chèn các applet Java - các chương trình nhỏ cho phép tạo các tệp tương tác. Hoạt hình được tạo thành từ một loạt các hình ảnh độc lập được sắp xếp theo trình tự để trông giống như hình ảnh chuyển động. Để xem hoặc nghe tất cả các tệp này, bạn phải có đúng plug-in, một chương trình phụ trợ mở rộng khả năng của trình duyệt web của bạn.

**UNIT 14:**

14.1 Giới thiệu về lập trình

Một chương trình là một tập hợp các lệnh yêu cầu máy tính thực hiện nhiều việc khác nhau; đôi khi lệnh mà máy tính phải thực hiện phụ thuộc vào những gì đã xảy ra khi máy tính thực hiện lệnh trước đó. Phần này cung cấp tổng quan về hai cách chính mà bạn có thể đưa ra các lệnh này, hay còn gọi là "lệnh". Một cách sử dụng trình thông dịch, cách còn lại là trình biên dịch. Vì ngôn ngữ của con người quá khó để máy tính có thể hiểu theo cách rõ ràng, nên các lệnh thường được viết bằng một hoặc nhiều ngôn ngữ được thiết kế riêng cho mục đích này.

Trình thông dịch dịch các lệnh cấp cao thành dạng trung gian, sau đó thực thi. Ngược lại, trình biên dịch dịch các lệnh cấp cao trực tiếp thành ngôn ngữ máy. Các chương trình biên dịch thường chạy nhanh hơn các chương trình được thông dịch. Tuy nhiên, ưu điểm của trình thông dịch là không cần phải trải qua giai đoạn biên dịch trong đó các lệnh máy được tạo ra. Quá trình này có thể tốn thời gian nếu chương trình dài. Ngược lại, trình thông dịch có thể thực thi ngay các chương trình cấp cao. Vì lý do này, đôi khi người ta sử dụng trình thông dịch trong quá trình phát triển chương trình, khi một lập trình viên muốn thêm các phần nhỏ cùng một lúc và kiểm tra chúng một cách nhanh chóng. Ngoài ra, trình thông dịch thường được sử dụng trong giáo dục vì chúng cho phép học sinh lập trình tương tác. Ngôn ngữ lập trình là ngôn ngữ được xây dựng chính thức được thiết kế để truyền đạt hướng dẫn cho máy móc, đặc biệt là máy tính. Ngôn ngữ lập trình có thể được sử dụng để tạo chương trình nhằm kiểm soát hành vi của máy móc hoặc để thể hiện thuật toán.

14.1.1. Trình thông dịch

Với trình thông dịch, ngôn ngữ đóng vai trò như một môi trường, nơi bạn nhập lệnh tại dấu nhắc và môi trường sẽ thực thi chúng thay bạn. Đối với các chương trình phức tạp hơn, bạn có thể nhập lệnh vào tệp và yêu cầu trình thông dịch tải tệp và thực thi lệnh trong đó. Nếu có bất kỳ sự cố nào xảy ra, nhiều trình thông dịch sẽ đưa bạn vào trình gỡ lỗi để giúp bạn theo dõi sự cố.

Ưu điểm của việc này là bạn có thể thấy kết quả của lệnh ngay lập tức và có thể dễ dàng sửa lỗi. Nhược điểm lớn nhất là khi bạn muốn chia sẻ chương trình của mình với ai đó. Họ phải có cùng một trình thông dịch, hoặc bạn phải có cách nào đó để cung cấp cho họ, và họ cần hiểu cách sử dụng nó. Ngoài ra, người dùng có thể không thích bị ném vào trình gỡ lỗi nếu họ nhấn nhầm phím! Về mặt hiệu suất, trình thông dịch có thể sử dụng rất nhiều bộ nhớ và thường không tạo mã hiệu quả bằng trình biên dịch.

Ngôn ngữ được thông dịch là cách tốt nhất để bắt đầu nếu bạn chưa từng lập trình trước đây. Loại môi trường này thường được tìm thấy trong các ngôn ngữ như Lisp, Smalltalk, Perl và Basic. Người ta cũng có thể lập luận rằng shell UNIX® (sh, csh) tự nó là một trình thông dịch và nhiều người thực sự viết "tập lệnh" shell để hỗ trợ nhiều tác vụ "quản lý" khác nhau trên máy của họ. Thật vậy, một phần trong triết lý UNIX ban đầu là cung cấp nhiều chương trình tiện ích nhỏ có thể được liên kết với nhau trong các tập lệnh shell để thực hiện các tác vụ hữu ích.

14.1.2. Trình biên dịch

Trình biên dịch khá khác biệt. Trước hết, bạn viết mã của mình trong một tệp (hoặc các tệp) bằng trình soạn thảo. Sau đó, bạn chạy trình biên dịch và xem liệu nó có chấp nhận chương trình của bạn không. Nếu nó không biên dịch, hãy nghiến răng và quay lại trình soạn thảo; nếu nó biên dịch và cung cấp cho bạn một chương trình, bạn có thể chạy nó tại dấu nhắc lệnh shell hoặc trong trình gỡ lỗi để xem nó có hoạt động bình thường không.

Rõ ràng, điều này không hoàn toàn trực tiếp như sử dụng trình thông dịch. Tuy nhiên, nó cho phép bạn thực hiện nhiều việc rất khó hoặc thậm chí là không thể thực hiện được bằng trình thông dịch, chẳng hạn như viết mã tương tác chặt chẽ với hệ điều hành hoặc thậm chí là viết hệ điều hành của riêng bạn! Nó cũng hữu ích nếu bạn cần viết mã rất hiệu quả, vì trình biên dịch có thể dành thời gian và tối ưu hóa mã, điều này sẽ không được chấp nhận trong trình thông dịch. Hơn nữa, việc phân phối một chương trình được viết cho trình biên dịch thường đơn giản hơn so với chương trình được viết cho trình thông dịch - bạn chỉ cần cung cấp cho họ một bản sao của tệp thực thi, giả sử họ có cùng hệ điều hành với bạn Các ngôn ngữ biên dịch bao gồm Pascal, C và C++. C và C++ là những ngôn ngữ khá khó khăn và phù hợp nhất với những lập trình viên có nhiều kinh nghiệm hơn; Mặt khác, Pascal được thiết kế như một ngôn ngữ giáo dục và là một ngôn ngữ khá tốt để bắt đầu. FreeBSD không bao gồm hỗ trợ Pascal trong hệ thống cơ sở, nhưng Free Pascal Compiler có sẵn trong Ports Collection dưới dạng lang/fpc.

Vì chu trình edit-compile-run-debug khá tẻ nhạt khi sử dụng các chương trình riêng biệt, nên nhiều nhà sản xuất trình biên dịch thương mại đã tạo ra Integrated Development Environments (viết tắt là IDE). FreeBSD không bao gồm IDE trong hệ thống cơ sở, nhưng devel/kdevelop có sẵn trong Ports Collection và nhiều người sử dụng Emacs cho mục đích này.

14.2 Ngôn ngữ lập trình

Có hàng trăm ngôn ngữ lập trình đang được sử dụng hiện nay. Làm sao bạn biết nên học ngôn ngữ nào trước? Làm sao bạn biết ngôn ngữ nào là tốt nhất cho lĩnh vực CNTT mà bạn lựa chọn? Vâng, tôi không thể trả lời câu hỏi đó thay bạn. Nhưng tại sao không bắt đầu bằng cách học một trong 10 ngôn ngữ phổ biến nhất? Bằng cách đó, bạn sẽ luôn có thể có được việc làm trong ngành CNTT.

Học ngôn ngữ lập trình không dễ, nhưng nó có thể rất bổ ích. Bạn sẽ có rất nhiều câu hỏi lúc đầu. Chỉ cần nhớ tìm kiếm sự trợ giúp khi bạn cần! Ngày nay, bạn có thể tìm thấy câu trả lời cho hầu hết mọi thứ trên Google.... vì vậy không có lý do gì để thất bại. Ngoài ra, hãy nhớ rằng phải mất nhiều năm để trở thành một lập trình viên chuyên nghiệp. Đừng mong đợi trở nên giỏi chỉ sau một đêm. Chỉ cần tiếp tục học một cái gì đó mới mỗi ngày và cuối cùng bạn sẽ đủ năng lực để hoàn thành công việc

14.2.1. Java

Java sử dụng trình biên dịch và là ngôn ngữ hướng đối tượng được Sun Microsystems phát hành vào năm 1995. Java là ngôn ngữ lập trình số một hiện nay vì nhiều lý do. Đầu tiên, đây là ngôn ngữ được tổ chức tốt với thư viện mạnh mẽ các thành phần phần mềm có thể tái sử dụng. Thứ hai, các chương trình được viết bằng Java có thể chạy trên nhiều kiến ​​trúc máy tính và hệ điều hành khác nhau do sử dụng JVM (máy ảo Java). Đôi khi điều này được gọi là tính di động của mã hoặc thậm chí là WORA (viết một lần, chạy ở mọi nơi). Thứ ba, Java là ngôn ngữ có nhiều khả năng được giảng dạy nhất trong các lớp khoa học máy tính tại trường đại học. Rất nhiều sách lý thuyết khoa học máy tính được viết trong thập kỷ qua sử dụng Java trong các ví dụ về mã. Vì vậy, việc học cú pháp Java là một ý tưởng hay ngay cả khi bạn chưa bao giờ thực sự viết mã bằng ngôn ngữ này.

Điểm mạnh của Java: tính di động, tính phổ biến

Điểm yếu của Java: Chậm hơn các ngôn ngữ được biên dịch gốc

14.2.2. C

C là ngôn ngữ biên dịch, thủ tục được Dennis Ritchie phát triển vào năm 1972 để sử dụng trong hệ điều hành UNIX. Mặc dù được thiết kế để có tính di động, các chương trình C phải được biên dịch riêng cho các máy tính có kiến ​​trúc và hệ điều hành khác nhau. Điều này giúp chúng nhanh như chớp.

Mặc dù C là ngôn ngữ tương đối cũ, nhưng nó vẫn được sử dụng rộng rãi để lập trình hệ thống, viết các ngôn ngữ lập trình khác và trong các hệ thống nhúng. Điểm mạnh: Tốc độ

Điểm yếu: Quản lý bộ nhớ có thể khó thành thạo

14.2.3. C++

C++ là ngôn ngữ biên dịch, Đa mô hình được Bjarne Stroustrup viết dưới dạng bản cập nhật cho C vào năm 1979. Ngôn ngữ này cố gắng tương thích ngược với C và mang lại hướng đối tượng, giúp ích cho các dự án lớn hơn. Mặc dù đã cũ, C++ vẫn được sử dụng để tạo ra nhiều ứng dụng từ trò chơi đến bộ ứng dụng văn phòng.

Điểm mạnh: Tốc độ

Điểm yếu: C++ cũ hơn và được coi là vụng về hơn các ngôn ngữ hướng đối tượng mới hơn như Java hoặc C#.

14.2.4. PHP

PHP sử dụng trình thông dịch thời gian chạy và là ngôn ngữ đa mô hình ban đầu được Ramus Lerdorf phát triển vào năm 1996 để tạo các trang web động. Lúc đầu, nó thậm chí không phải là ngôn ngữ lập trình thực sự, nhưng theo thời gian, cuối cùng nó đã phát triển thành ngôn ngữ lập trình hướng đối tượng có đầy đủ tính năng. Mặc dù PHP đã bị chỉ trích nhiều trong quá khứ vì hơi cẩu thả và không an toàn, nhưng nó đã khá tốt kể từ khi phiên bản 5 ra mắt vào năm 2004. Thật khó để tranh luận về thành công. Ngày nay, PHP là ngôn ngữ phổ biến nhất được sử dụng để viết các ứng dụng web. Ngay cả English 4IT, chương trình bạn đang sử dụng hiện tại, cũng được viết bằng PHP

Điểm mạnh: Lập trình web, tài liệu hướng dẫn tốt

Điểm yếu: Cú pháp không nhất quán, quá nhiều cách để thực hiện cùng một việc, lịch sử các quyết định bảo mật kỳ lạ

14.2.5. VB (hay Visual Basic)

Visual Basic là ngôn ngữ được thông dịch, đa mô hình do Microsoft Corporation phát triển cho nền tảng Windows. Ngôn ngữ này đã phát triển qua nhiều năm và được coi là hậu duệ trực tiếp của BASIC cũ của Microsoft từ những năm 1970. Visual Basic là ngôn ngữ tốt để viết kịch bản cho các ứng dụng Windows không cần sức mạnh và tốc độ của C#.

Điểm mạnh: Không có.

Điểm yếu: Chỉ chạy trên Windows

14.2.6. Python

Python là ngôn ngữ lập trình đa mô hình được Guido van Rossum viết vào cuối những năm 1980 và dành cho mục đích lập trình chung. Python không được đặt tên theo loài rắn mà thực chất là theo tên nhóm hài Monty Python. Python được đặc trưng bởi việc sử dụng thụt lề để dễ đọc và khuyến khích viết mã thanh lịch bằng cách khiến các nhà phát triển làm những việc tương tự theo những cách tương tự. Python được sử dụng làm lựa chọn lập trình chính của cả Google và Ubuntu.

Điểm mạnh: Khả năng đọc tuyệt vời và triết lý chung

Điểm yếu: Không có

14.2.7. C#

C# là ngôn ngữ hướng đối tượng được biên dịch do Microsoft viết. Đây là một thông số kỹ thuật mở, nhưng hiếm khi thấy trên bất kỳ nền tảng nào không phải Windows. C# được coi là ngôn ngữ cao cấp của Microsoft trong NET Framework. Nó rất giống với Java về cả cú pháp và bản chất.

Điểm mạnh: Mạnh mẽ và khá nhanh

Điểm yếu: Chỉ thực sự phù hợp với Windows

14.2.8. JavaScript

JavaScript là một ngôn ngữ đa mô hình được diễn giải. Một ngôn ngữ rất kỳ lạ. Mặc dù có tên như vậy, nhưng nó không liên quan gì đến Java. Bạn sẽ hiếm khi, nếu không muốn nói là không bao giờ, nhìn thấy ngôn ngữ này bên ngoài trình duyệt web. Về cơ bản, đây là ngôn ngữ dùng để lập trình hành vi trong trình duyệt web và được sử dụng cho những thứ như xác thực biểu mẫu web và ứng dụng web theo phong cách AJAX. Xu hướng trong tương lai có vẻ là xây dựng ngày càng nhiều ứng dụng phức tạp hơn bằng JavaScript, thậm chí là các trò chơi trực tuyến đơn giản và bộ ứng dụng văn phòng. Sự thành công của xu hướng này sẽ phụ thuộc vào những tiến bộ về tốc độ của trình thông dịch JavaScript của trình duyệt. Nếu bạn muốn chính xác, tên thực sự của ngôn ngữ lập trình này là ECMAScript, mặc dù hầu như không ai thực sự gọi nó như vậy.

Điểm mạnh: Đây là cách duy nhất đáng tin cậy để lập trình web phía máy khách

Điểm yếu: Nó chỉ thực sự hữu ích trong trình duyệt web

14.2.9. Perl

Perl là ngôn ngữ đa mô hình được Larry Wall biên soạn vào năm 1986. Ngôn ngữ này có cú pháp hơi lộn xộn và đáng sợ, chỉ những lập trình viên PERL khác mới hiểu ;) Tuy nhiên, rất nhiều lập trình viên kỳ cựu lại yêu thích và sử dụng nó hàng ngày như ngôn ngữ chính của họ. 10 năm trước, Perl phổ biến hơn bây giờ. Chuyện gì đã xảy ra? Rất nhiều lập trình viên mới và thậm chí cả những lập trình viên Perl kỳ cựu (như tôi) đã chuyển sang các ngôn ngữ khác như PHP, Python và Ruby. Perl có lẽ vẫn là ngôn ngữ tốt nhất để xử lý văn bản và viết kịch bản quản trị hệ thống. Tuy nhiên, cá nhân tôi không khuyến nghị Perl là ngôn ngữ lập trình chính.

Điểm mạnh: xử lý văn bản và quản trị hệ thống

Điểm yếu: cú pháp lạ và có lẽ có quá nhiều cách để thực hiện cùng một việc.

14.2.10. Ruby

Ruby là ngôn ngữ hướng đối tượng được Yukihiro Matsumoto biên soạn vào khoảng năm 1995. Đây là một trong những ngôn ngữ hướng đối tượng nhất trên thế giới. Mọi thứ đều là đối tượng trong Ruby, ngay cả chữ cái và số cũng có thể có lệnh gọi phương thức. Đây là ngôn ngữ tuyệt vời để học nếu bạn yêu thích đối tượng. Điểm trừ duy nhất là việc yêu thích hướng đối tượng khiến nó hơi chậm, ngay cả đối với ngôn ngữ được thông dịch.

Điểm mạnh: Có lẽ là ngôn ngữ hướng đối tượng nhất thế giới

Điểm yếu: mô hình đối tượng vượt trội của nó có cái giá của nó...

Đó là 10 ngôn ngữ lập trình được sử dụng nhiều nhất hiện nay và một số bình luận cá nhân về chúng. Hãy nhớ rằng ý kiến ​​giống như mũi, ai cũng có một cái và tất cả đều ngửi thấy) Nếu bạn không đồng ý, vui lòng gửi email cho tôi hoặc viết ý kiến ​​của riêng bạn trên diễn đàn.