



XÂY DỰNG CHƯƠNG TRÌNH DỊCH

TS.Nguyễn Thị Thu Hương –Viện CNTT &TT – ĐHBKHN

Tel (04) 38696121 - Mobi : 0903253796

Email:huongnt@soict.hust.edu.vn,huong.nguyenthithu.hust.edu.vn



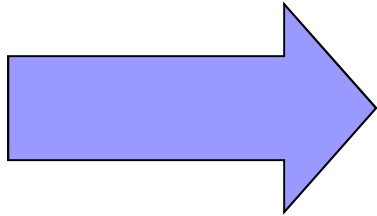
Môn học sẽ nghiên cứu

- Cách thức làm việc của bộ xử lý ngôn ngữ và chương trình dịch
- Sinh mã máy cho những cấu trúc ngôn ngữ cụ thể
- Thiết kế ngôn ngữ: Cú pháp và ngữ nghĩa



Tại sao cần nghiên cứu CT dịch?

- Rèn kỹ năng phát triển ứng dụng quy mô lớn
- Làm việc với các cấu trúc dữ liệu phức tạp
- Tìm hiểu sự tương tác giữa các giải thuật



Bước chuẩn bị cho những dự án lớn trong tương lai.



Những vấn đề chính

- Bộ xử lý ngôn ngữ
- Cấu trúc của một trình biên dịch (1 pha)
- Biểu diễn cú pháp: văn phạm hình thức, BNF và sơ đồ cú pháp
- Phân tích từ vựng
- Phân tích cú pháp: quay lui và tiên định
- Văn phạm LL(k) và phân tích kiểu đệ quy trên xuống
- Ngôn ngữ lập trình KPL: cú pháp và ngữ nghĩa
- Phân tích ngữ nghĩa
- Sinh mã: sinh mã trung gian và sinh mã đích
- Tối ưu mã



Tài liệu tham khảo

- Aho.A.V, Sethi.R., Ullman.J.D.
Compiler : Principles, Techniques and Tools.
Addison Wesley.1986
- Bal.H. E.
Modern Compiler Design.
John Wiley & Sons Inc (2000)
- William Allan Wulf.
The Design of an Optimizing Compiler
Elsevier Science Ltd (1980)
- Charles N. Fischer.
Crafting a Compiler
Benjamin-Cummings Pub Co (1987)




Tài liệu tham khảo (tiếp)

- Niklaus Wirth
Compiler Construction.
Addison Westley. 1996
- Andrew.W.Appel
Modern Compiler Implementation in Java
Princeton University.1998
- Nguyễn Văn Ba
Giáo trình kỹ thuật biên dịch
Đại học Bách Khoa Hà Nội.1994
- Vũ Lục
Phân tích cú pháp
Đại học Bách Khoa Hà Nội.1990
- Bài giảng về ngôn ngữ và phương pháp dịch
- www.sourceforge.net



Đánh giá kết quả học tập

- Giữa kỳ (30%): Tự luận – Lập trình phân tích từ vựng và phân tích cú pháp
- Cuối kỳ (70%): Thi trắc nghiệm



Bài 1. Bộ xử lý ngôn ngữ và trình biên dịch



Ngôn ngữ lập trình cấp cao

- Các ngôn ngữ lập trình được chia thành 5 thế hệ.
- Việc phân chia cấp cao hay thấp phụ thuộc mức độ trừu tượng của ngôn ngữ
 - Cấp thấp : gần với máy
 - Cấp cao : gần với ngôn ngữ tự nhiên



Ngôn ngữ lập trình thể hệ thứ nhất và thứ hai

- Thể hệ thứ nhất : ngôn ngữ máy
- Thể hệ thứ hai : Assembly
- Các ngôn ngữ thuộc thể hệ thứ nhất và thứ hai là ngôn ngữ lập trình cấp thấp



Ngôn ngữ lập trình thế hệ thứ ba

- Dễ hiểu hơn
- Cho phép thực hiện các khai báo, chẳng hạn biến
- Phần lớn các ngôn ngữ cho phép lập trình cấu trúc
- Ví dụ: Fortran, Cobol, C, C++, Basic



Ngôn ngữ lập trình thể hệ thứ tư

- Thường được sử dụng trong một lĩnh vực cụ thể (chẳng hạn thương mại)
- Dễ lập trình, xây dựng phần mềm
- Có thể kèm công cụ tạo form, báo cáo
- Ví dụ :SQL, Visual Basic, Oracle (SQL plus, Oracle Form, Oracle Report). . . .



Ngôn ngữ lập trình thể hệ thứ năm

- Giải quyết bài toán dựa trên các ràng buộc đưa ra cho chương trình chứ không phải giải thuật của người lập trình.
- Việc giải quyết bài toán do máy tính thực hiện
- Phần lớn các ngôn ngữ dùng để lập trình logic, giải quyết các bài toán trong lĩnh vực trí tuệ nhân tạo



Đặc trưng của ngôn ngữ lập trình cấp cao

- Độc lập với máy tính
- Gần với ngôn ngữ tự nhiên
- Chương trình dễ đọc, viết và bảo trì
- Muốn thực hiện chương trình phải dịch sang ngôn ngữ máy
- Chương trình thực hiện chậm hơn



Cú pháp và ngữ nghĩa của ngôn ngữ lập trình

- Cú pháp : Chính tả và văn phạm của các cấu trúc ngôn ngữ
- Ngữ nghĩa : Ý nghĩa và hiệu quả của các cấu trúc ngôn ngữ



Bộ xử lý ngôn ngữ (Language Processor)

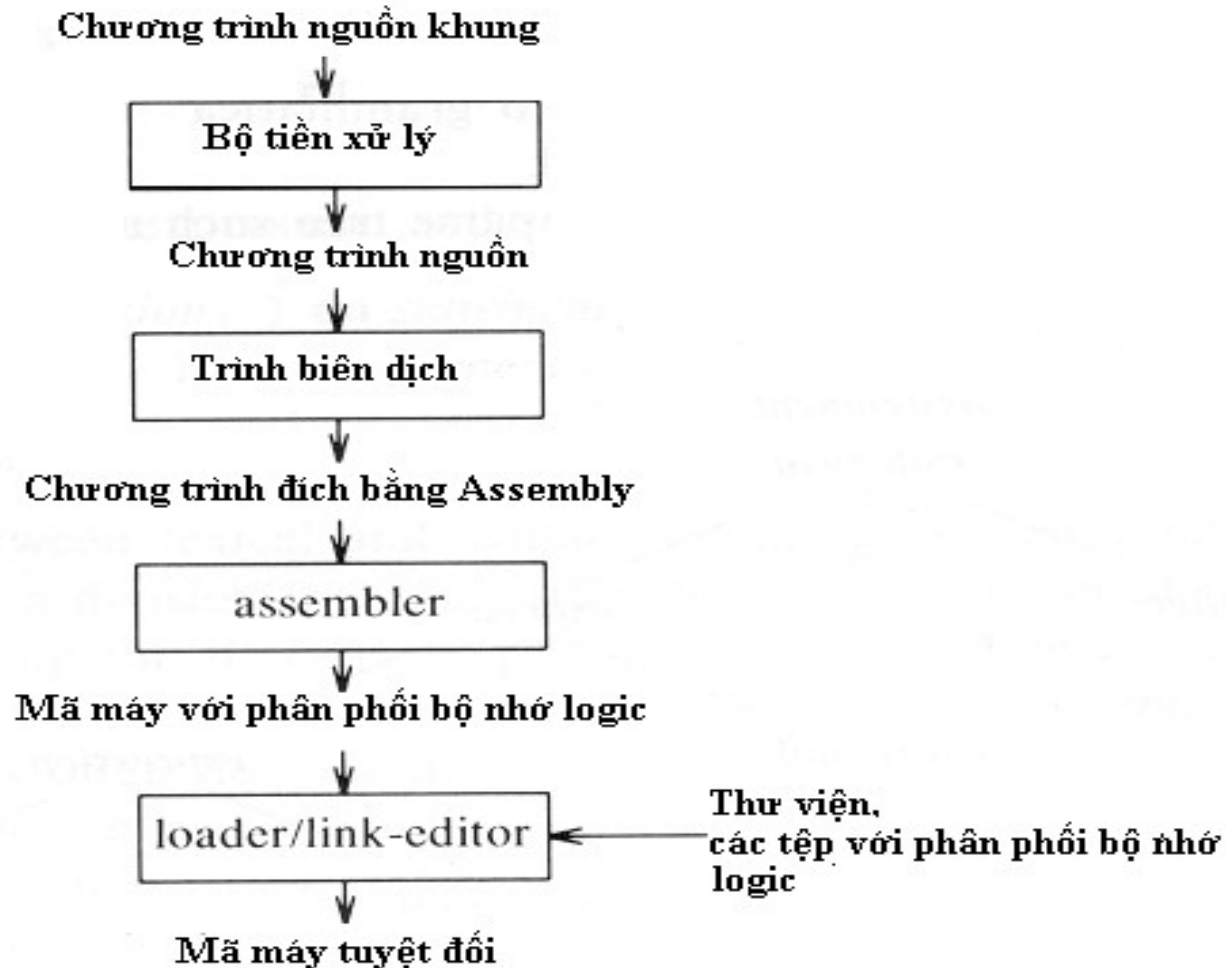
- Phần mềm đọc một chương trình viết bằng một ngôn ngữ (ngôn ngữ nguồn) dịch sang một chương trình tương đương trên ngôn ngữ khác (ngôn ngữ đích)
- Nếu chương trình đích viết trên mã máy, chương trình có thể được người dùng ra lệnh thực hiện



Ví dụ về bộ xử lý ngôn ngữ

- **Compiler: dịch từ ngôn ngữ nguồn sang mã máy**
- **Assembler: dịch từ Assembly sang mã máy**
- **Interpreter: dịch và thực thi trực tiếp từng thao tác của chương trình nguồn dựa trên dữ liệu do người sử dụng cung cấp**
- **Compiler – Compiler: bộ sinh compiler**

Bộ xử lý ngôn ngữ cho những ngôn ngữ cho phép viết chương trình trên nhiều tệp

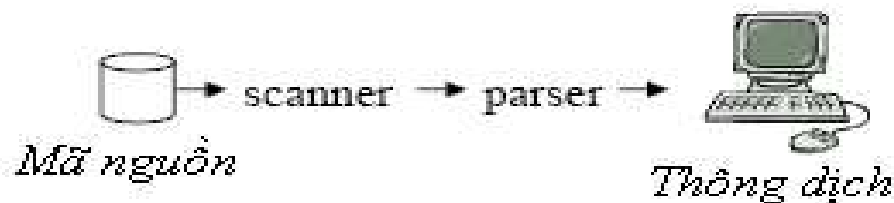


Compiler & Interpreter

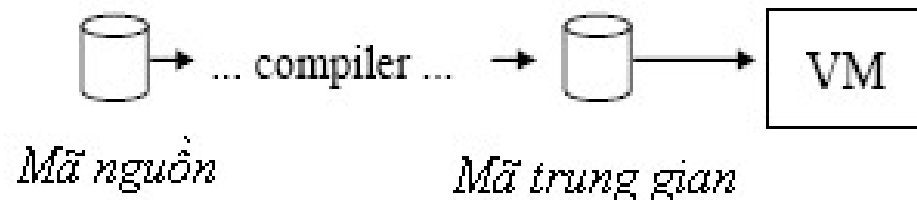
- Compiler : Dịch trực tiếp ra mã máy



- Interpreter : Trực tiếp thực hiện từng lệnh mã nguồn



- Biến thể của Interpreter : thông dịch mã trung gian





Interpreter: Trình thông dịch

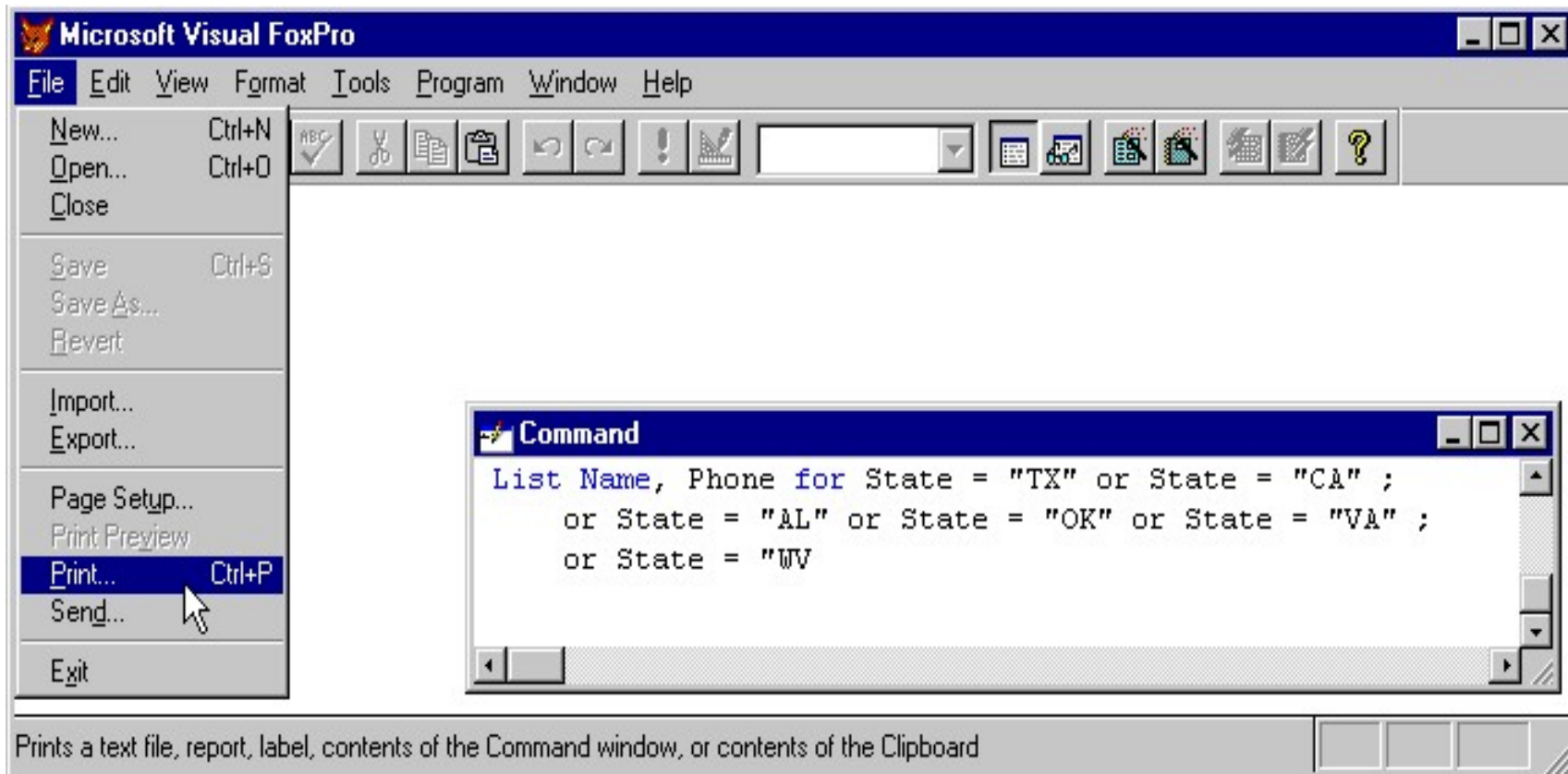
- Một số ngôn ngữ sử dụng trình thông dịch cho phép dịch và chạy trực tiếp từng lệnh
- Mỗi lệnh được dịch thành một đoạn chương trình trong một ngôn ngữ trung gian. Ngôn ngữ trung gian dùng trình dịch compiler.
- Ngôn ngữ hoàn toàn dùng trình thông dịch: Foxpro
- Ngôn ngữ kết hợp thông dịch và biên dịch: Visual Basic, Python



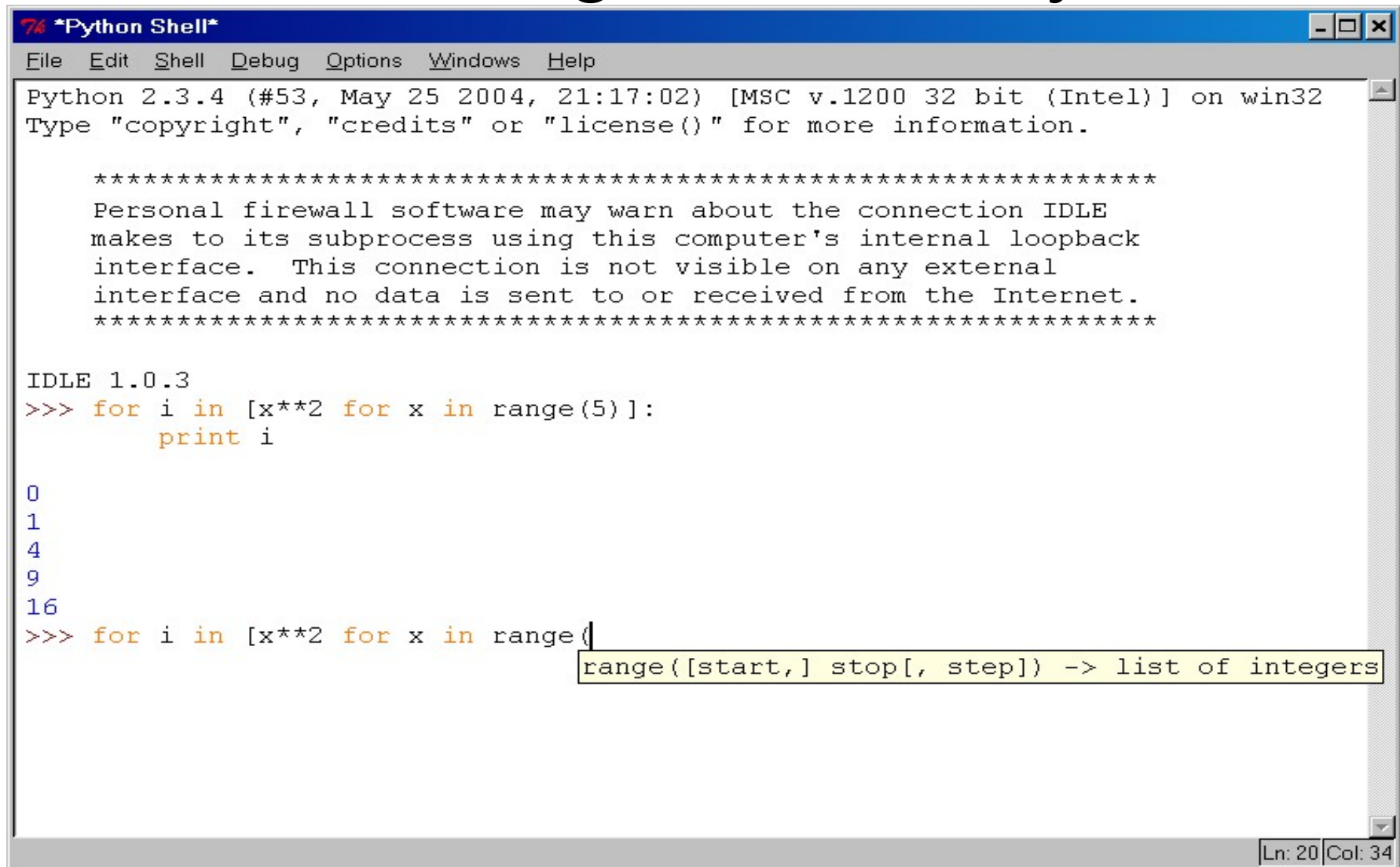
Ngôn ngữ dùng interpreter: Foxpro

- Hai chế độ làm việc
 - Cửa sổ lệnh: Thực hiện từng lệnh
 - Chương trình: Chạy từng lệnh. Các lệnh trước lệnh đầu tiên bị lỗi trong chương trình vẫn được thực hiện

Cửa sổ lệnh của Foxpro



Thực hiện từng lệnh trên Python



The screenshot shows a Python Shell window titled "Python Shell". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area displays the following content:

```
Python 2.3.4 (#53, May 25 2004, 21:17:02) [MSC v.1200 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 1.0.3
>>> for i in [x**2 for x in range(5)]:
    print i

0
1
4
9
16
>>> for i in [x**2 for x in range(
```

A tooltip is visible over the "range(" part of the last line, displaying the text: "range([start,] stop[, step]) -> list of integers".

The status bar at the bottom right shows "Ln: 20 Col: 34".



Compiler (trình biên dịch)

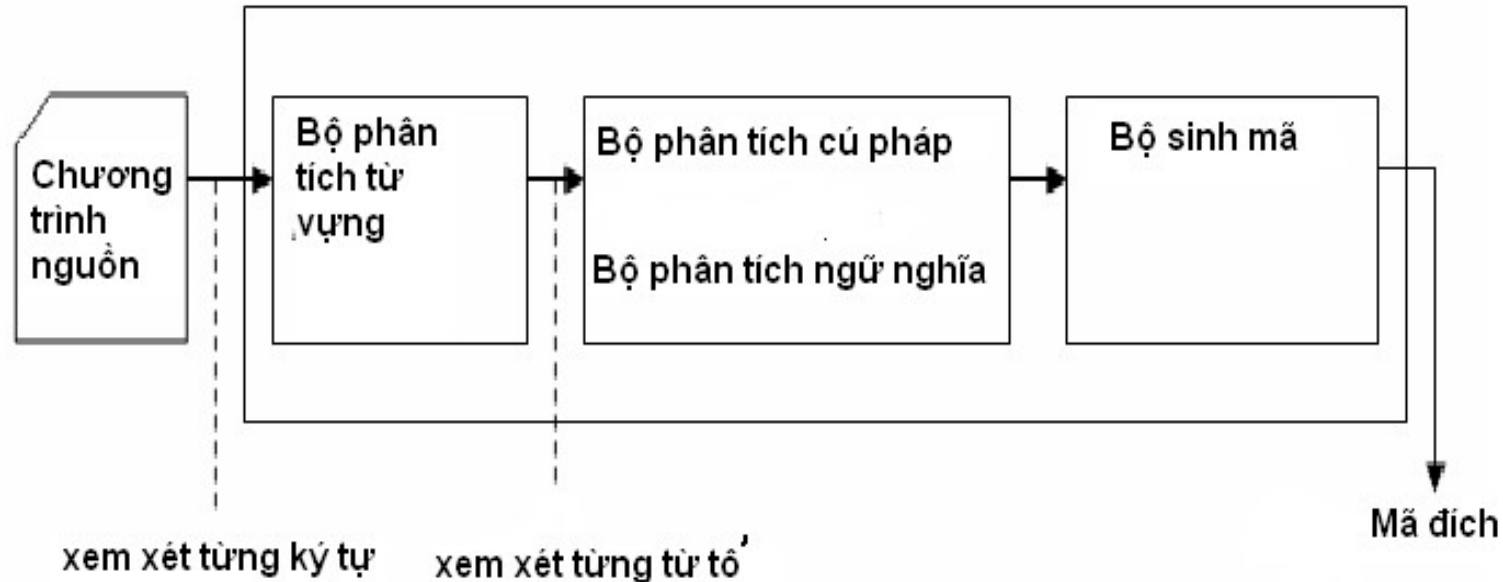
- Mục đích : Dịch chương trình từ ngôn ngữ cấp cao (ngôn ngữ nguồn) sang ngôn ngữ cấp thấp (ngôn ngữ đích).
- Bản thân compiler được viết trên một ngôn ngữ gọi là ngôn ngữ thực hiện



Các công cụ liên quan đến trình biên dịch

- Trình thông dịch (Interpreter)
- Assembler
- Linker
- Loader
- Bộ tiền xử lý (Preprocessor)
- Editor
- Debugger
- Profiler

Các thành phần chính của trình biên dịch





Các giai đoạn của trình biên dịch

- Phân tích từ vựng (Lexical Analysis - Scanner)

Lần lượt xem xét từng ký tự của chương trình nguồn, phân nhóm chúng thành những đơn vị cú pháp gọi là từ tố (token)

- Phân tích cú pháp (Syntax Analysis)

Dãy token do bộ phân tích từ vựng đưa ra được kiểm tra xem có đúng cú pháp không?



Các giai đoạn của trình biên dịch

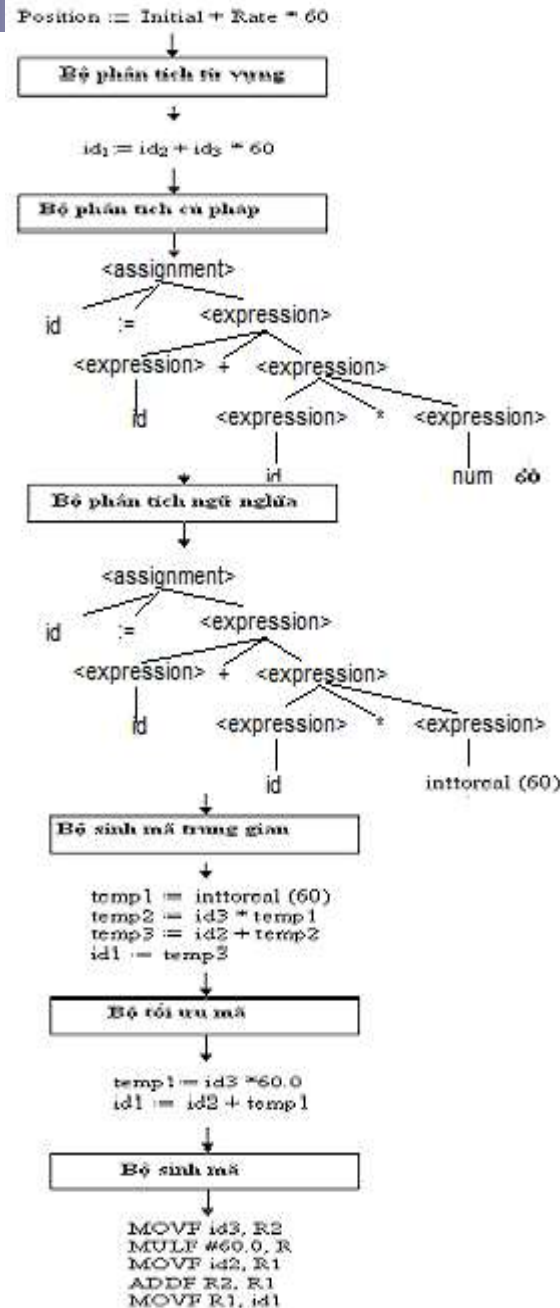
- Phân tích ngữ nghĩa (Semantic Analysis) phân tích ý nghĩa từng lệnh của ngôn ngữ nguồn.
- Sinh mã trung gian (Intermediate Code Generation) thường là mã 3 địa chỉ. Mã trung gian không phụ thuộc máy nên dễ tối ưu.



Các giai đoạn của trình biên dịch

- Sinh mã đích: Sinh ra các lệnh máy để thực hiện thao tác.
- Tối ưu mã: Thực hiện với mã trung gian và cả mã đích nhằm làm cho chương trình hiệu quả hơn.

Quá trình dịch một câu lệnh





Giai đoạn 1: Phân tích từ vựng

- Bộ từ vựng: Chương trình làm nhiệm vụ phân tích từ vựng

- Các công việc của bộ từ vựng

Nhóm các ký tự thành từ tố

Từ tố : đơn vị cú pháp được xử lý trong quá trình dịch như một thực thể không thể chia nhỏ hơn nữa

Nhóm các từ tố theo loại

Loại bỏ các khoảng trống, chú thích



Một số loại từ tổ

Loại từ tổ (token)

- Định danh
- Từ khóa
- Số
- Toán tử
- Dấu phân cách

Thể hiện (Lexeme)

- a, chuongtrinh, x1
- if, else, for
- -1, -2.3E10
- +, -, *, /, =, ==, >>
- :, ;, (,)



Pha 2: Phân tích cú pháp

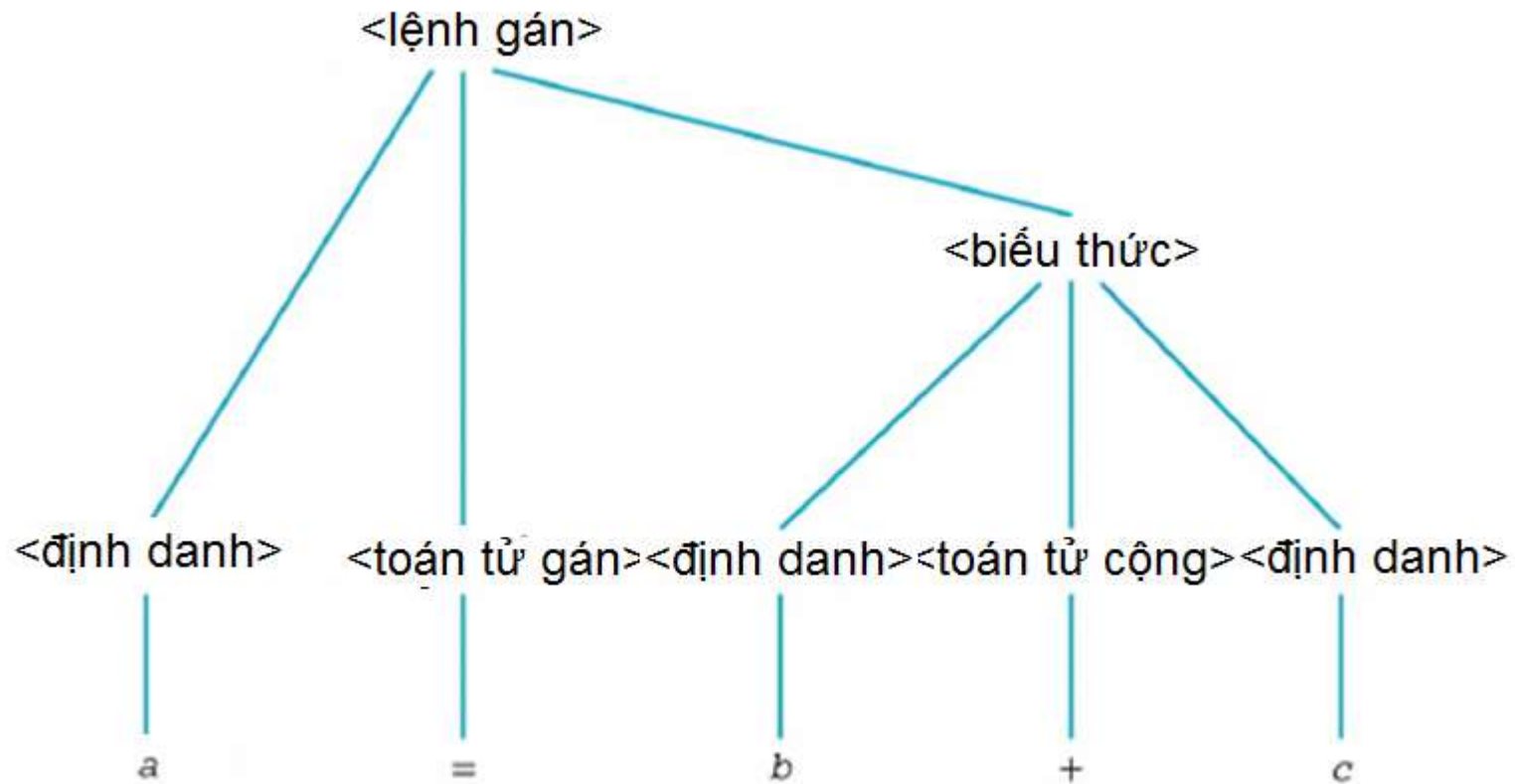
- Trình biên dịch kiểm tra xem những từ tổ mà bộ từ vựng nhận biết được có kết hợp thành những câu lệnh đúng cú pháp không
- Do bộ phân tích cú pháp đảm nhận
- Các thành phần khác của CT dịch xây dựng dựa theo cấu trúc cú pháp



Pha 2: Phân tích cú pháp

- Đầu ra của bộ phân tích cú pháp:
 - Cây phân tích cú pháp (nếu có)
 - Thông báo lỗi nếu ngược lại
- Việc xây dựng được cây phân tích cú pháp chứng tỏ chương trình đúng về cú pháp

Ví dụ: câu lệnh $a = b + c$





Biểu diễn cú pháp

- Cú pháp
 - Cấu trúc văn phạm của một ngôn ngữ
- Bộ phân tích cú pháp cần đưa ra phân tích cho mỗi câu của ngôn ngữ (chương trình)
- Văn phạm hình thức: Mô hình để mô tả cú pháp ở dạng máy đọc được
- BNF: Dạng chuẩn để mô tả văn phạm của ngôn ngữ
- Sơ đồ cú pháp: cách mô tả văn phạm trực quan dưới dạng đồ thị định hướng



Đặc trưng của văn phạm hình thức và BNF

- Các luật của BNF cũng như văn phạm hình thức sử dụng 2 loại ký hiệu ở vế phải
- Ký hiệu kết thúc :
 - Từ tổ của ngôn ngữ
 - Không xuất hiện ở vế trái
- Ký hiệu không kết thúc
 - Ký hiệu trung gian của văn phạm để mô tả cấu trúc ngôn ngữ
 - Cần xuất hiện ở vế trái của ít nhất một luật
- Ký hiệu đầu
 - Ký hiệu không kết thúc ở mức cao nhất
 - Xuất hiện ở gốc cây cú pháp



Khái niệm và kỹ thuật phân tích cú pháp

- Bằng cách áp dụng liên tục các luật mô tả văn phạm
- Nếu bộ PTCP chuyển thành công từ xâu vào thành ký hiệu đầu thì xâu vào đúng cú pháp
- Ngược lại, câu được xem xét không đúng cú pháp



Khái niệm và kỹ thuật phân tích cú pháp

- Vấn đề quan trọng nhất khi xây dựng trình biên dịch là xây dựng một văn phạm
- Bao gồm đầy đủ các cấu trúc của một chương trình
- Không thể tạo nên một luật nào khác



Khái niệm và kỹ thuật phân tích cú pháp

- Văn phạm phải không nhập nhằng
- Nếu văn phạm nhập nhằng, xây dựng được nhiều hơn 1 cây cho mỗi câu được đưa ra phân tích



Pha 3: Phân tích ngữ nghĩa

- Duyệt cây cú pháp của chương trình để xem mọi cấu trúc ngữ nghĩa có đúng không
- Chương trình đúng cả về cú pháp và ngữ nghĩa mới sinh mã được



Pha 4: Sinh mã trung gian

- Chương trình với mã nguồn được chuyển sang chương trình tương đương trong ngôn ngữ trung gian bằng bộ sinh mã trung gian.
- *Mã trung gian* là mã máy độc lập tương tự với tập lệnh trong máy.



Ưu điểm của mã trung gian

- Thuận lợi khi cần thay đổi cách biểu diễn chương trình đích.
- Có thể tối ưu hóa mã độc lập với máy đích cho dạng biểu diễn trung gian.
- Giảm thời gian thực thi chương trình đích vì mã trung gian có thể được tối ưu



Ngôn ngữ trung gian

- Được người thiết kế trình biên dịch quyết định, có thể là:
 - Cây cú pháp
 - Ký pháp Ba Lan sau (hậu tố)
 - Mã 3 địa chỉ ...



Pha 5: Sinh mã đích

- Vào: biểu diễn trung gian của chương trình nguồn
- Ra: chương trình đích
 - Mã Assembly
 - Mã mô phỏng trên máy đích ảo



Các vấn đề thiết kế bộ sinh mã đích

- Input
- Output
- Máy đích
- Lựa chọn câu lệnh
- Cấp phát thanh ghi