

Tradutores

Compiladores e Interpretadores

Adriano B Miranda 7233587
Daniel A dos Santos 7963060
Matheus F N Machado 7965591
Mayara M Yoshino 7965138

1

Tradutores

O que é um tradutor

- Aceitar um conjunto de instruções escritas em uma linguagem de programação de alto nível, que é independente da máquina, e fazer com que as atividades especificadas por estas instruções sejam executadas pelo computador;
- Interpretadores e Compiladores.

2

Compiladores

O que é um compilador

- Um **compilador** tem a finalidade de converter uma linguagem – Linguagem Fonte – de fácil escrita e leitura para os programadores, numa linguagem – Linguagem alvo ou objeto – que possa ser executada pelas máquinas.
- O código executável gerado pelo compilador é dependente do sistema operacional e da linguagem de máquina para o qual o código fonte foi traduzido.
- A enorme variedade de compiladores existentes é bem vinda, visto que existem milhares de linguagens fonte, e as linguagens alvo são também muito variadas.

3

Compiladores

O que é um compilador

- Começaram a aparecer no início da década de 50.
- Muito do trabalho inicial dos compiladores resumia-se a tradução de fórmulas aritméticas para código máquina.

4

Compiladores

O que é um compilador

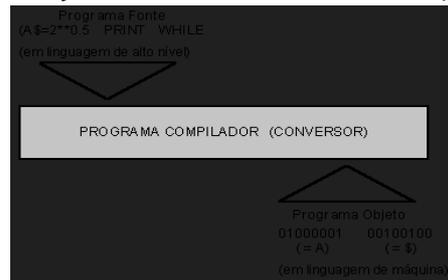
- O primeiro compilador de FORTRAN, por exemplo, demorou 18 trabalhosos meses para implementar.
- Boas linguagens de implementação, ambientes de programação, e ferramentas de software têm também vindo a ser desenvolvidas.

5

Compiladores

O que é um compilador

Ilustração do funcionamento de um compilador:



6

Compiladores

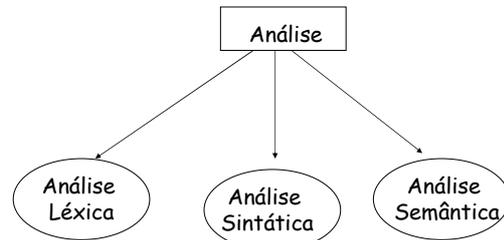
Modelo Análise- síntese da compilação

- Podemos dividir o processo de compilação em duas fases:
- **Análise** : parte o programa fonte em peças constituintes e cria uma representação intermédia do programa fonte.
- **Síntese** : Constrói o desejado programa alvo (código de máquina) a partir da representação intermédia.
- A parte da síntese é a que requer técnicas mais especializadas.

7

Compiladores

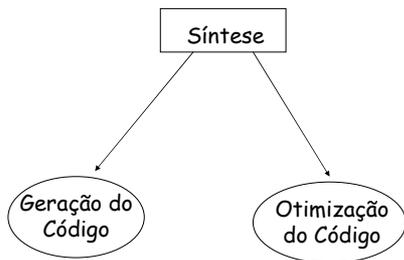
Modelo Análise- síntese da compilação



8

Compiladores

Modelo Análise- síntese da compilação



9

Compiladores

Análise do programa fonte

- Na compilação a análise consiste em 3 partes:
 - Análise Léxica ou Linear:
 - Em que a cadeia de caracteres que forma a estrutura do programa fonte é lido da esquerda para a direita e agrupado em *tokens* que são sequências de caracteres tendo o sentido coletivo.
 - A sua função básica é o reconhecimento e a classificação das estruturas elementares ou classes sintáticas das linguagens.

10

Compiladores

Análise do programa fonte

- Análise sintática ou hierárquica:
 - Na qual caracteres ou tokens são agrupados hierarquicamente em coleções aninhadas com sentido coletivo.
 - Verifica se a estrutura geral do texto ou programa fonte está correta.

11

Compiladores

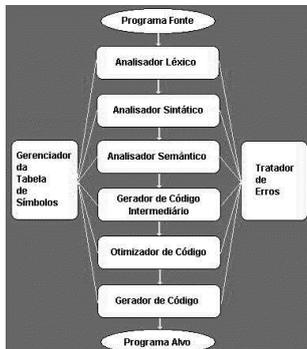
Análise do programa fonte

- **Análise semântica:**
 - Na qual são executadas certas paragens para assegurar que os componentes de um programa são juntamente ajustados em sentido absoluto.
 - Verifica se o programa fonte tem erros semânticos e reúne a informação dos tipos para a fase de gerador de código subsequente.
 - Uma componente importante da análise semântica é a verificação do tipo.

12

Compiladores

Fases de um compilador



13

Compiladores

Fases de um compilador

- Detecção de erros e aviso do erro:
 - Cada fase pode encontrar erros. Porém, depois de descobrir um erro, a fase tem de ocupar-se de alguma maneira com aquele erro, para que a compilação possa prosseguir.
- As fases de análise sintática e semântica normalmente tratam de uma grande fração dos erros detectáveis pelo compilador.

14

Compiladores

Fases de um compilador

- Otimização do código:
 - Esta fase tenta melhorar o código intermédio, de forma a que resulte num código de máquina mais rápido a executar.
- Geração do código:
 - A fase final do compilador é a geração de código alvo, consistindo normalmente no restabelecimento no código máquina.
 - Neste ponto, após o programa fonte ter sido analisado e aprovado, segundo a sua sintaxe, e livre de erros semânticos, o compilador tem condições de escrever um programa equivalente na linguagem alvo.

15

Interpretadores

Como funcionam os interpretadores

- O funcionamento dos interpretadores é muito parecido ao dos compiladores.
- O interpretador traduz o código linha a linha.
- O código fonte não é totalmente traduzido antes de ser executado.
- Não existem fases distintas nem se produz código intermediário.
- Passa o tempo todo a ler e a traduzir código.

16

Interpretadores

Os Exemplos de interpretadores

- **Internet;**
- **Excel, Word Basic, Access, ...**
;
- **SmallTalk;**
- **AutoLisp;**
- **Lisp.**

17

Exemplos de Linguagens

Compiladas e Interpretadas

- **Java;**
- **BASIC;**
- **Bash;**
- **C;**
- **Perl.**

18

Comparação

	Vantagens	Desvantagens
Compiladores	Execução mais rápida	Várias etapas de tradução
	Permite estruturas de programação mais completas	Programação final é maior, necessitando mais memória para a sua execução
	Permite a optimização do código fonte	Processo de correcção de erros e depuração é mais demorado
Interpretadores	Depuração do programa é mais simples	Execução do programa é mais lenta
	Consome menos memória	Estruturas de dados demasiado simples
	Resultado imediato do programa ou rotina desenvolvida	Necessário fornecer o programa fonte ao utilizador