Interpretadores

Existem outras soluções para a implementação de linguagem de alto nível. Ex: interpretadores.

Interpretadores: estes programas em vez de traduzir de uma vez o programa fonte para então executá-lo, decodificam unidades básicas do programa (por exemplo, comandos) p/ executá-los imediatamente [KOWALTOWSKI, 83].

Certos tradutores transformam uma linguagem de programação em uma linguagem simplificada, chamada código intermediário, que pode ser execeutado por um interpretador. Ex: Java, Basic.

Interpretadores

Por que são usados os interpretadores?

- para executar linguagens de comandos, dado que cada operador numa tal linguagem é usualmente uma invocação de uma rotina complexa, como um editor ou compilador.

Similarmente, algumas linguagens de "nível muito alto", como APL, são normalmente interpretadas, pois existem muitos atributos de dados que não podem ser determinados em tempo de compilação [AHO, 86].

Interpretadores

Desvantagens dos interpretadores:

- em geral, acarreta a decodificação repetida de várias partes do programa-fonte e pode ser muito ineficiente.

Porém, na prática, uma parte da decodificação é feita antes de iniciar-se a interpetação, tornando o processo mais eficiente [KOWALTOWSI, 83].

Vantagem da interpretação:

- às vezes de implementação, dependendo das características da linguagem.

Interpretadores

- Não existe uma linha divisória muito clara entre a interpretação e a compilação.
- Mesmo numa linguagem compilada, certas partes do programa podem ser interpretadas durante a execução. Ex: as operações de E/S e operações que não tem correspondência direta em linguagem de máquina.

Passos de Compilação

Denomina-se **passos de compilação** a atividade do compilador que exija, para seu processamento, uma lieitura completa do texto-fonte ou de alguma de suas formas intermediárias equivalentes [JOSÉ NETO, 87].

Compilador de um passo e de vários passos

Definições [JOSÉ NETO, 87]:

Compilador de um único passo: é aquele que executa todas as suas atividades com uma única leitura do texto-fonte, gerando como saída o código-objeto final a ser executado por meio de um único programa.

Compilador de vários passos:efetua a leitura do texto-fonte e sua conversão para uma forma intermediária equivalente através de um programa que corresponde ao primeiro passo do compilador. A saída deste programa alimenta um outro programa, que é o segundo passo do compilador, que por sua vez gera outro código intermediário que irá abastecer o próximo passo de compilação e assim por diante, até que o código-objeto final seja finalmente produzido pelo último passo de compilação.

Compilador de vários passos

<u>Vantagens</u> do compilador de vários passos:

- menor utilização de memória do computador já que cada passo exerce apenas uma parte das funções de todo o computador;
- maior possibilidade de se efetuar otimizações levando a um programa objeto mais eficiente quanto ao espaço ocupado ou ao termo de processamento;
- os projetos e implementação das várias partes do compilador são mais independentes.

Compilador de vários passos

<u>Desvantagens</u> do compilador de vários passos:

- maior volume de E/S, quando os programas intermediários e os passos do compilador não ficam residentes na memória;
- normalmente, aumento do tempo de compilação;
- aumento do projeto total, com a necessidade de introdução das linguagens intermediárias.