# Capítulo 6

# Estrutura geral de compiladores

Nos primeiros sistemas computacionais, a programação era feita por meio de painéis com fios e cabos. O programador precisava possuir um conhecimento avançado sobre a arquitetura do computador – cada uma de suas peças – para poder criar uma aplicação no sistema. Atualmente, com as linguagens de programação, o processo de desenvolvimento de softwares se tornou mais fácil, pois não exige que o profissional saiba, em profundidade, como funciona o hardware, porque isso fica a critério do próprio sistema operacional.

O compilador é um software que tem a função de traduzir o código-fonte desenvolvido pelo programador em um software que possa ser executado diretamente pelo usuário, ou seja, você escreve todo o código-fonte e depois pede para o compilador convertê-lo em um programa. Uma vez tendo o programa em mãos, você pode distribuí-lo aos amigos, instalá-lo em uma empresa etc. O compilador é um tipo de tradutor, como veremos adiante.

Quando um programador desenvolve um software por meio de um código-fonte, ele necessita converter esse código para uma linguagem de máquina, ou seja, que a máquina entenda. Para isso, usa-se um tradutor, que é um utilitário com a função de facilitar a vida do programador, convertendo o código desenvolvido em uma linguagem de alto nível (entendida mais facilmente pelo programador) em uma linguagem de máquina (entendida pelo computador). Dependendo da linguagem utilizada pelo programador, o tradutor pode ser de um destes dois tipos:

- **Montador:** traduz o código-fonte que foi desenvolvido em linguagem de montagem, como por exemplo, a Assembly;
- **Compilador:** traduz o código-fonte que foi desenvolvido em linguagem de alto nível, como C, C++, Pascal, Java etc.

Os compiladores são bastante utilizados. Nos cursos técnicos de informática e nos de graduação da área de computação existem diversas disciplinas de programação de computadores em que eles se constituem na principal ferramenta utilizada, além da própria linguagem de programação. Confira dois exemplos de softwares compiladores:

- Javac: compilador da linguagem Java;
- g++: compilador OpenSource da linguagem c++.





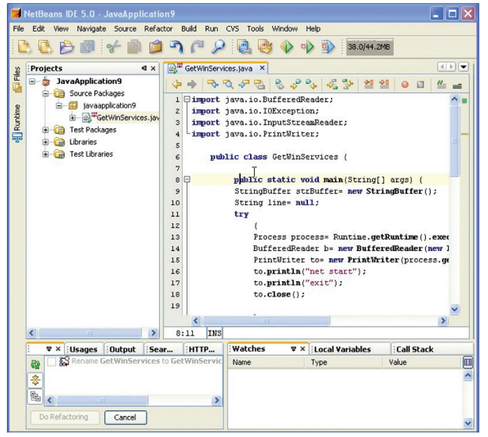



**Figura 170**
IDE NetBeans.

Na maioria das vezes, para melhorar sua rotina o programador adota uma IDE (Integrated Development Environment), ou seja, um ambiente de desenvolvimento integrado, onde há – além de um editor de textos, muitas vezes capaz de corrigir e sugerir o código durante sua digitação – um depurador, um compilador, um linker e uma interface facilitada para executar o software, depois de compilado. Exemplos de IDEs são o NetBeans, o Eclipse (Java) e o DevC++ (C++), ilustrados nas figuras 170, 171 e 172.

**Interpretador** é um tipo de tradutor que não gera programas, simplesmente executa as instruções no momento em que o usuário as solicita. Exemplos de linguagens de programação interpretadas: PHP, Perl e Basic.

A maior desvantagem, nesse caso, é que o código-fonte fica visível para o usuário, de maneira que qualquer pessoa poderá copiar o que você criou. Outra ressalva é

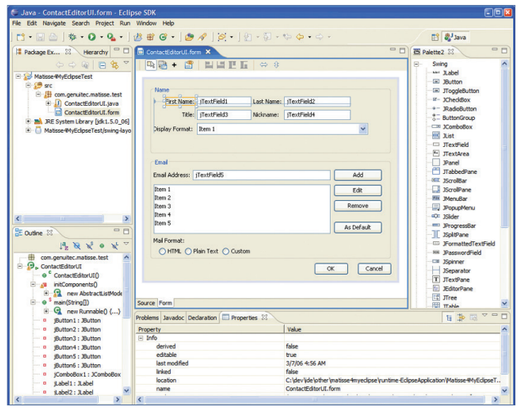
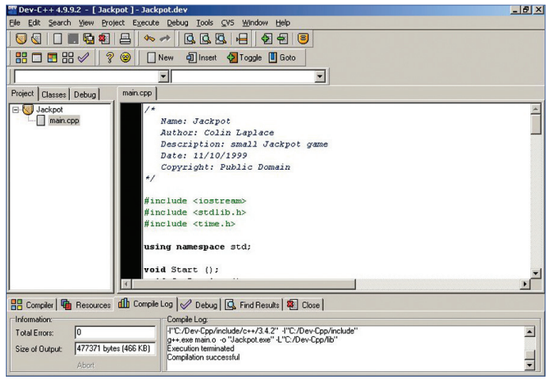



**Figura 171**
Eclipse.

**Figura 172**
Dev C++.




o desempenho, já que toda vez que o programa for executado o processador terá de processar as instruções linha a linha, algo que demora (figura 173).

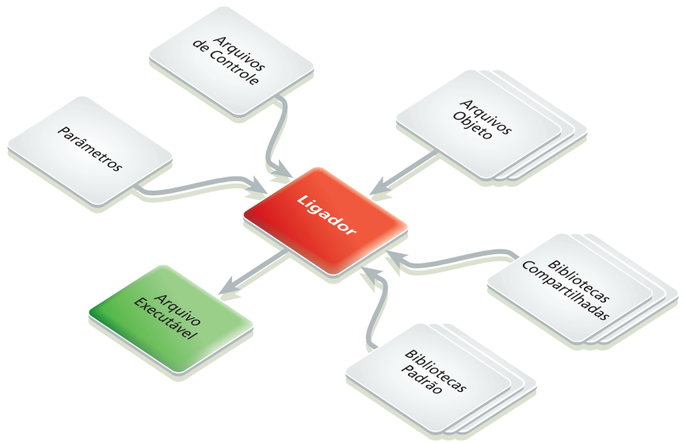
**Linker (ou ligador)** é o utilitário (figura 174) responsável por pegar um programa que já foi traduzido e introduzir as bibliotecas necessárias para seu correto funcionamento. Muitas vezes, quando desenvolvemos programas, precisamos adicionar bibliotecas externas com recursos úteis para o programa que estamos desenvolvendo.

```
<?php
echo "Olá Mundo!";
?>
```

O linker tem a responsabilidade de concretizar essa adição para a geração do programa final: software binário. Alguns compiladores, mais suas funções básicas, agregam também as do linker, que, neste caso, se torna desnecessário, já que o próprio compilador resolve também essas referências simbólicas a bibliotecas externas. Se você já programa em alguma linguagem ou já se atreveu a escrever alguma linha de código em algum software de desenvolvimento, deve ter ouvido falar do **depurador**, também conhecido como **debugger**. Trata-se de uma ferramenta muito útil para o programador. Imagine que você esteja desenvolvendo um programa e o resultado apresentado por ele não é o que você esperava. Da mesma forma, pode ocorrer que esse programa tenha muitas linhas de código e você não tem a mínima ideia de onde está acontecendo o problema. A utilidade do depurador está em sua função de permitir ao usuário acompanhar a execução do programa, visualizando os resultados em tempo real, em busca de possíveis erros de lógica. Com ele você pode seguir cada linha de código, verificando a saída do programa para localizar onde o problema está e podendo, assim, corrigi-lo (figura 175).

**Figura 173**
Site desenvolvido em PHP.


**Figura 174**
O utilitário Linker.




O depurador também pode ser utilizado em conjunto com o copilador para apresentar as linhas onde existem possíveis erros de codificação constatados durante o processo de compilação.

Em IDEs, o depurador normalmente entra em execução durante o processo de compilação, depois que o usuário executa essa ação, clicando no botão "Compilar". O debugger normalmente é apresentado na parte de baixo da janela e é muito importante para quem pretenda desenvolver um bom software.

**Figura 175**
Resultados do depurador do compilador g++.