第2章 二进制文件

2.1 从源代码到可执行文件

一个 C 语言程序的生命是从源文件开始的，这种高级语言的形式更容易被人理解。然而，要想在操作系统上运行程序，每条 C 语句都必须被翻译为一系列的低级机器语言指令。最后，这些指令按照可执行目标文件的格式打包，并以二进制文件的形式存放起来。

本节我们首先回顾编译原理的基础知识，然后以经典著作 *The C Programming Language* 中的第一个程序 hello world 为例，讲解 Linux 下默认编译器 GCC（版本 5.4.0）的编译过程。

2.1.1 编译原理

编译器的作用是读入以某种语言（源语言）编写的程序，输出等价的用另一种语言（目标语言）编写的程序。编译器的结构可分为前端（Front end）和后端（Back end）两部分。前端是机器无关的，其功能是把源程序分解成组成要素和相应的语法结构，通过这个结构创建源程序的中间表示，同时收集和源程序相关的信息，存放到符号表中；后端则是机器相关的，其功能是根据中间表示和符号表信息构造目标程序。

编译过程可大致分为下面 5 个步骤，如图 2-1 所示。

（1）词法分析（Lexical analysis）：读入源程序的字符流，输出为有意义的词素（Lexeme）；

（2）语法分析（Syntax analysis）：根据各个词法单元的第一个分量来创建树型的中间表示形式，通常是语法树（Syntax tree）；

（3）语义分析（Semantic analysis）：使用语法树和符号表中的信息，检测源程序是否满足语言定义的语义约束，同时收集类型信息，用于代码生成、类型检查和类型转换；

（4）中间代码生成和优化：根据语义分析输出，生成类机器语言的中间表示，如三地址码。然后对生成的中间代码进行分析和优化；