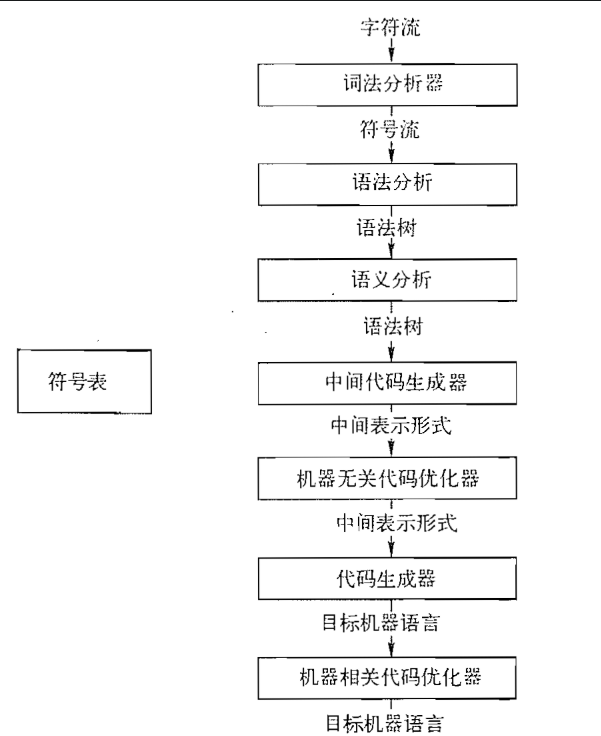
**高级编译原理课后作业**

——什么是编译原理，及编译原理的作用

编译原理是想要从事计算机行业学生的一门必修课。编译原理介绍了编译程序构造的过程和方法。内容包括语言和文法、词法分析、语法分析、语法制导翻译、中间代码生成、存储管理、代码优化和目标代码生成。

虽然只有极少部分的人会从事编译器的开发工作，但通过学习编译原理会了解到当我们使用高级计算机语言编写程序时，编译器在计算机后台都干了什么。在冯诺依曼体系结构完善初时，程序都是用0101的机器码组合而成。尽管汇编语言的出现代替的费时、乏味、难于学习的机器码，但是汇编语言中存在的编写起来也不容易，阅读和理解很难，汇编语言的编写严格依赖于特定的机器等问题促使人们开发更加高级，更加符合人类习惯的程序设计语言。高级的程序设计语言对于人类来说是易于理解的、方便的，但是对于计算机来说则是不可读的，无法直接运行的。这时就需要编译原理所支持的编译器来对高级编程语言做支持，将代码翻译成计算机可以看懂的01机器码，以供计算机执行。

编译原理最大的衍生就是针对于不同高级语言的编译器（或者说有了编译原理及编译器，才慢慢出现越来越多的高级语言），虽然对于不同的高级编程语言，其编译器内部实现稍有不同，但原理是互通的。编译器的基本流程如下图所示：

编译器的第一个步骤称为词法分析或扫描。词法分析器读入组成源程序的字符流，并将其组成有意义的词素的序列。

第二个步骤为语法分析或解析。语法分析器使用由词法分析器生成的各词法单元的第一个分量来创建树形的中间表示。

步骤三为语义分析，通过前几步所生成的语法树、符号表来检查源程序是否和语言定义的语义一致以便在中间代码生成过程使用。

步骤四为中间代码生成其特点有两个：1.易于生成；2.能够轻松地翻译为目标机器上的语言。

步骤五为代码优化，代码优化试图改进中间代码，以便生成更好的目标代码。即省时、省空间或能耗更低。

步骤六为最终代码生成，以中间表示形式作为输入，并把它映射为目标语言。如果目标语言是机器代码，则必须为每个变量选择寄存器或内存位置，中间指令则被翻译为能够完成相同任务的机器指令序列。

以上为编译器执行的整个过程。

例如编译原理中的词法分析可以应用与自然语言处理中，使用词法分析将输入的字符串从左至右逐个字符进行扫描，从而产生一个个的单词符号。其主要任务是：能正确的把一串连续的字符切分成一个一个的词；能正确地判断每个词的词性，以便于后续的句法分析的实现。为自然语言处理的基本。