
编译原理实验指导书

实验三 中间代码生成

实验目的

1. 巩固对中间代码生成的基本功能和原理的认识。
2. 能够基于语法指导翻译的知识进行中间代码生成。
3. 掌握类高级语言中基本语句所对应的语义动作。

实验内容

在词法分析、语法分析和语义分析程序的基础上，将C--源代码翻译为中间代码（三地址代码形式）。

在本次实验中，我们对输入的C--语言源代码文件做如下假设：

- 1) 假设1：不会出现注释、八进制或十六进制整型常数、浮点型常数或者变量。
- 2) 假设2：不会出现类型为结构体或高维数组（高于1维的数组）的变量。
- 3) 假设3：任何函数参数都只能为简单变量，也就是说，结构体和数组都不会作为参数传入函数中。
- 4) 假设4：没有全局变量的使用，并且所有变量均不重名。
- 5) 假设5：函数不会返回结构体或数组类型的值。
- 6) 假设6：函数只会进行一次定义（没有函数声明）。
- 7) 假设7：输入文件中不包含任何词法、语法或语义错误。

程序需要将符合以上假设的C--源代码翻译为中间代码。

实验要求

（1）输入格式

程序的输入是一个包含C--源代码的文本文件，程序需要能够接收一个输入

文件名和一个输出文件名作为参数。例如，假设程序名为cc、输入文件名为tst1、输出文件名为out1.ir，程序和输入文件都位于当前目录下，那么在Linux命令行下运行./cctest1out1.ir即可将输出结果写入当前目录下名为out1.ir的文件中。

（2）输出格式

实验三要求程序将运行结果输出到文件。输出文件要求每行一条中间代码，每条中间代码的含义如前文所述。如果输入文件包含多个函数定义，则需要通过FUNCTION语句将这些函数隔开。FUNCTION语句和LABEL语句的格式类似，具体例子见后面的样例。

对每个特定的输入，并不存在唯一正确的输出。我们将使用虚拟机小程序对中间代码的正确性进行测试。任何能被虚拟机小程序顺利执行并得到正确结果的输出都将被接受。此外，虚拟机小程序还会统计中间代码所执行过的各种操作的次数，以此来估计程序生成的中间代码的效率。

（3）测试环境

程序将在如下环境中被编译并运行（同实验一）：

- 1) GNU Linux Release: Ubuntu 20.04, kernel version 5.1.3.0 - 44-generic
- 2) GCC version 7.5.0
- 3) GNU Flex version 2.6.4
- 4) GNU Bison version 3.5.1

（4）现场验收

要求当堂完成实验内容，并进行现场验收。未当堂完成验收者将扣除相应分数。如需补查，需在每次实验结束一周内向实验助教提出申请，并在实验助教指定的时间完成补查。

（5）提交要求

实验三结束后一周内提交如下内容（超过指定时间未提交将影响实验成绩）：

- 1) Flex、Bison以及C语言的可被正确编译运行的源代码程序。
- 2) 一份word格式的实验报告，内容包括：
 - a) 程序实现了哪些功能？简要说明如何实现这些功能。清晰的说明有助于助教对你的程序所实现的功能进行合理的测试。
 - b) 程序应该如何被编译？可以使用脚本、makefile或逐条输入命令进行编译，

请详细说明应该如何编译你的程序。无法顺利编译将导致助教无法对你的程序所实现的功能进行任何测试，从而丢失相应的分数。

c) 实验报告的长度不得超过三页。所以实验报告中需要重点描述的是程序中的亮点，是你认为最个性化、最具独创性的内容，而相对简单的、任何人都可以做的内容则可不提或简单地提一下，尤其要避免大段地在报告里贴代码。实验报告中所出现的最小字号不得小于五号字（或英文11号字）。

实验评分标准

一、课堂表现（10分）

1. 出勤情况（迟到，早退，缺席）
2. 是否遵守课堂纪律

二、操作表现（50分）

1. 功能齐全，结果正确无误（25分）
2. 程序输出格式规范（5分）
3. 讲解清楚明了，具有良好的演示效果（10分）
4. 当堂按时完成（10分）

三、实验报告（40分）