

编译原理实验指导书

(Introduction to the Experiments of Compiling Principles)

北京工业大学计算机学院软件系编译原理课程组

2015 年 2 月修订

目 录

1.	教学定位与基本安排.....	2
2.	词法分析程序.....	4
3.	语法制导的三地址代码生成程序.....	6
4.	实验验收.....	12
5.	实验报告.....	14

教学定位与基本安排

编译原理的课程实验既是培养学生理论联系实际的必要手段也是检验我们教学理念的贯彻与实现、检查教学效果、让学生展示能力的重要途径。具体目标如下。

1. 经历复杂系统的设计与实现，能够进行多种方法、工具、环境的比较、评价和选择。方法选择：选择实现词法、语法分析的方法；实现途径选择：直接设计实现、使用某种自动生成工具设计实现(自学)；工具与环境选择：使用的开发语言和环境；比较与评价：在组间相互评价中锻炼评价能力。
2. 通过按组完成系统设计与实现提升自身的团队协作能力。能够在分工、设计、实现、口头和书面报告等环节中相互协调、相互配合。
3. 能够在实验系统设计实现过程中的组内讨论，验收报告撰写、陈述发言中有效表达实验系统开发问题。

为了达到上述目的，作为编译原理课程的上机实验，最好是能完成一个小型语言的基本编译系统，但这需要有较好的实验条件特别是较多的实验学时（理想的情形是专门配置一个两到三周的课程设计）的支撑。对于一个只有 56 学时的编译原理课程的上机实验（仅为 12 学时），构造一个小型语言的完整的编译系统显然是不现实的。为最大程度达到教学目的，将本课程实验分解为 3 个部分：

- （1）词法分析器设计与实现；
- （2）语法分析器的设计与实现；
- （3）语法制导的三地址代码生成器的设计与实现。

这虽然是 3 个部分可以分离（分阶段）实现，我们要求大家将其当作将一个简单语言翻译成等价的三地址码的编译系统的设计与实现的 3 个相关模块，在设计和实现中统筹考虑，并且在验收中作为一个系统进行验收。

通过这个系统的设计与实现，首先是引导学生走过构造一个微型编译系统的基本框架的主要工作流程，具体体验如何在理论指导下开展实践活动——将基本的原理用于系统设计与实现的实践，在实践中如何加深对理论的理解，培养学生按照编译理论和编译系统设计与实现的基本方法，实现一个系统（理论指导实践）；其次是培养学生系统能力（系统的设、分析与实现）；第三是培养学生的软件系统实现能力（算法、程序设计与实现）；第四是通过分

小组，培养学生的团队合作精神与能力；第五是培养学生查阅资料，获取适当工具、使用适当工具，了解编译技术进展的能力；第六是培养学生表达（书面语口头）能力。

要求学生在学习理论的过程中就逐步考虑实现的问题，开始进行适当的设计与初步实现，在实验学时执行阶段，按照要求进行设计与实现工作，以此来解决 12 个学时的学时量不够的问题。在 12 个学时中，前 10 个学时留给各小组各自集中开展研究、设计与实现工作，并准备验收报告（要求制作 PPT，准备好系统演示），两个学时用于“验收报告”（口头报告、系统演示），各组可以安排一人主持，保证全体成员参与报告。除了口头的报告外，每个小组还需要撰写一份实验报告，对系统的设计与实现进行描述。

按照要求设计实现这样一个系统，是本专业教学中的一个相对复杂的系统，能够体现对学生进行“复杂工程问题”处理能力的培养。首先，这是一个既有设计又有实现的工程问题；其次这个工程问题的解决必须运用编译系统设计的基本原理，包括使用一些基本的处理方法；第三，必须根据基本原理和系统实现的需要，通过抽象，实现对问题及其处理过程的分析 and 符号（形式化）描述，包括建立语言相关的描述模型；第四，至少包括 3 个主干子系统和一些辅助系统，整个系统具有一定的规模；第五，该系统的构建，不仅涉及到编译的基本原理，还涉及程序设计方法、算法设计、系统集成，以及相互间协调，按照要求，还涉及到问题的书面和语口头表达等。这些特征表明，该系统的设计与实现确实满足华盛顿协议等对“复杂工程问题”的界定，作为课程中的一项实验，能够体现对学生进行“复杂工程问题”处理能力的培养。

考虑到这是一次教学锻炼，可以根据教学班级的情况，采用随机方式安排分组，每组 3-4 人，各组随机安排组长一名，负责组织组内的分工协调等工作。提醒各组长重视并利用好这次机会，好好锻炼一下系统的设计、任务的恰当划分、人员的分工协调等工作，注意各个子系统之间的关联关系以及各系统负责同学之间的沟通。总体设计的角色不限于组长，本组内的任何成员都是总体设计的候选人。

实验一：词法分析程序的设计与实现

(Scanner for a tiny programming language)

实验学时：4 学时

实验类型：综合程序设计

先修课程（含实验环节）：

高级语言程序设计及其课程设计、数据结构及其课程设计

适用专业：计算机类专业

一. 实验目的

基本掌握计算机高级程序设计语言的词法分析程序的开发方法。

二. 实验内容

选择适当的方法（自行设计、使用 Lex 等自动生成工具），设计实现一个能够分析三种整数、标识符、主要运算符和主要关键字的词法分析程序。

三. 实验要求

1. 根据以下的正规式，编制正规文法，画出状态图；

基本要求

标识符	$\langle \text{字母} \rangle (\langle \text{字母} \rangle \langle \text{数字字符} \rangle)^*$
十进制整数	$0 (1 2 3 4 5 6 7 8 9)(0 1 2 3 4 5 6 7 8 9)^*$
八进制整数	$0(1 2 3 4 5 6 7)(0 1 2 3 4 5 6 7)^*$
十六进制整数	$0x(0 1 2 3 4 5 6 7 8 9 a b c d e f)(0 1 2 3 4 5 6 7 8 9 a b c d e f)^*$
运算符和分隔符	$+ - * / > < = () ;$
关键字	$\text{if} \text{then} \text{else} \text{while} \text{do}$

附加要求

标识符 $\langle \text{字母} \rangle (\langle \text{字母} \rangle | \langle \text{数字字符} \rangle)^* (\epsilon | ((_ | .)(\langle \text{字母} \rangle | \langle \text{数字字符} \rangle)^+))$

十进制数 $((1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)^* 0)(\epsilon | ((0|1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)^*))$

八进制数 $0((0|1|2|3|4|5|6|7)^+ (\epsilon |(. (0|1|2|3|4|5|6|7)(0|1|2|3|4|5|6|7)^+))$

十六进制数 $0x(0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f)(0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f)^* (\epsilon |(. (0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f)(0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f)^+))$

2. 根据状态图，设计词法分析函数 `int scan()`，完成以下功能：

基本要求

- 1) 从键盘读入数据，分析出每一个单词；
- 2) 返回单词种别（用整数表示）；
- 3) 返回单词属性。

附加要求

- 1) 作为一个独立的程序，从源程序文件中读取源程序，将其变换成相应的符号序列，建议用文件的形式存放这一序列；
- 2) 作为一个独立的子程序设计，从文件中读取源程序，每被调用一次，返回当前的一个单词建议作为中间结果，同时用文件的形式存放相应的单词序列；
- 3) 可以将 `if`、`then`、`else`、`while`、`do` 等作为保留字处理。鼓励大家将这些定义符当做非保留字处理；
- 4) 注意构建相应的符号表，记录标识符、常量的有关信息（你认为后面会用得到的，而现在又能知道的信息）。

3. 编写测试程序完成调用，输出单词种别和属性。

四. 实验环境

PC;

LINUX 操作系统或 Windows 操作系统;

Turbo C 程序集成环境或 Visual C++、JAVA 等程序集成环境（可以选择自己希望使用的语言）。

五. 实验步骤

1. 根据状态图，设计词法分析算法；

2. 采用 C 语言等适当的高级语言，设计实现词法分析器。也可以用 Lex 等自动生成工具生成该词法分析器；
3. 设计测试数据，编制测试程序；
4. 调试程序。

注意：程序需要对一些典型的问题有相应的处理。

六. 测试数据示例

输入数据：

```
0 92+data> 0x3f 00 while a+acc>xx do x=x-1;
a=6.2+a*0X88.80;
if a>b then a=b else a=b-1+c;
```

输出数据（这些单词的单词种别及其属性）：

INT10	0	INT10	1
INT10	92	;	—
+	—	IDN	a
IDN	data	=	—
>	—	REAL10	6.2
INT16	63	+	—
INT8	0	IDN	a
WHILE	—	*	—
IDN	a	REAL16	136.5
+	—	;	—
IDN	acc	IF	—
>	—	IDN	a
IDN	xx	>	—
DO	—	IDN	b
IDN	x	THEN	—
=	—	IDN	a
IDN	x	=	—
-	—	IDN	b

ELSE	—		INT10	1
IDN		a	+	—
=	—		IDN	c
IDN		b	;	—
-	—			

实验二/三：语法制导的三地址代码生成

(Parser and code generator for a tiny programming language)

实验学时：8 学时

实验类型：综合程序设计

先修课程（含实验环节）：

高级语言程序设计及其课程设计、数据结构及其课程设计

使用专业：计算机科学与技术

一. 实验目的

掌握计算机语言的语法分析程序设计与属性文法应用的实现方法。

二. 实验内容

采用适当的方法（递归子程序法、LL 分析法、LR 分析法，使用 Yacc 等自动生成工具），实现一个能够进行语法分析并生成三地址代码的微型编译程序。

三. 实验要求

基本要求

1. 考虑下述语法制导定义中文法，采用适当的方法（递归子程序法、LL 分析法、LR 分析法，使用 Yacc 等自动生成工具），构造语法分析程序，要求输出语法树，或者按照最左派生的顺序输出派生的产生式序列；
2. 处理的源程序存放在文件中，它可以包含多个语句；
3. 分析程序可以对“源程序”中的一些错误做出适当的处理。

附加要求

（希望能够达到下列要求，这个要求属于实验三的内容）

4. 考虑下述语法制导定义中语义规则，改写语法分析程序，构造三地址代码生成程序；
5. 构建一个自动生成程序，生成实现 LR 分析的分析表，并利用此分析表实现

语法分析；

6. 分析程序可以对“源程序”中的一些错误做出更多地的处理。如：错误的定位，续编译，简单错误的纠正等。

产生式	语义规则
$P \rightarrow L$	$P.code = L.code$
$P \rightarrow LP_1$	$P.code = L.code \parallel P_1.code$
$L \rightarrow S ;$	$L.code = S.code$
$S \rightarrow id = E$	$S.code = E.code \parallel \text{gen}(id.place := 'E.place)$
$S \rightarrow \text{if } C \text{ then } S_1$	$C.true = \text{newlabel}; \quad C.false = S.next;$ $S_1.next = S.next;$ $S.code = C.code \parallel \text{gen}(C.true ':') \parallel S_1.code$
$S \rightarrow \text{if } C \text{ then } S_1$ $\text{else } S_2$	$C.true = \text{newlabel}; \quad C.false = \text{newlabel};$ $S_1.next = S_2.next = S.next;$ $S.code = C.code \parallel \text{gen}(C.true ':') \parallel S_1.code$ $\parallel \text{gen}(\text{'goto' } S.next) \parallel \text{gen}(C.false ':') \parallel S_2.code;$
$S \rightarrow \text{while } C \text{ do } S$	$S.begin = \text{newlabel}; \quad C.true = \text{newlabel};$ $C.false = S.next; \quad S_1.next = S.begin;$ $S.code = \text{gen}(S.begin ':') \parallel C.code \parallel$ $\text{gen}(C.true ':') \parallel S_1.code \parallel \text{gen}(\text{'goto' } S.begin);$
$S \rightarrow \{ P \}$	$S.code = P.code$
$C \rightarrow E_1 > E_2$	$C.code = E_1.code \parallel E_2.code \parallel$ $\text{gen}(\text{'if' } E_1.place > 'E_2.place' \text{ goto 'C.true' } \parallel$ $\text{gen}(\text{'goto' } C.false)$
$C \rightarrow E_1 < E_2$	$C.code = E_1.code \parallel E_2.code \parallel$ $\text{gen}(\text{'if' } E_1.place < 'E_2.place' \text{ goto 'C.true' } \parallel$ $\text{gen}(\text{'goto' } C.false)$
$C \rightarrow E_1 = E_2$	$C.code = E_1.code \parallel E_2.code \parallel$ $\text{gen}(\text{'if' } E_1.place = 'E_2.place' \text{ goto 'C.true' } \parallel$ $\text{gen}(\text{'goto' } C.false)$
$E \rightarrow E_1 + T$	$E.place = \text{newtemp};$ $E.code = E_1.code \parallel T.code \parallel$ $\text{gen}(E.place := 'E_1.place' + 'T.place)$
$E \rightarrow E_1 - T$	$E.place = \text{newtemp}; \quad E.code = E_1.code \parallel T.code \parallel$ $\text{gen}(E.place := 'E_1.place' - 'T.place)$
$E \rightarrow T$	$E.place = T.place; \quad E.code = T.code$
$T \rightarrow F$	$T.place = F.place; \quad T.code = F.code$
$T \rightarrow T_1 * F$	$T.place = \text{newtemp};$ $T.code = T_1.code \parallel F.code \parallel$ $\text{gen}(T.place := 'T_1.place' * 'F.place)$
$T \rightarrow T_1 / F$	$T.place = \text{newtemp}; \quad T.code = T_1.code \parallel F.code \parallel$ $\text{gen}(T.place := 'T_1.place' / 'F.place)$

F → (E)	F.place = E.place; F.code = E.code
F → id	F.place = id.name; F.code = ' '
F → int8	F.place = int8.value; F.code = ' '
F → int10	F.place = int10.value; F.code = ' '
F → int16	F.place = int16.value; F.code = ' '

四. 实验环境

PC;

DOS 操作系统或 Windows 操作系统;

Turbo C 程序集成环境或 Visual C++ 程序集成环境。

五. 实验步骤

1. 考虑给定的文法，消除左递归，提取左因子。
2. 编制并化简语法图。
3. 设计递归子程序的算法。
4. 编制各个递归子程序。
5. 连接实验一的词法分析函数 scan()，进行测试。
6. 设计三地址代码生成子系统的数据结构和算法。
7. 将各个递归子程序改写为代码生成函数。
8. 编制测试程序（main 函数）。
9. 调试程序：输入一个语句，检查输出的三地址代码

六. 测试数据示例

输入数据示例 1:

输入数据 1:

while (a3+15)>0xa do if x2 = 07 then while y<z do y = x * y / z; c=b*c+d;

输入数据 2:

while (a3+15)>0xa do

if x2 = 07 then

while y<z do

$y = x * y / z;$
 $c = b * c + d;$

正确结果：等效的三地址代码序列

L1: $t1 := a3 + 15$
 if $t1 > 10$ goto L2
 goto L0
L2: if $x2 = 7$ goto L3
 goto L1
L3: if $y < z$ goto L4
 goto L1
L4: $t2 = x * y$
 $t3 = t2 / z$
 $y = t3$
 goto L3
 goto L1
L0: $t4 = b * c$
 $t5 = t4 + d$
 $c = t5$

实验验收

1. 分组完成实验

- (1) 每组 3-4 人，设组长 1 名；
- (2) 每人必须参与总体设计，并分工完成其中一部分的设计与实现；
- (3) 组内要加强合作，时常在一起讨论设计与实现；
- (4) 届时每组得到一个基础分，按照每个成员的贡献在基础分上上下下浮动。

2. 实验内容要求

- (1) 根据实验题目和实验指导书的要求分工完成一个限定语言编译核心部分的设计与实现；
- (2) 具体包括：词法分析、语法分析、中间代码生成（语义）、适当的错误处理；
- (3) 具体的语法分析方法由各组自己选定；
- (4) 可以使用自动生成技术；
- (5) 更高要求：鼓励将“形式语言与自动机理论”中学到的知识和方法和本课程所学综合运用，构建适当的自动计算系统。……

3. 按组进行验收

- (1) 每组 10-15 分钟，面向全班介绍、演示所完成的系统（可以由一人主讲，其他补充，也可以按照分工进行报告）；
- (2) 各组的报告要有 PPT，要有实际运行演示；
- (3) 验收报告期间，分组集中就坐，以便给其他组评分；
- (4) 满分为 15 分；
- (5) 验收关键点
 - ① 词法分析：是否正确实现。
 - ② 语法分析：是否正确实现，实现用方法。
 - ③ 三地址码：是否正确实现。
 - ④ 多行多语句：是否正确实现。既可以接受多行语句的输入，且输入中含有多个语句。

- ⑤ 输入方式：文件输入，希望是通过文件输入被编译的源程序。用不鼓励终端输入。
- ⑥ 处理错误能力：是否具有较好的源程序的错误处理。例如，发现错误、错误的局部化、续编译等。
- ⑦ 实数：是否有能力处理实数。
- ⑧ 组织：组内协调好，各组员发挥了作用。
- ⑨ 创意：功能特殊的可以适当加分。
- ⑩ 表达与展示：现场报告准备充分、计划好，表达好。能够在限定的时间内突出本组所设计实现系统的精彩之处。

(6) 验收训练与成绩评价

在验收过程中，每组和授课教师一起各自给其他小组评分，以锻炼判别与评价能力。现场给出各项的 A、B、C 等，在验收课后的第二天，各组将认真讨论后给出的最终评分交给任课教师。

实验报告

实验报告由综合报告、源程序、可运行程序包 3 大部分组成，已完成小组为单位，在课程结束前通过 email 提交给授课教师。

(1) 综合报告

① 组内分工与贡献介绍

② 系统功能概述（包括系统的总体结构）

③ 包含实验指导书要求的内容分系统报告

✓ 词法分析子系统

- i. 词法的正规式描述
- ii. 变换后的正规文法
- iii. 状态图
- iv. 词法分析程序的主要数据结构与算法

- ✓ 如果采用自动生成技术，请描述采用的技术、相应的描述程序（如 LEX 程序）、辅助程序，以及生成的词法分析器的总体结构。

✓ 语法分析子系统

- i. 根据选择的语法分析方法进行描述。例如，如果采用递归子程序法，建议包括改写后的产生式集合，化简后的语法图。
- ii. 语法分析子系统结构
- iii. 语法分析子系统的主要数据结构与算法

- ✓ 如果采用自动生成技术，请描述采用的技术、相应的描述程序（如 YACC 程序）、辅助程序，以及生成的语法分析器的总体结构。

✓ 三地址代码生成器

- i. 语法制导定义。
- ii. 算法基本思想

④ 实验体会

(2) 源程序

- ① 源程序;
 - ② 实验输入文件典型示例;
 - ③ 对应的实验输出文件典型示例。
- (3) 可运行程序包，其中启动文件名为“***学院 20**年第 X 组.exe”

附：思考题

这些思考题作为实验设计与实现过程中要考虑的问题的示例，希望引导大家在实验过程中考虑相关的类似问题，并不是要求大家在实验报告后面像一个个简答题那样驻各地进行解答。相应的内容可以体现在综合报告的第四部分——实验体会中。

1. 词法分析能否采用空格来区分单词?
2. 程序设计中哪些环节影响词法分析的效率? 如何提高效率?
3. 生成的三地址代码可否直接输出(不采用数据结构来实现属性 code)?
4. 如何保证四则运算的优先关系和左结合性?
5. 如何采用代码段相对地址代替三地址代码序列中的标号?
6. 完成一个实验，通常要进行实验设计、实验实现、实验结果分析，你在本次实验中，是否都涉及的这三个类工作，在大学学习的这几年的实验中，是不是都或多或少设计了这三类工作?