实验 3 使用 Lex 自动生成扫描程序

实验难度：★★☆☆☆
建议学时：2 学时

一、 实验目的

- 掌握 Lex 输入文件的格式。
- 掌握使用 Lex 自动生成扫描程序的方法。

二、 预备知识

- 要求已经学习了正则表达式的编写方法，能够正确使用“*”、“?”、“+”等基本的元字符，并且学习了 Lex 程序中定义的特有的元字符，例如“[]”、“\n”等。
- 了解了标识符和关键字的识别方法。
- 本实验使用 Lex 的一个实现版本——GNU Flex 作为扫描程序。

三、 实验内容

3.1 阅读实验源代码

sample.txt 文件

在 sample.txt 文件中是一个使用 TINY 语言编写的小程序,这个小程序在执行时从标准输入（键盘）读取一个整数，计算其阶乘后显示到标准输出（显示器）。

在本实验中并不需要执行这个小程序，只需要使用 Lex 生成的扫描程序对这个 TINY 源代码文件进行扫描，最后统计出各种符号的数量。TINY 语言中的符号说明可以参考下面的表格。

序号	符号或语句	说明
1	{…}	在两个大括号之间的是注释。
2	read	从标准输入（键盘）读取数据。是一个关键字。
3	write	将数据写入标准输出(屏幕)。是一个关键字。
4	if … then… else…end	if 语句。if 的后面是一个布尔表达式，then 和 else 的后面是一个语句块，其中 else 是可选的。end 表示结束。包括了四个关键字。
5	repeat…until…	repeat 语句。repeat 后面是一个语句块，until 后面是一个布尔表达式。包括了两个关键字。
6	<、=	比较运算符。<是小于符号。=是等于符号
7	+、-、*、/	算数运算符。
8	:=	赋值运算符。
9	正整数	由数字 0-9 组成。
10	标识符	由大写字母和小写字母组成。
11	;	在语句的结束位置有一个分号。

define.h 文件

在此文件中定义了一个枚举类型，对应于 TINY 语言中的各种符号。注意，还包括了“文件结束”和“错误”，其中 yylex 函数在遇到文件结束时，默认会返回 0，所以将“文件结束”定义在开始位置，这样它的值也为 0。

scan.txt 文件

此文件是Lex的输入文件。根据Lex输入文件的格式,此文件分为三个部分(由%%分隔)，各个部分的说明可以参见下面的表格。

名称	说明
第一部分	● 在%{和%}之间的直接插入 C 源代码文件的内容。包括了要包含的头文件，以及 TINY 语言中各种符号的计数器，用于保存扫描的结果。 ● 定义了换行（newline）和空白（whitespace）的正则表达式。注意，空白包括了空格和制表符。
第二部分	包含了一组规则，当规则中的正则表达式匹配时，yylex 函数会执行规则提供的源代码。主要包含了下面的规则： ● 比较运算符、算数运算符等的规则。这些规则直接使用字符串进行匹配。若匹配成功，yylex 函数返回对应的枚举值。 ● 匹配换行的规则。匹配成功，就统计行数。 ● 匹配空白的规则。匹配成功，就忽略。 ● 由于编写注释的正则表达式比较复杂，所以在匹配注释的开始符号后，直接编写 C 代码来匹配注释的结束符号。这里用到的 input 函数是一个 Lex 的内部函数，它会返回一个输入字符。 ● 使用 Lex 正则表达式中的“.”元字符匹配其他的任何字符。当之前的规则均匹配失败时，就会在此规则匹配成功，从而返回错误类型。
第三部分	这部分中的内容会直接插入 C 源代码文件。此部分内容的说明可以参见下面的表格。

内容	说明
main 函数	主函数。这里用到了 C 语言定义的 main 函数的两个参数，这两个参数的用法可以参考函数的注释。 在 main 函数中首先使用 fopen 函数打开了待处理的文件，然后将此文件作为 Lex 扫描程序的输入（yyin），之后调用 yylex 函数开始扫描，并调用 stat 函数统计各种符号的数量，最后调用 output 函数输出统计的结果。
id2keyword 函数	此函数将标识符转换为对应的关键字类型，如果通过参数传入的字符串不能与任何关键字匹配，就仍然返回标识符类型。此函数的函数体还不完整，留给读者完成。
stat 函数	此函数根据 tt 参数传入的符号类型，增加符号类型对应的计数器。如果 tt 是标识符类型（ID），会首先调用 id2keyword 函数,尝试将标识符类型转换为对应的关键字类型。
output 函数	输出统计的结果。

main.c 文件

此文件默认是一个空文件。Lex 根据输入文件生成的 C 源代码会输出到此文件中。此文

件会包含自动生成的 yylex 函数的定义，此函数实现了与输入文件相对应的 DFA 表驱动。

3.2 生成项目

按照下面的步骤生成项目：

1. 按 Ctrl+Shift+B,在弹出的下拉列表中选择“生成项目”。

在生成的过程中，首先使用 Flex 程序根据输入文件 scan.txt 来生成 main.c 文件，然后，将 main.c 文件重新编译、链接为可以运行的可执行文件。

在生成的 main.c 文件中，尝试找到 scan.txt 文件中第一部分和第三部分 C 源代码插入的位置，并尝试查找 input 函数和 yylex 函数的定义，以及 yyin 和 yytext 等变量的定义。

3.3 运行项目

在没有对项目的源代码进行任何修改的情况下，按照下面的步骤运行项目：

1、 选择“Run”菜单中的“Run Without Debugging”（快捷键 Ctrl+F5）。

在启动运行时，会自动将 sample.txt 文件的名称作为参数传给可执行文件（即 main 函数中 argv[1]指向的字符串,在 lauch.json 文件中的"configurations"配置中的"args"的属性值中进行了设置，读者也可以将其值更改其它文件名称，例如：readfile1.txt），所以程序运行完毕后，会在 Windows 控制台窗口中显示对 sample.txt 文件的扫描结果，如图 3-1 所示。

```
if: 0
then: 0
else: 0
end: 0
repeat: 0
until: 0
read: 0
write: 0
id: 0
num: 0
assign: 3
eq: 1
lt: 1
plus: 0
minus: 1
times: 1
over: 0
lparen: 0
rparen: 0
semi: 4
comment: 4
error: 55
line: 14
```

图 3-1：对 sample.txt 文件的扫描结果。

由于此时还没有在 scan.txt 中添加能够与标识符和正整数匹配的正则表达式，所以 id 和 num 的值均为 0，而标识符和正整数会被匹配为错误，所以，error 的数量大于 0。

3.4 添加标识符和正整数的统计功能

按照下面的步骤完成此练习：

1. 在 scan.txt 文件的第一部分添加标识符和正整数的正则表达式。
2. 在 scan.txt 文件的第二部分添加标识符和正整数的正则表达式匹配时的 C 源代码。

注意，标识符的正则表达式也用来匹配所有的关键字，不要直接使用字符串来逐个的匹配关键字。

3. 按 Ctrl+Shift+B,在弹出的下拉列表中选择“生成项目”。如果生成失败，根据“TERMINAL”窗口中的提示信息修改源代码中的语法错误。
4. 按 Ctrl+F5 启动执行项目。

执行的结果应该如图 3-2 所示，已经可以正确统计出标识符和正整数的数量。注意，由于此时 id2keyword 函数还无法将标识符转换为关键字，所以，所有的关键字都匹配成了标识符，关键字的数量都为 0。

```
if: 0
then: 0
else: 0
end: 0
repeat: 0
until: 0
read: 0
write: 0
id: 17
num: 4
assign: 3
eq: 1
lt: 1
plus: 0
minus: 1
times: 1
over: 0
lparen: 0
rparen: 0
semi: 4
comment: 4
error: 0
line: 14
```

图 3-2：正确统计出标识符和正整数的数量。

3.5 添加关键字的统计功能

按照下面的步骤完成此练习：

1. 为 id2keyword 函数编写源代码。要求使用此函数前面定义的 key_table 表格中的数据，通过线性搜索的方式，根据标识符的字符串确定其对应的关键字类型。
2. 按 Ctrl+Shift+B,在弹出的下拉列表中选择“生成项目”。如果生成失败，根据“TERMINAL”窗口中的提示信息修改源代码中的语法错误。
3. 按 Ctrl+F5 启动执行项目。

执行的结果应如图 3-3 所示，已经可以正确统计出各个关键字的数量。

注意：

1、 如果执行的结果不正确，可以通过添加断点和单步调试的方法来查找错误的原因。

2、 断点应该添加在 scan.txt 文件中需要中断的 C 源代码行,不要添加在 main.c 文件中，否则无法命中断点。

图 3-3：正确统计出各个关键字的数量。

3.6 添加 C 语言风格的注释

按照下面的步骤完成此练习：

1. 将 sample.txt 文件中的多行注释包括在“/*”和“*/”之间，将语句末尾的注释放在“//”的后面。
2. 修改 scan.txt 文件，使之能够正确匹配和统计这两种 C 语言风格的注释。
3. 按 Ctrl+Shift+B,在弹出的下拉列表中选择“生成项目”。如果生成失败，根据“TERMINAL”窗口中的提示信息修改源代码中的语法错误。
4. 按 Ctrl+F5 启动执行项目，确保统计的注释数量是正确的。

3.7 通过验证

按照下面的步骤继续实验：

1. 按 Ctrl+Shift+B，然后在下拉列表中选择“测试”，启动验证。在“TERMINAL”窗口中会显示验证的结果。如果验证失败，可以在“EXPLORER”窗口中，右击“result_comparation.html”文件，在弹出的菜单中选择“Open Preview”，可以查看用于答案结果文件与读者编写程序产生的结果文件的不同之处，从而准确定位导致验证失败的原因。

四、思考与练习

1. 修改 key_table 表格中数据的顺序，使关键字按照字母顺序排列，然后修改 id2keyword 函数中的代码，使用二分法从 key_table 表格中查找关键字。
2. 使用 gperf 工具为 TINY 语言的关键字生成杂凑表(哈希表)，然后修改 id2keyword 函数中的代码从杂凑表中查找关键字。