中山大学数据科学与计算机学院本科生实验报告

课程名称：编译器构造实验 任课教师：陈炬桦 教学助理（TA）：

|  |  |  |  |
| --- | --- | --- | --- |
| 学年度 | 2018-2019 | 学期 | 第二学期 |
| 年级 | 2016级 | 专业（方向） | 计算机科学与技术 |
| 学号 | 16337233 | 姓名 | 王凯祺 |
| 电话 | 13434386182 | Email | [Wangkq3@mail2.sysu.edu.cn](mailto:Wangkq3@mail2.sysu.edu.cn) |
| 开始日期 | 2019/3/29 | 完成日期 | 2019/3/29 |

1. 实验题目

**词法分析程序设计**

**1.1 Description**

设一语言的关键词、运算符、分界符的个数与单词如下：   
struct { int number; string str[10]; } keywords={3,"int","main","return"} ; //关键词  
struct { int number; string str[10]; } operators ={5,"+","\*","=","+=","\*="}; //运算符  
struct { int number; string str[10]; } boundaries ={6,"(",")","{","}",",",";"} ; //分界符  
struct { int number; string str[100];} identifieres={0}; //标识符  
struct { int number; string str[100];} Unsigned\_integer={0}; //无符号整数  
以上类号分别为1~5，序号从0开始；  
标识符是字母开头的字母数字串；常量为无符号整数；  
用C++设计一程序实现词法分析。

**1.2 Input**

输入一程序，结束符用”#”；

**1.3 Output**

输出单词数对：<类号，序号>。 输出标识符表，用空格分隔； 输出无符号整数表，用空格分隔；

2. 算法描述(介绍程序模块功能；流程图)

(1) 将关键词、运算符、分界符加入map。

(2) 写一个循环结构，不断地读字符。若读到的第一个字符是字母，则接下来一直读到不是字母/数字为止；若读到的第一个字符是数字，则接下来读到不是数字为止；若读到的第一个字符是运算符，则接下来读到不是运算符为止；若读到的第一个字符是分界符，则只读这个字符。先判断它是不是关键词、运算符、分界符，若是，则直接输出；否则，根据字符串的第一个字符决定是标识符（首位为字母）还是数字（首位为数字），并把该字符串加入标识符/数字集合。

(3) 最后将标识符/数字集合输出。

3. 测试数据

样例输入

main()

{ int a=2,b=3;

return 2\*b+a;

}#

样例输出

<1,1><3,0><3,1><3,2><1,0><4,0><2,2><5,0><3,4><4,1><2,2><5,1><3,5><1,2><5,0><2,1>

<4,1><2,0><4,0><3,5><3,3>

identifieres:a b

Unsigned\_integer:2 3

4. 程序清单

#include <bits/stdc++.h>

using namespace std;

struct { int number; string str[10]; } keywords={3,"int","main","return"} ;

struct { int number; string str[10]; bool in(char x) { for (int i = 0; i < number; ++i) if (str[i][0] == x) return true; return false; } } operators ={5,"+","\*","=","+=","\*="};

struct { int number; string str[10]; bool in(char x) { for (int i = 0; i < number; ++i) if (str[i][0] == x) return true; return false; } } boundaries ={6,"(",")","{","}",",",";"} ;

struct { int number; string str[100];} identifieres={0}; //鏍囪瘑绗?struct { int number; string str[100];} Unsigned\_integer={0};

map<string, pair<int, int> > mp;

int main() {

for (int i = 0; i < keywords.number; ++i) mp[keywords.str[i]] = make\_pair(1, i);

for (int i = 0; i < operators.number; ++i) mp[operators.str[i]] = make\_pair(2, i);

for (int i = 0; i < boundaries.number; ++i) mp[boundaries.str[i]] = make\_pair(3, i);

for (int i = 0; i < identifieres.number; ++i) mp[identifieres.str[i]] = make\_pair(4, i);

for (int i = 0; i < Unsigned\_integer.number; ++i) mp[Unsigned\_integer.str[i]] = make\_pair(5, i);

char w = getchar();

while (1) {

string str = "";

while (w <= 32) w = getchar();

if (w == '#') break;

pair<int, int> newans;

if (isalpha(w)) {

while (isalpha(w) || isdigit(w)) {

str += w;

w = getchar();

}

newans = make\_pair(4, identifieres.number);

} else if (boundaries.in(w)) {

str += w;

w = getchar();

} else if (operators.in(w)) {

while (operators.in(w)) {

str += w;

w = getchar();

}

} else if (isdigit(w)) {

while (isdigit(w)) {

str += w;

w = getchar();

}

newans = make\_pair(5, Unsigned\_integer.number);

}

if (mp.find(str) != mp.end()) {

pair<int, int> ans = mp[str];

printf("<%d,%d>", ans.first, ans.second);

} else {

if (newans.first == 4) identifieres.str[identifieres.number++] = str;

if (newans.first == 5) Unsigned\_integer.str[Unsigned\_integer.number++] = str;

mp[str] = newans;

printf("<%d,%d>", newans.first, newans.second);

}

}

puts("");

printf("identifieres:");

for (int i = 0; i < identifieres.number; ++i) printf("%s%s", identifieres.str[i].c\_str(), i == identifieres.number - 1 ? "\n" : " ");

printf("Unsigned\_integer:");

for (int i = 0; i < Unsigned\_integer.number; ++i) printf("%s%s", Unsigned\_integer.str[i].c\_str(), i == Unsigned\_integer.number - 1 ? "\n" : " ");

}