



SHANGHAI UNIVERSITY

**<编译原理>实验报告**

|  |  |
| --- | --- |
| **学 院** | **计算机工程与科学学院** |
| **组 号** | **10** |
| **实验题号** | **二** |
| **日期** | **2023.04.06** |

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 主要工作 | 贡献因子 |
| 20124652 | 徐驰 | 代码优化，使之优美 | 0.25 |
| 20124698 | 陈世锦 | 代码构思，初步实现 | 0.25 |
| 20124699 | 李子浩 | 报告撰写，代码测试 | 0.25 |
| 20124703 | 邵启承 | 代码优化，使之高效 | 0.25 |

**<编译原理>实验报告**

**一、实验目的与要求**

* 根据 PL/0 语言的文法规范，编写 PL/0 语言的词法分析程序。
* 通过设计调试词法分析程序，实现从源程序中分出各种单词的方法加深对课堂教学的理解；提高词法分析方法的实践能力。 
* 掌握从源程序文件中读取有效字符的方法和产生源程序的内部表示文件的方法。
* 掌握词法分析的实现方法。 
* 上机调试编出的词法分析程序。

**二、实验环境**

微机 CPU P4 以上，256M 以上内存，安装好 C 语言，或 C++，或 Visual C++

**三、实验内容**

输入 PL/0 语言程序，输出程序中各个单词符号（关键字、专用符号以及其它标记）。

**四、实验内容的设计与实现**

**说明：**选择本实验中最有程序设计技巧或特色的并具有独立功能的源代码片断，并对其算法的技巧或特色进行必要的文字说明。

1. **头文件说明**

#include<bits/stdc++.h> 常用的头文件，包含了一些常用的标准库头文件。

1. **读入说明**

首先按行读入，然后对字符进行切分，并对读入的每个英文字符都转为小写，以实现不区分大小写。然后对读入的每一个字符进行判断其是否为标识符、表格或关键字。

下面为主函数示例以及详细说明：

1. int main()
2. {
3. ifstream infile("in.txt");
4. ofstream outfile("out.txt");
5. string str,word;
6. while(infile>>word) str = str+' '+word;
7. for(int i=0;i<str.length();i++)
8. {
9. if(str[i]==' ' || str[i]=='\n') continue;
10. *//识别常数*
11. else if(isdigit(str[i]))
12. {
13. string digit;
14. while(isdigit(str[i]))
15. {
16. digit += str[i];
17. i++;
18. }
19. i--;
20. print\_digit(digit,outfile);
21. }
22. *//识别基本字或标识符*
23. else if(isalpha(str[i]))
24. {
25. string letter;
26. while(isalnum(str[i]))
27. {
28. letter += str[i];
29. i++;
30. }
31. i--;
32. reshape(letter);
33. print\_alpha(letter,outfile);
34. }
35. *//识别运算符*
36. else
37. {
38. print\_others(str[i],str[i+1],outfile);
39. string cnt;
40. cnt += str[i];
41. cnt += str[i+1];
42. if(cnt==">="||cnt=="<="||cnt==":="||cnt=="<>") i++;
43. }
44. }
45. infile.close();
46. outfile.close();
47. cout<<"\nResult Saved."<<endl;
48. return 0;
49. }

* 文件操作

**ofstream outfile("out.txt")** 创建了一个输出文件流对象，将输出结果写入到名为 "out.txt" 的文件中。**ifstream infile("in.txt")** 创建了一个输入文件流对象，读取输入数据来自名为 "in.txt" 的文件。

**while (infile>>word)** 通过输入文件流对象从文件中循环读取数据，以空格为分隔，存储在字符串变量 str 中。

1. **标识符判断及统计**

为了对字符串进行分类，可以通过其开头的特征来判断其属于哪种类型。如果开头是符号，则其必定不是数字或字母；如果是关键字，则整个字符串必须与对应的关键字匹配；如果是数字，则开头必须是数字；如果是标识符，则其开头必须是字母且不是关键字。可以通过一个函数对标识符进行分类输出。

关键字和标识符输出函数将字符串将字符串和所有关键字对比，如果不符合则说明是标识符

1. void print\_alpha(string str,ofstream& out) *//关键字及标识符*
2. {
3. if(str=="begin"){
4. cout<<"(beginsym,begin)"<<endl;
5. out<<"(beginsym,begin)"<<endl;
6. }
7. else if(str=="call"){
8. cout<<"(callsym,  call)"<<endl;
9. out<<"(callsym,  call)"<<endl;
10. }
11. else if(str=="const"){
12. cout<<"(constsym,const)"<<endl;
13. out<<"(constsym,const)"<<endl;
14. }
15. else if(str=="do"){
16. cout<<"(dosym,      do)"<<endl;
17. out<<"(dosym,      do)"<<endl;
18. }
19. else if(str=="end"){
20. cout<<"(endsym,    end)"<<endl;
21. out<<"(endsym,    end)"<<endl;
22. }
23. else if(str=="if"){
24. cout<<"(ifsym,      if)"<<endl;
25. out<<"(ifsym,      if)"<<endl;
26. }
27. else if(str=="odd"){
28. cout<<"(oddsym,    odd)"<<endl;
29. out<<"(oddsym,    odd)"<<endl;
30. }
31. else if(str=="procedure"){
32. cout<<"(proceduresym,procedure)"<<endl;
33. out<<"(proceduresym,procedure)"<<endl;
34. }
35. else if(str=="read"){
36. cout<<"(readsym,  read)"<<endl;
37. out<<"(readsym,  read)"<<endl;
38. }
39. else if(str=="then"){
40. cout<<"(thensym,  then)"<<endl;
41. out<<"(thensym,  then)"<<endl;
42. }
43. else if(str=="while"){
44. cout<<"(whilesym,while)"<<endl;
45. out<<"(whilesym,while)"<<endl;
46. }
47. else if(str=="var"){
48. cout<<"(varsym,    var)"<<endl;
49. out<<"(varsym,    var)"<<endl;
50. }
51. else if(str=="write"){
52. cout<<"(writesym,write)"<<endl;
53. out<<"(writesym,write)"<<endl;
54. }
55. else{
56. cout<<"(ident,"<<setw(7)<<str<<" )"<<endl;
57. out<<"(ident,"<<setw(7)<<str<<" )"<<endl;
58. }
59. return;
60. }

**数字输出**

1. void print\_digit(string str,ofstream& out){ *//数字*
2. cout<<"(number,"<<setw(6)<<str<<" )"<<endl;
3. out<<"(number,"<<setw(6)<<str<<" )"<<endl;
4. return;
5. }

**其他符号输出**

1. void print\_others(char str,char str1,ofstream& out){ *//其他字符*
2. if(str=='+'){
3. cout<<"(plus,       + )"<<endl;
4. out<<"(plus,       + )"<<endl;
5. }
6. else if(str=='-'){
7. cout<<"(minus,      - )"<<endl;
8. out<<"(minus,      - )"<<endl;
9. }
10. else if(str=='\*'){
11. cout<<"(times,      \* )"<<endl;
12. out<<"(times,      \* )"<<endl;
13. }
14. else if(str=='/'){
15. cout<<"(slash,      / )"<<endl;
16. out<<"(slash,      / )"<<endl;
17. }
18. else if(str=='='){
19. cout<<"(eql,        = )"<<endl;
20. out<<"(eql,        = )"<<endl;
21. }
22. else if(str=='<'){
23. if(str1=='>'){
24. cout<<"(neq,       <> )"<<endl;
25. out<<"(neq,       <> )"<<endl;
26. }
27. else if(str1=='='){
28. cout<<"(leq,       <= )"<<endl;
29. out<<"(leq,       <= )"<<endl;
30. }
31. else{
32. cout<<"(lss,        < )"<<endl;
33. out<<"(lss,        < )"<<endl;
34. }
35. }
36. else if(str=='>'){
37. if(str1=='='){
38. cout<<"(geq,       >= )"<<endl;
39. out<<"(geq,       >= )"<<endl;
40. }
41. else{
42. cout<<"(gtr,        > )"<<endl;
43. out<<"(gtr,        > )"<<endl;
44. }
45. }
46. else if(str==':'){
47. if(str1=='='){
48. cout<<"(becomes,   := )"<<endl;
49. out<<"(becomes,   := )"<<endl;
50. }
51. }
52. else if(str=='('){
53. cout<<"(lparen,     ( )"<<endl;
54. out<<"(lparen,     ( )"<<endl;
55. }
56. else if(str==')'){
57. cout<<"(rparen,     ) )"<<endl;
58. out<<"(rparen,     ) )"<<endl;
59. }
60. else if(str==','){
61. cout<<"(comma,      , )"<<endl;
62. out<<"(comma,      , )"<<endl;
63. }
64. else if(str==';'){
65. cout<<"(semicolon,  ; )"<<endl;
66. out<<"(semicolon,  ; )"<<endl;
67. }
68. else if(str=='.'){
69. cout<<"(period,     . )"<<endl;
70. out<<"(period,     . )"<<endl;
71. }
72. else{
73. cout<<"error"<<endl;
74. out<<"error"<<endl;
75. }
76. return;
77. }
78. **文件输出**
79. infile.close();
80. outfile.close();
81. cout<<"\nResult Saved."<<endl;
82. **测试用例与实验结果**

由于结果过长不方便截图展示，此处仅展示样例1的输入与输出，其余样例请移步样例文件夹查看。

**测试用例：**

文本

描述已自动生成

**运行结果：**

文本

中度可信度描述已自动生成

文本

中度可信度描述已自动生成

**五、收获与体会**

每个同学完成该实验后的收获和体会。

徐驰：这次实验让我更加深入地了解了编译原理中的词法分析，之前只是了解了大概的原理和流程，但通过本次实验的实践，我更加直观地感受到了词法分析器的重要性和实现的难度。同时，也意识到了编写高效的词法分析器需要对语言的语法和结构有深入的理解和掌握。

陈世锦：本次实验让我更加重视代码的规范和可读性，词法分析器的实现需要涉及到大量的字符串处理和正则表达式匹配，如果代码不规范、不清晰，就会导致代码难以维护和调试。因此，我认为在编写代码的同时，也要注重代码的可读性和可维护性，这对于长期的软件开发来说非常重要。

李子浩：通过本次实验，我深刻地认识到了编译原理的重要性和应用价值。编译器是现代计算机系统中不可或缺的一部分，它们负责将高级语言翻译成机器语言，使得程序员可以更加高效地编写程序。同时，也意识到编译器的开发需要多个领域的知识和技能，包括计算机体系结构、算法和数据结构、语言学等多个方面。

邵启承：本次实验让我更加深入地了解了 PL/0 语言的语法和结构，学习了如何编写词法分析器并将其应用到实际的程序中。同时，也意识到编写高效的词法分析器需要对字符串处理和正则表达式等方面有深入的了解和掌握。我认为这些知识和技能对于我的未来发展和工作都有着重要的意义，我会继续学习和探索相关的领域。

**2023.04.06**