

c 语言词法分析器

一、项目说明

1. 项目使用

本实验环境采用带桌面的 Ubuntu Linux 环境，实验中会用到桌面上的程序：

- LX 终端 (LXTerminal) : Linux命令行终端，打开后会进入Bash环境，可以使用Linux命令
- GVim：非常好用的编辑器，最简单的用法可以参考课程 Vim编辑器 (<http://www.shiyanlou.com/courses/2>)。

2. 环境使用

使用 GVim 编辑器输入实验所需的代码及文件，使用 LX 终端 (LXTerminal) 运行所需命令进行操作。

完成实验后可以点击桌面上方的“实验截图”保存并分享实验结果到微博，向好友展示自己的学习进度。实验楼提供后台系统截图，可以真实有效证明您已经完成了实验。

实验记录页面可以在“我的主页”中查看，其中含有每次实验的截图及笔记，以及每次实验的有效学习时间（指的是在实验桌面内操作的时间，如果没有操作，系统会记录为发呆时间）。这些都是您学习的真实性证明。

4. 项目简介

项目目的：设计并实现一个包含预处理功能的词法分析程序，加深对编译中词法分析过程的理解。

项目要求：

1、实现预处理功能 源程序中可能包含有对程序执行无意义的符号，要求将其剔除。首先编制一个源程序的输入过程，从键盘、文件或文本框输入若干行语句，依次存入输入缓冲区（字符型数据）；然后编制一个预处理子程序，去掉输入串中的回车符、换行符和跳格符等编辑性文字；把多个空白符合并为一个；去掉注释。

2、实现词法分析功能 输入：所给文法的源程序字符串。输出：二元组构成的序列。具体实现时，可以将单词的二元组用结构进行处理。

3、待分析的C语言子集的词法 1) 关键字 main if then while do static int double struct break else long switch case typedef char return const float short continue for void default sizeof do
所有的关键字都是小写。

2) 运算符和界符 " + - * / : := < <> <= > >= = ; () # "

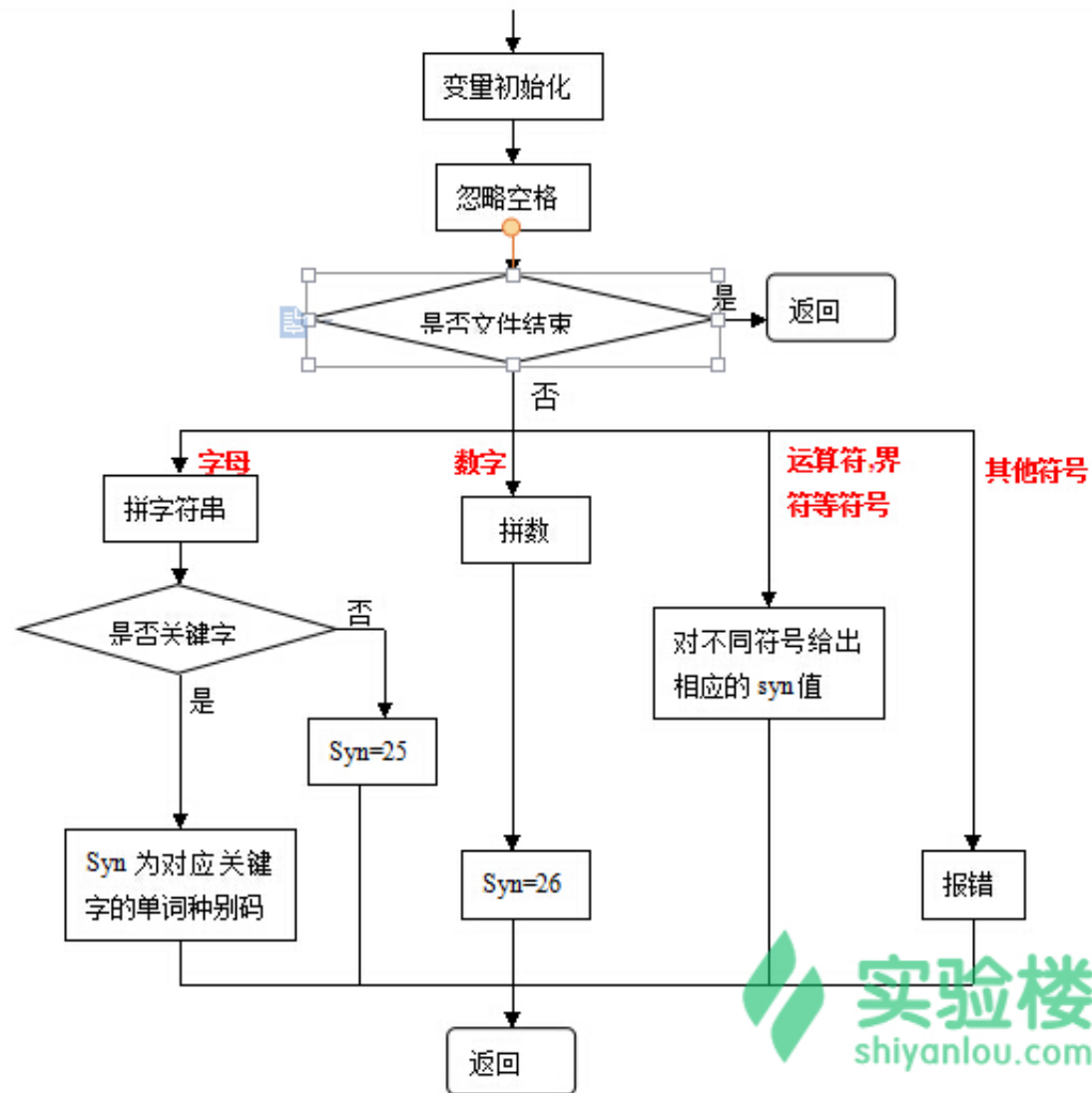
3) 其他标记ID和NUM 通过以下正规式定义其他标记：ID→letter(letter|digit)
NUM→digit digit letter→a|...|z|A|...|Z digit→0|...|9...

4) 空格由空白、制表符和换行符组成 空格一般用来分隔ID、NUM、专用符号和关键字，词法分析阶段通常被忽略。

4、各种单词符号对应的种别码

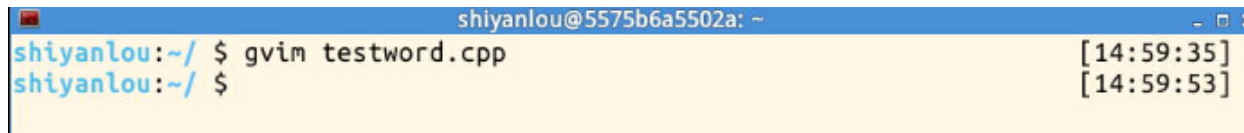
单词符号	种别码	单词符号	种别码
main	1	:	41
if	2	(42
then	3)	43
while	4	int	7
do	5	double	8
static	6	<u>struct</u>	9
ID	25	break	10
NUM	26	else	11
+	27	long	12
-	28	switch	13
*	29	case	14
/	30	<u>typedef</u>	15
:	31	char	16
:=	32	return	17
<	33	<u>const</u>	18
<>	34	float	19
<=	35	short	20
>	36	continue	21
>=	37	for	22
=	38	void	23
default	39	<u>sizeof</u>	24
do	40	#	0

功能流程图（代码实现思路基本根据流程图来的）：



二、项目实现

实践出真知，只有在实践过程中才能发现不足，现在让我们打开命令行，键入命令新建一个 .cpp 的文本：



```
shiyanolou@5575b6a5502a: ~  
shiyanolou:~/ $ gvim testword.cpp [14:59:35]  
shiyanolou:~/ $ [14:59:53]
```

从流程图可以知道，我们需要判断字符是否为数字、字母、定界符、关键字 我们通过 IsDigit()、IsLetter()、IsSymbol()、IsKeyword() 去实现这四个功能

第一个函数 IsDigit()

```
//判断是否为数字  
bool IsDigit(char ch)  
{  
    if(ch>='0'&&ch<='9')  
        return true;  
    return false;  
}
```

第二个函数 IsLetter()

```
//判断是否为字母  
bool IsLetter(char ch)  
{  
    if((ch>='a'&&ch<='z')||(ch>='A'&&ch<='Z'))  
        return true;  
    return false;  
}
```

第三个函数 IsSymbol()

```
//判断是否为定界符等
int IsSymbol(char ch)
{
    for(int i=0; i<9; i++)
    {
        if(ch==symbol[i])
            return i;
    }
    return -1;
}
```

第四个函数 IsKeyword()

```
//判断是否为关键字
int IsKeyword(string str)
{
    for(int i=0; i<26; i++)
    {
        if(str==keyword[i])
        {
            return i;
        }
    }
    return 25;
}
```

读者仔细阅读上面项目要求会发现，还有一个预处理的要求，需要合并空格，去掉注释的功能，下面我们就来完成合并空格的功能。

```
//空格处理
void HandleSpace(char a[])
{
    int j=0;
    memset(word,0,255);
    temp=false;
    for(int i=0; i<strlen(a); i++)
    {
        if(a[i]!=' ' && a[i]!='\t') //'\t'是table键
        {
            word[j++]=a[i];
            temp=false;
        }
        else
        {
            if(!temp&&a[i]!='\t')
            {
                word[j++]=a[i];
                temp=true;
            }
        }
    }
}
```

然后是处理注释，这里我是将 // 注释进行了预处理， /* */ 注释是在主程序中处理的>

```
//处理"/"注释
void prePro()
{
    int j=0;
    memset(tempstr,0,255);
    for(int i=0; i<strlen(word); i++)
    {
        if(word[i]=='/'&&word[i+1]=='/')
        {
            while(i<strlen(word))
            {
                i++;
            }
        }
        else {
            tempstr[j++]=word[i];
        }
    }
}
```

这样整个程序的核心大部分就完成了，思路就是判断读入的第一个单词是否为字母，若为字母，则为关键字或者标识符，若为数字则为 NUM。

三、完整源码

整个程序的源代码如下:


```
/*
*author:leetao
*contact:leetao94cn@gmail.com
*/
#include<iostream>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
using namespace std;

//存放处理后的字符串
char tempstr[255]={};
//空格标志
bool temp=false;
//临时数组
char word[255]={};
//keyword关键字
string keyword[26]={
    "main","if","then","while","do","static","default","do","int","double",
    "struct","break","else","long","switch","case","typedef","char","return",
    "const","float","short","continue","for","void","sizeof"};

int keyword_num[26]={1,2,3,4,5,6,39,40,7,8,9,10,11,
    12,13,14,15,16,17,18,19,20,21,22,23,24};
//部分运算符，定界符等
char symbol[9]={'+', '-', '*', '/', '=', ';', '(', ')', '#'};
//对应的种码值
int symbol_num[9]={27,28,29,30,38,41,42,43,0};

//判断是否为字母
bool IsLetter(char ch)
```

```
{
    if((ch>='a'&&ch<='z')||(ch>='A'&&ch<='Z'))
        return true;
    return false;
}

//判断是否为数字
bool IsDigit(char ch)
{
    if(ch>='0'&&ch<='9')
        return true;
    return false;
}

//判断是否为定界符等
int IsSymbol(char ch)
{
    for(int i=0; i<9; i++)
    {
        if(ch==symbol[i])
            return i;
    }
    return -1;
}

//判断是否为关键字
int IsKeyword(string str)
{
    for(int i=0; i<26; i++)
    {
        if(str==keyword[i])
        {
            return i;
        }
    }
}
```

```
    }
    //不是关键字即为ID
    return 25;
}

//空格处理
void HandleSpace(char a[])
{
    int j=0;
    memset(word,0,255);//需要清空，不然可能残留上次的字符串
    temp=false;
    for(int i=0; i<strlen(a); i++)
    {
        if(a[i]!=' ' && a[i]!='\t')
        {
            word[j++]=a[i];
            temp=false;
        }
        else
        {
            if(!temp&&a[i]!='\t')
            {
                word[j++]=a[i];
                temp=true;
            }
        }
    }
}

//处理"//"注释
void prePro()
```

```
{
    int j=0;
    memset(tempstr,0,255);
    for(int i=0;    i<strlen(word);    i++)
    {
        if(word[i]=='/'&&word[i+1]=='/')
        {
            while(i<strlen(word))
            {
                i++;
            }
        }
        else {
            tempstr[j++]=word[i];
        }
    }
}

int main()
{
    char instr[255]={}; //接收输入字符串
    bool flag=false; //多行注释标志,false为未处于注释区域
    string Token;//存放字符串
    char *str=NULL;//存放每行的字符串
    char delims[]=" ";//分割标志
    freopen("test.cpp","r",stdin);
    freopen("result.txt","w",stdout); //此行注释后，控制台输出，
    //否则文本输出
    while((gets(instr))!=NULL)
    {
        HandleSpace(instr);
        prePro();
    }
}
```

```
str=strtok(tempstr,delims);//分割字符串

while(str!=NULL)
{
    //头文件，宏定义
    if(*(str)=='#')
    {
        printf("#\n");
        break;
    }

    for(int i=0; i<strlen(str);i++)
    {
        if(*(str+i)=='/')
        {
            {
                if(*(str+i+1)=='*')
                {
                    flag=true;
                    break;
                }
            }
            //注释处理：*/,注释区域结束
            if(*(str+i)=='*'&&flag)
            {
                if(*(str+i+1)=='/')
                {
                    flag=false;
                    i++;
                    break;
                }
            }
            //标识符，关键词
            if(IsLetter(*(str+i))&&(!flag))
            {
```

```

//          printf("进入标识符判断\n");
while(IsLetter(*(str+i))||IsDigit(*(str+i))
||*(str+i)=='_')
    {
        Token+=*(str+i);
        i++;
    }

if(IsKeyword(Token)!=25)
    {
        printf("%s---->%d\n",Token.c_str(),
            keyword_num[IsKeyword(Token)]);
    }
    else printf("%s---->25\n",Token.c_str());

    Token="";
//          printf("退出标识符判断\n");
    }
    if(IsDigit(*(str+i))&&(!flag))
    {
//          printf("进入数字判断\n");
        while(IsDigit(*(str+i)))
        {
            Token+=*(str+i);
            i++;
        }
        printf("%s----->26\n",Token.c_str());
        Token="";
    }

//<,<=,<>
if(*(str+i)=='<'&&(!flag))
    {
        if(*(str+i)=='=')    {printf("<----->3

```

```

5\n");i++;}

4\n");i++;}

2\n");}

        if(*(str+i)=='>')    {printf("<>----->3

        else printf("<----->33\n");
    }
    //>,>=
    else if(*(str+i)=='>'&&!flag))
    {
        if(*(str+i+1)=='=') {printf(">----->37\n");}
        else printf(">-----36\n");
    }
    //:,:=
    else if(*(str+i)==' ':'&&!flag))
    {
        if(*(str+i+1)=='=') {printf(":=----->3

        else printf(":----->31\n");
    }
    //余下定界符等
    else if(IsSymbol(*(str+i))!=-1&&!flag))
    {
        printf("%c----->%d\n",*(str+i),
            symbol_num[IsSymbol(*(str+i))]);
    }
    }
    str=strtok(NULL,delims);
}

return 0;
}

```

这个代码完成了我们还需要一个程序,在当前目录下使用命令行：gvim test.cpp新建一个测试程序test.cpp文件代码如下(读者也可以自行发挥):

```
#include<stdio.h>
int main()
{
    //test

    /* test */
    for(int i=1;i<0;i++)
        printf("%d",i);

    return 0;
}
```

```
test.cpp + (~/.C-C-) - GVIM
文件(F) 编辑(E) 工具(T) 语法(S) 缓冲区(B) 窗口(W) Plugin 帮助(H)

1 #include<stdio.h>
2 int main()
3 {
4     // test
5     /* test */
6     for(int i=0; i<10; i++)
7         printf("%d",i);
8     return 0;
9 }
```

四、编译运行

自此准备工作都完成了，现在开始编译了：

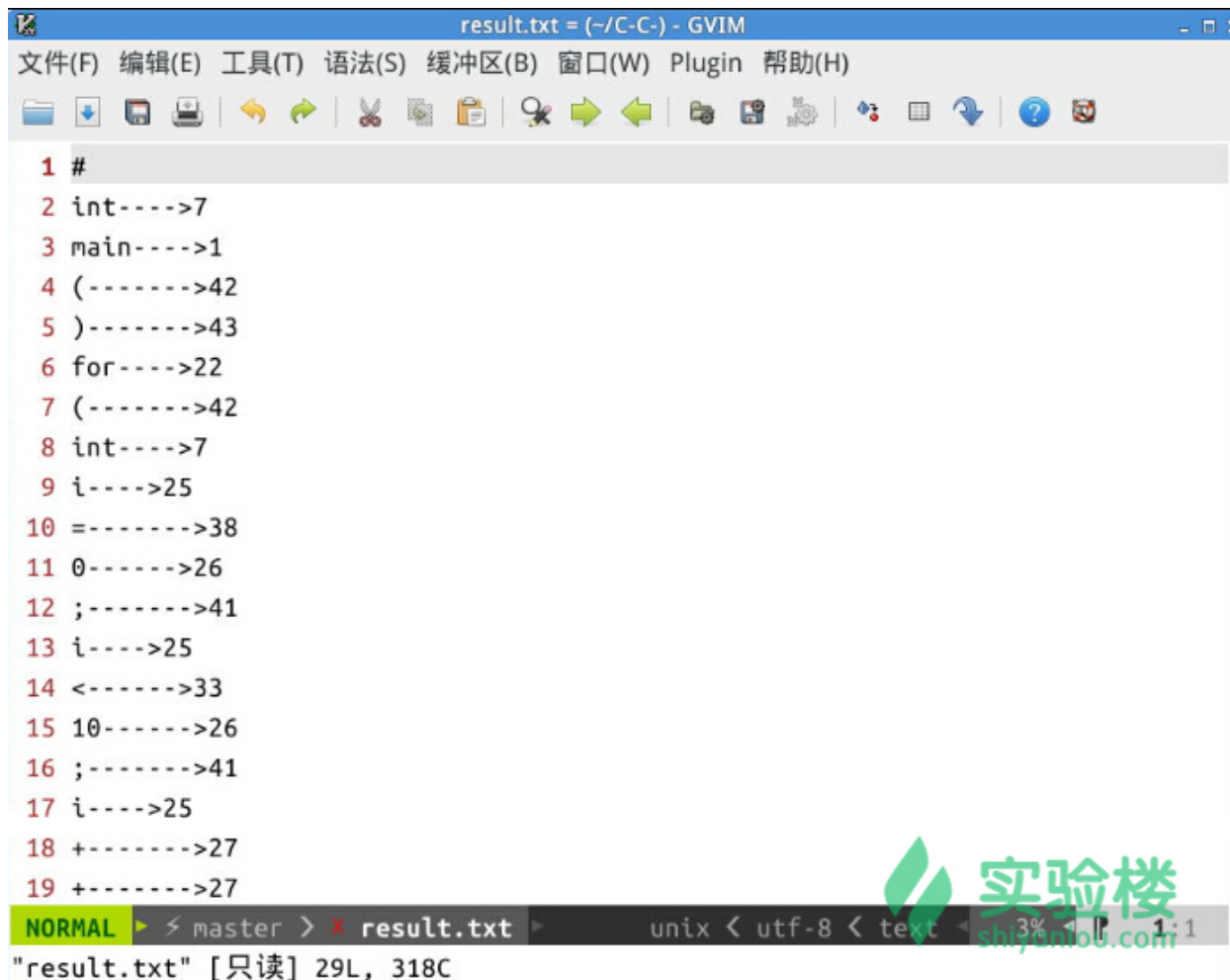

```
g++ testword.cpp -o testword
```

```
shianlou:C-C-/ (master*) $ g++ testword.cpp -o testword [15:16:03]
testword.cpp: In function 'int main()':
testword.cpp:128:9: warning: 'char* gets(char*)' is deprecated (declared at /usr
/include/stdio.h:638) [-Wdeprecated-declarations]
    while((gets(instr))!=NULL)
            ^
testword.cpp:128:19: warning: 'char* gets(char*)' is deprecated (declared at /us
r/include/stdio.h:638) [-Wdeprecated-declarations]
    while((gets(instr))!=NULL)
                  ^
/tmp/cccQ08fl.o: 在函数'main'中:
testword.cpp:(.text+0x90a): 警告: the `gets' function is dangerous and should n
ot be used.
shianlou:C-C-/ (master*) $ ls [15:17:04]
test.cpp  testword  testword.cpp
```

注意是使用 g++ 而不是 gcc 编译，会出现 warning，不用管，gets() 函数在输入时没有限定字符串的长度，而 linux 是很严谨的，所以这里给出一 warning。这个时候输入 ls，发现目录下已经出现编译成功的 testword 可运行程序，然后运行，成功运行结果如图：

```
shianlou:C-C-/ (master*) $ ls
test.cpp  testword  testword.cpp
shianlou:C-C-/ (master*) $ sudo ./testword
shianlou:C-C-/ (master*) $ ls
result.txt  test.cpp  testword  testword.cpp
```

有个 result.txt 的文件，打开它，内容如下：



```
1 #
2 int---->7
3 main---->1
4 (----->42
5 )----->43
6 for---->22
7 (----->42
8 int---->7
9 i---->25
10 =----->38
11 0----->26
12 ;----->41
13 i---->25
14 <----->33
15 10----->26
16 ;----->41
17 i---->25
18 +----->27
19 +----->27
```

NORMAL master result.txt unix utf-8 text 3% 1:1
"result.txt" [只读] 29L, 318C

本课程到此结束，谢谢学习，如有问题请留言，我会定期回复。

五、作业思考

实验楼环境中暂时无法输入中文字符，但是在实际生活应用中，中文是很常见的。考虑一下，如果遇到中文字符，词法分析器该怎么解决。

动手做实验，轻松学IT。

实验楼-通过动手实践的方式学会IT技术。

公司简介 (/aboutus) 联系我们 (/contact) 常见问题 (/faq#howtostart) 我要开课 (/labs) 隐私协议 (/privacy)

会员条款 (/terms) 友情链接 (/friends)

站长统计 (http://www.cnzz.com/stat/website.php?web_id=5902315)

蜀ICP备13019762号 (<http://www.miibeian.gov.cn/>)



QQ群



微信



微博

(<http://weibo.com/shiyanlou2013>)