

**院 系：计 算 机 学 院**

**实验课程：编译原理**

**实验项目：XLEX生成器**

**指导老师：黄煜廉**

**专 业：计算机科学与技术**

**班 级：2018级 3 班**

**学 生：**

**学 号：**

**华南师范大学教务处**

**C++源代码单词扫描程序（词法分析）**

1. **中文摘要**

本次实验的目标是实现一个C++源代码扫描程序，利用程序对C++源代码中的标识符，关键字，数（包括整数、浮点数），字符串、注释、特殊符号（分界符）和运算符号进行识别。

本次实验实现的扫描程序主要完成了对C++中各种记号的识别并完成相应的打印功能。

1. **关键词**

C++源代码单词扫描程序 扫描功能

1. **前言**

本次实验通过编写Qt应用程序实现了交互界面，可供用户选择文件（仅限.txt文件）进行扫描，识别出C++源代码的各种记号，并将结果存储为.txt文件。

1. **软件开发过程**
2. **开发平台及工具**

操作系统：Windows10

开发平台：Qt Creator 4.9.1 (Enterprise)

1. **需求分析**
2. 基本功能需求

（1）C++源代码扫描程序识别C++记号。

C++语言包含了几种类型的记号：标识符，关键字，数（包括整数、浮数），字符串、注释、特殊符号（分界符）和运算符号等。

（2）打开一个C++源文件，打印出所有以上的记号。

1. 界面需求

Widows界面

1. **概要设计**
2. C++记号

（1）关键字（66个）：

asm,auto,bool,break,case,catch,char,class,const,const\_cast,continue,default,delete,do,double,dynamic\_cast,else,enum,explicit,export,extern,false,float,for,friend,goto,if,inline,int,long,mutable,namespace,new,operator,private,protected,public,register,reinterpret\_cast,return,short,signed,sizeof,static,static\_cast,struct,switch,template,this,throw,true,try,typedef,typeid,typename,union,usigned,using,virtual,void,volatile,wchar\_t,while,stdio.h,iostream,fstream

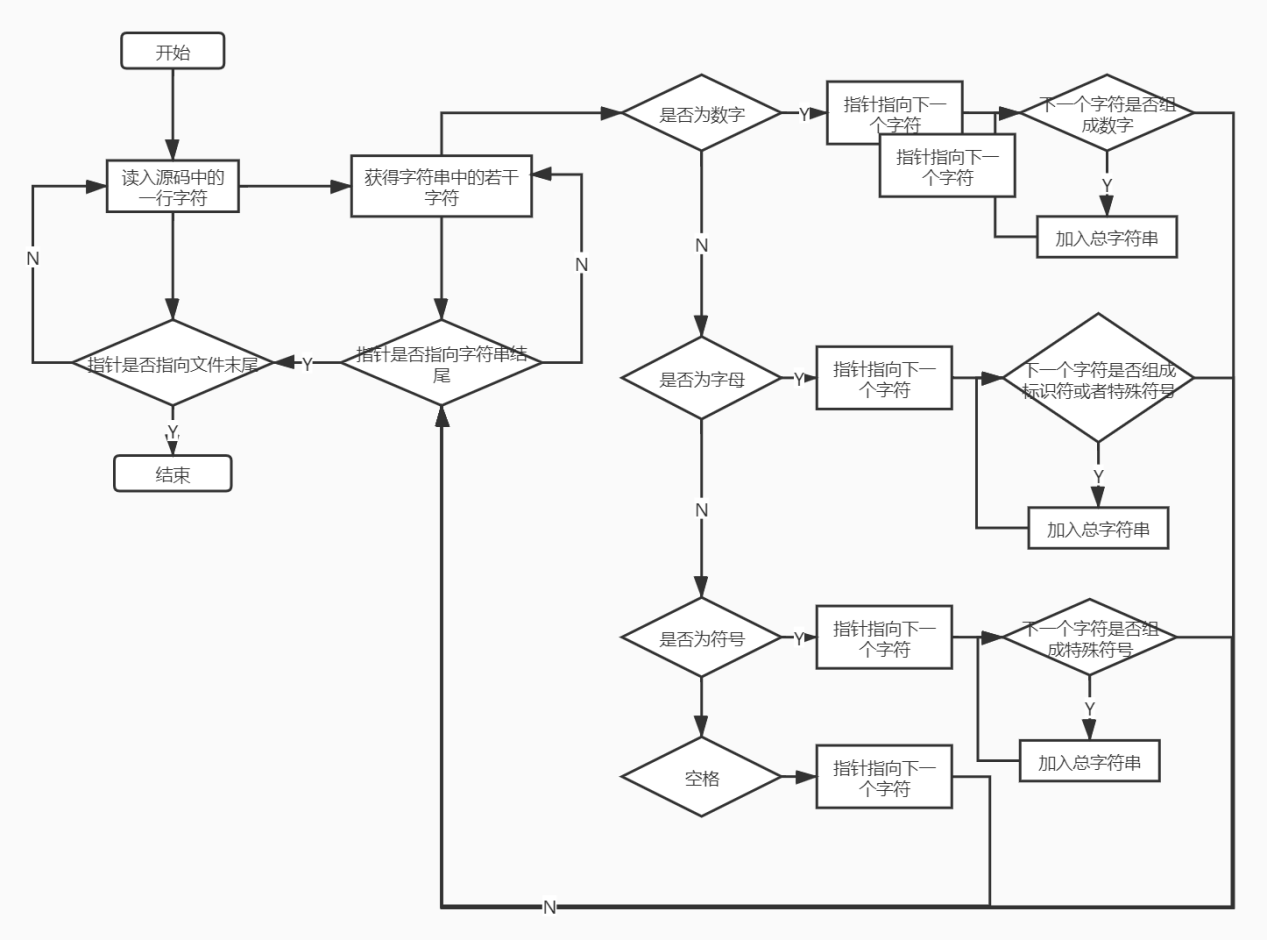
（2）一元运算符或分隔符（25个）：

, ; ( ) { } # ^ ? : . [ ] + - \* / % = > < ! ~ | &

（3）二元运算符或分隔符（18个）：

&& || == >= <= != ++ -- << >> += -= \*= /= %= &= ^= ->

1. **重要功能算法流程图**
2. C++记号扫描分析功能



1. **核心代码分析**
2. C++记号扫描分析功能代码

//开始分词

//startbutton

void MainWindow::on\_startButton\_clicked()

{

QString QSkey,QSfile;

string skey,sfile;

string Program;

KeyWord \*key;

unsigned int now,next;

bool out=true;

//准备开文件

QSkey=ui->keyEdit->text(); QSfile=ui->fileEdit->text();

skey=QSkey.toStdString(); sfile=QSfile.toStdString();

//清空Final.txt文件

ofstream write("Final.txt");

write.close();

//打开文件

key=Build(skey);

if(key==NULL) return ;

ifstream in;

in.open(sfile,ios\_base::in);

if(!in){//打开失败时用

return ;

}

while( getline(in, Program) ){

for(now=0; now<Program.size(); ){

//首为字母

if( (Program[now]>='A' && Program[now]<='Z' ) || (Program[now]>='a' && Program[now]<='z' ) ){

next = wJudge(Program, now, key); now = next;

}

//首为数字

else if( Program[now]>='0' && Program[now]<='9' ){

next = nJudge(Program, now); now = next;

}

//首为符号

else{

if(Program[now]==' ' || Program[now] == ' '){

now++; continue;

}

next = cJudge(Program, now, out); now = next;

if(now == Program.size()+1) out=false;

else out=true;

}

}

}

in.close();//关闭文件

QMessageBox::information(this,"信息","分词结果已写入Final.txt文件");

}

//建造二叉树

KeyWord \*Build(string name){

KeyWord \*now, \*temp, \*root;

ifstream Key\_in;

string keyword;

//根结点设置

root = new KeyWord[1];

root->lc = NULL; root->rc = NULL; root->Key = "Root";

Key\_in.open(name, ios::in);

if( !Key\_in ){

return NULL;

}

while( getline(Key\_in, keyword) ){

temp = root;

now = new KeyWord[1];

now->lc=NULL; now->rc=NULL; now->Key=keyword;

while( temp!=NULL ){

if( keyword < temp->Key ){

if( temp->lc == NULL){

temp->lc=now; break;

}

else temp=temp->lc;

}

else{

if( temp->rc == NULL ){

temp->rc=now; break;

}

else temp=temp->rc;

}

}

}

Key\_in.close();

return root;

}

//判断是否为关键词

bool ifKey(string Word, KeyWord \*key){

KeyWord \*now = key;

while( now!=NULL ){

if( Word < now->Key ){

now=now->lc;

}

else if( Word > now->Key){

now=now->rc;

}

else return true;

}

return false;

}

//首为字母分词

unsigned int wJudge(string Program, unsigned int now, KeyWord\* Key){

string KeyWord, Total = "";

unsigned int next;

ofstream wirte("Final.txt",ios::app);

Total += Program[now];

for(next=now+1; next<Program.size(); next++){

if (Program[next]=='\_' || Program[next]=='.' || (Program[next]>='A' && Program[next]<='Z') || (Program[next]>='a' && Program[next]<='z') || (Program[next]>='0' && Program[next]<='9') )

Total += Program[next];

else break;

}

//检测是否为关键字,运算符，函数

if( ifKey(Total,Key) ){

wirte<<Total<<" 关键字"<<endl;

return next;

}

if( Total=="sizeof" || Total=="Cast" ){

wirte<<Total<<" 运算符"<<endl;

return next;

}

for(unsigned int i=0; i<Total.size(); i++){

if(Total[i]=='.'){

wirte<<Total<<" 函数"<<endl;

return next;

}

}

wirte<<Total<<" 标识符"<<endl;

wirte.close();

return next;

}

//首为数字分词

unsigned int nJudge(string Program, unsigned int now){

string Total="";

unsigned int next;

ofstream wirte("Final.txt",ios::app);

Total += Program[now];

for(next=now+1; next<Program.size(); next++){

if (Program[next]=='.' || (Program[next]>='0' && Program[next]<='9' ) || Program[next]=='E' ||Program[next]=='e' )

Total += Program[next];

else if( (Program[next-1]=='e' || Program[next-1]=='E') && Program[next]=='-'){

Total += Program[next];

}

else break;

}

wirte<<Total<<" 数值"<<endl;

wirte.close();

return next;

}

//首为符号分词

unsigned int cJudge(string Program, unsigned int now, bool out){

string Total="";

unsigned int next;

Total += Program[now];

ofstream wirte("Final.txt",ios::app);

if(!out){ //在多行注释中

for(next=now; next<Program.size()-1; next++ ){

if(Program[next]=='\*' && Program[next+1]=='/'){//注释结束

wirte<<Program[next]<<Program[next+1]<<" 特殊符号"<<endl;

return (next+2);

}

else return ( Program.size()+1 );

}

}

//‘/’情况

if( Program[now]=='/' ){

next = now+1;

if( Program[next]=='/'){ //单行注释

Total += Program[next]; next = Program.size();

}

else if( Program[next]=='\*' ){ //多行注释

Total += Program[next]; next = Program.size()+1;

}

else if( Program[next] == '='){

Total += Program[next]; next++;

}

wirte<<Total<<" 特殊符号"<<endl; return next;

}

//为字符串或者字符类型的情况

if( Program[now] == '\"'){

for( next = now+1; next < Program.size(); next++ ){

if( Program[next] != '\"'){

Total += Program[next];

}

else{

Total += Program[next];

break;

}

}

wirte<<Total<<" 为字符串"<<endl;

return next+1;

}

if( Program[now] == '\'' ){

for( next = now+1; next<Program.size(); next++){

if( Program[next] != '\''){

Total += Program[next];

}

else{

Total += Program[next];

break;

}

}

wirte<<Total<<" 字符"<<endl;

return next+1;

}

//为数值情况

if( Program[now] == '-' ){

for( next=now+1; next<Program.size(); next++){

if( Program[next]>='0' && Program[next]<='9' )

Total += Program[next];

else break;

}

if( Total.size() > 1 ){

wirte<<Total<<" 数值"<<endl;

return next;

}

}

next = now+1;

switch( Program[now] ){

case '-':{

if( Program[next] == '-' || Program[next] == '>' || Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '!':{

if( Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '\*':{

if( Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '%':{

if( Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '^':{

if( Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '=':{

if( Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '+':{

if( Program[next] == '=' || Program[next] == '+' ){

Total += Program[next]; next++;

}

break;

}

case '&':{

if( Program[next] == '=' || Program[next] == '&' ){

Total += Program[next]; next++;

}

break;

}

case '|':{

if( Program[next] == '=' || Program[next] == '|' ){

Total += Program[next]; next++;

}

break;

}

case '<':{

if( Program[next] == '<' ){

Total += Program[next]; next++;

if( Program[next] =='=' ){

Total += Program[next]; next++;

}

}

break;

}

case '>':{

if( Program[next] == '>' ){

Total += Program[next]; next++;

if( Program[next] =='=' ){

Total += Program[next]; next++;

}

}

break;

}

}

wirte<<Total<<" 特殊符号"<<endl;

wirte.close();

return next;

}

C++源文件压缩功能代码

//////////////////////////////////////////////////////////////

////函数名：compress

////函数功能：对一行字符进行压缩处理，去掉空格以及屏蔽段

void compress(string s)

{

s.resize(300);

int i = 0;

char front = NULL;

if(note\_flag)

{

while(s[i]!='\*'&&s[i]!='\0'||s[i]=='\*'&&s[i+1]!='/')

i++;

if(s[i]!='\0')

{

i= i+2;

note\_flag = false;

}

}

while(s[i]!='\0')

{

if(s[i]==' '||s[i]=='\t')

{

i++;

continue;

}

else if(s[i]=='/'&&s[i+1]=='\*')

{

note\_flag = true;

i = i+2;

while(s[i]!='\*'&&s[i]!='\0'||s[i]=='\*'&&s[i+1]!='/')

i++;

if(s[i]!='\0')

note\_flag = false;

}

else if(s[i]=='\''&&s[i+1]==' '&&s[i+2]=='\'')

{

outfile<<s[i]<<" "<<s[i+2];

i=i+3;

}

else if(s[i]=='"'&&s[i-1]!='\''&&s[i+1]!='\'')

{

outfile<<s[i];

i++;

while(s[i]!='"'||s[i-1]=='\\')

{

outfile<<s[i];

i++;

}

outfile<<s[i];

i++;

}

else if(s[i]!=' '&&s[i]!='\t'&&s[i]!='\0')

{

if(i>1&&(s[i-1]==' '||s[i-1]=='\t'))

{

if((isalpha(front)||isdigit(front))&&(isalpha(s[i])||isdigit(s[i])))

outfile<<" ";

}

if(s[i]=='\*'&&s[i+1]=='/'||(s[i]=='/'&&s[i-1]=='\*'))break;

else outfile<<s[i];

if(isdigit(s[i])&&s[i+1]=='\0')

outfile<<" ";

if(s[i+1]==' '||s[i+1]=='\t')

front = s[i];

i++;

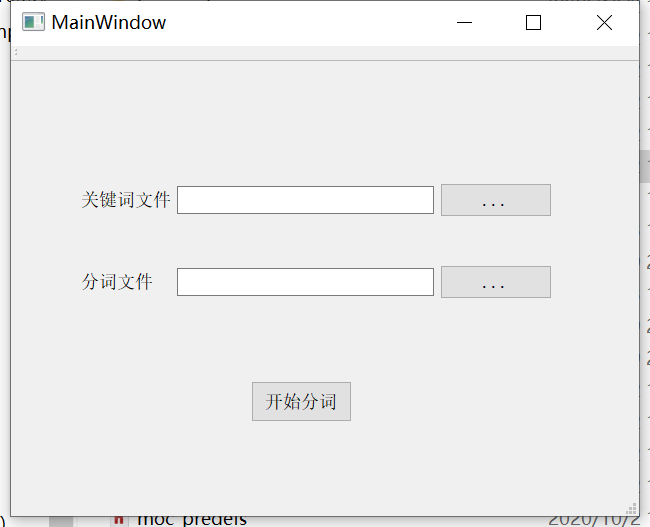
}

}

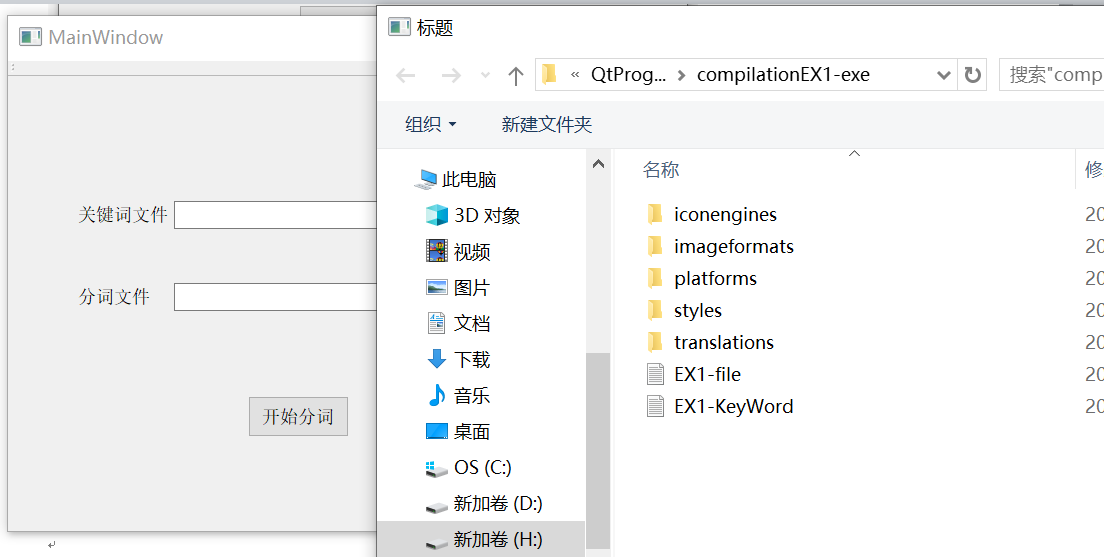
}

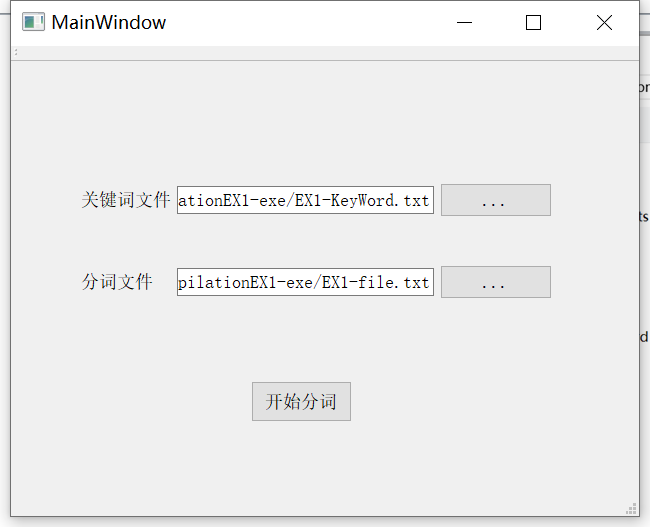
1. **测试情况**
2. 启动应用程序

系统界面



1. 选择程序文件和关键词文件

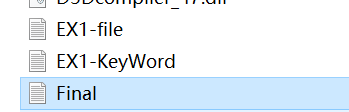




1. 扫描源文件得到结果文件



1. 得到结果Final.txt.文件





1. **设计中遇到的问题及解决方法**
2. 按照原思路，多行注释的判断存在问题，于是后来调整思路添加了一个辅助判断变量out。
3. 在QT中使用size()函数时会出现警告信息，经过阅读信息提示将与其相关的比较均定义为unsigned int。
4. 因为一开始是在DevC上进行编写，所以文件打开函数需要char\*类型变量。当后面为了实现图形化而搬运至QT时，因为QT从输入框中读入的为QString类型而不能直接使用，经过搜索可以了解到可以使用toStdString()来对其进行转换。
5. **附录**

**.h头文件**

mainwindow.h

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

namespace Ui {

class MainWindow;

}

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

explicit MainWindow(QWidget \*parent = nullptr);

~MainWindow();

private slots:

void on\_keyButton\_clicked();

void on\_fileButton\_clicked();

void on\_startButton\_clicked();

private:

Ui::MainWindow \*ui;

};

**cpp源文件**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "QFileDialog"

#include "fstream"

#include "cstring"

#include "stdio.h"

#include "QMessageBox"

using namespace std;

//树结点

class KeyWord{

public:

KeyWord \*lc, \*rc;

string Key;

};

KeyWord \*Build(string filename);

bool ifKey(string Word, KeyWord \*key);

unsigned int wJudge(string Program, unsigned int now, KeyWord\* Key);

unsigned int nJudge(string Program, unsigned int now);

unsigned int cJudge(string Program, unsigned int now, bool out);

MainWindow::MainWindow(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~MainWindow()

{

delete ui;

}

//选择文件

//keybutton

void MainWindow::on\_keyButton\_clicked()

{

QString fileName = QFileDialog::getOpenFileName(NULL,"标题",".","\*.txt");

ui->keyEdit->setText(fileName);

}

//filebutton

void MainWindow::on\_fileButton\_clicked()

{

QString fileName = QFileDialog::getOpenFileName(NULL,"标题",".","\*.txt");

ui->fileEdit->setText(fileName);

}

//开始分词

//startbutton

void MainWindow::on\_startButton\_clicked()

{

QString QSkey,QSfile;

string skey,sfile;

string Program;

KeyWord \*key;

unsigned int now,next;

bool out=true;

//准备开文件

QSkey=ui->keyEdit->text(); QSfile=ui->fileEdit->text();

skey=QSkey.toStdString(); sfile=QSfile.toStdString();

//清空Final.txt文件

ofstream write("Final.txt");

write.close();

//打开文件

key=Build(skey);

if(key==NULL) return ;

ifstream in;

in.open(sfile,ios\_base::in);

if(!in){//打开失败时用

return ;

}

while( getline(in, Program) ){

for(now=0; now<Program.size(); ){

//首为字母

if( (Program[now]>='A' && Program[now]<='Z' ) || (Program[now]>='a' && Program[now]<='z' ) ){

next = wJudge(Program, now, key); now = next;

}

//首为数字

else if( Program[now]>='0' && Program[now]<='9' ){

next = nJudge(Program, now); now = next;

}

//首为符号

else{

if(Program[now]==' ' || Program[now] == ' '){

now++; continue;

}

next = cJudge(Program, now, out); now = next;

if(now == Program.size()+1) out=false;

else out=true;

}

}

}

in.close();//关闭文件

QMessageBox::information(this,"信息","分词结果已写入Final.txt文件");

}

//建造二叉树

KeyWord \*Build(string name){

KeyWord \*now, \*temp, \*root;

ifstream Key\_in;

string keyword;

//根结点设置

root = new KeyWord[1];

root->lc = NULL; root->rc = NULL; root->Key = "Root";

Key\_in.open(name, ios::in);

if( !Key\_in ){

return NULL;

}

while( getline(Key\_in, keyword) ){

temp = root;

now = new KeyWord[1];

now->lc=NULL; now->rc=NULL; now->Key=keyword;

while( temp!=NULL ){

if( keyword < temp->Key ){

if( temp->lc == NULL){

temp->lc=now; break;

}

else temp=temp->lc;

}

else{

if( temp->rc == NULL ){

temp->rc=now; break;

}

else temp=temp->rc;

}

}

}

Key\_in.close();

return root;

}

//判断是否为关键词

bool ifKey(string Word, KeyWord \*key){

KeyWord \*now = key;

while( now!=NULL ){

if( Word < now->Key ){

now=now->lc;

}

else if( Word > now->Key){

now=now->rc;

}

else return true;

}

return false;

}

//首为字母分词

unsigned int wJudge(string Program, unsigned int now, KeyWord\* Key){

string KeyWord, Total = "";

unsigned int next;

ofstream wirte("Final.txt",ios::app);

Total += Program[now];

for(next=now+1; next<Program.size(); next++){

if (Program[next]=='\_' || Program[next]=='.' || (Program[next]>='A' && Program[next]<='Z') || (Program[next]>='a' && Program[next]<='z') || (Program[next]>='0' && Program[next]<='9') )

Total += Program[next];

else break;

}

//检测是否为关键字,运算符，函数

if( ifKey(Total,Key) ){

wirte<<Total<<" 关键字"<<endl;

return next;

}

if( Total=="sizeof" || Total=="Cast" ){

wirte<<Total<<" 运算符"<<endl;

return next;

}

for(unsigned int i=0; i<Total.size(); i++){

if(Total[i]=='.'){

wirte<<Total<<" 函数"<<endl;

return next;

}

}

wirte<<Total<<" 标识符"<<endl;

wirte.close();

return next;

}

//首为数字分词

unsigned int nJudge(string Program, unsigned int now){

string Total="";

unsigned int next;

ofstream wirte("Final.txt",ios::app);

Total += Program[now];

for(next=now+1; next<Program.size(); next++){

if (Program[next]=='.' || (Program[next]>='0' && Program[next]<='9' ) || Program[next]=='E' ||Program[next]=='e' )

Total += Program[next];

else if( (Program[next-1]=='e' || Program[next-1]=='E') && Program[next]=='-'){

Total += Program[next];

}

else break;

}

wirte<<Total<<" 数值"<<endl;

wirte.close();

return next;

}

//首为符号分词

unsigned int cJudge(string Program, unsigned int now, bool out){

string Total="";

unsigned int next;

Total += Program[now];

ofstream wirte("Final.txt",ios::app);

if(!out){ //在多行注释中

for(next=now; next<Program.size()-1; next++ ){

if(Program[next]=='\*' && Program[next+1]=='/'){//注释结束

wirte<<Program[next]<<Program[next+1]<<" 特殊符号"<<endl;

return (next+2);

}

else return ( Program.size()+1 );

}

}

//‘/’情况

if( Program[now]=='/' ){

next = now+1;

if( Program[next]=='/'){ //单行注释

Total += Program[next]; next = Program.size();

}

else if( Program[next]=='\*' ){ //多行注释

Total += Program[next]; next = Program.size()+1;

}

else if( Program[next] == '='){

Total += Program[next]; next++;

}

wirte<<Total<<" 特殊符号"<<endl; return next;

}

//为字符串或者字符类型的情况

if( Program[now] == '\"'){

for( next = now+1; next < Program.size(); next++ ){

if( Program[next] != '\"'){

Total += Program[next];

}

else{

Total += Program[next];

break;

}

}

wirte<<Total<<" 为字符串"<<endl;

return next+1;

}

if( Program[now] == '\'' ){

for( next = now+1; next<Program.size(); next++){

if( Program[next] != '\''){

Total += Program[next];

}

else{

Total += Program[next];

break;

}

}

wirte<<Total<<" 字符"<<endl;

return next+1;

}

//为数值情况

if( Program[now] == '-' ){

for( next=now+1; next<Program.size(); next++){

if( Program[next]>='0' && Program[next]<='9' )

Total += Program[next];

else break;

}

if( Total.size() > 1 ){

wirte<<Total<<" 数值"<<endl;

return next;

}

}

next = now+1;

switch( Program[now] ){

case '-':{

if( Program[next] == '-' || Program[next] == '>' || Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '!':{

if( Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '\*':{

if( Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '%':{

if( Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '^':{

if( Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '=':{

if( Program[next] == '='){

Total += Program[next]; next++;

}

break;

}

case '+':{

if( Program[next] == '=' || Program[next] == '+' ){

Total += Program[next]; next++;

}

break;

}

case '&':{

if( Program[next] == '=' || Program[next] == '&' ){

Total += Program[next]; next++;

}

break;

}

case '|':{

if( Program[next] == '=' || Program[next] == '|' ){

Total += Program[next]; next++;

}

break;

}

case '<':{

if( Program[next] == '<' ){

Total += Program[next]; next++;

if( Program[next] =='=' ){

Total += Program[next]; next++;

}

}

break;

}

case '>':{

if( Program[next] == '>' ){

Total += Program[next]; next++;

if( Program[next] =='=' ){

Total += Program[next]; next++;

}

}

break;

}

}

wirte<<Total<<" 特殊符号"<<endl;

wirte.close();

return next;

}