



杭州电子科技大学
《编译原理课程实践》
实验报告

题 目：实验 3 词法分析
学 院：计算机学院_____

专 业：计算机科学与技术_____

班 级：21052314_____

学 号：21051408_____

姓 名：蔡龙祥_____

完成日期：2023-11-9_____

一、 实验目的

在熟悉 SysY 文法的基础上,编写一个程序,对使用SysY 语言书写的源代码进行词法分析

程序要能够查出 SysY 源代码中可能包含的词法错误

二、 实验内容与实验要求

词法错误: 出现 SysY 词法中未定义的字符以及任何不符合 SysY 词法单元定义的字符。

定义错误: 出现定义数组时中括号中无参数的情况

进制错误: 出现不符合进制数规则的错误

程序在输出错误提示信息时,需要输出具体的错误类型、出错的位置(源程序的行号)以及相关的说明文字。

三、 设计方案与算法描述

使用 flex 和 bison 生成词法分析和语法分析器,在 flex 中定义好正则匹配的规则,用来捕获源代码文件中的符合 sysy 文法的词根,并根据不同的匹配执行不同的代码段,例如对于字符串或者数字类型的词,先统一转化为字符串,对 8 进制和 16 进制的数字需要分开处理,既要区分报错信息,也要计算出对应的 10 进制值并打印结果,对于注释也做了特殊处理,比如 /* 和 //,当捕获到注释标记符时会进入新的词法分析的命名空间,以此跳过注释中的文字

如果不使用 bison,匹配到的结果返回的 token 属性值需要自己定义,但由于使用了 bison,所以这部分逻辑需要另行补充,所以虽然只是词法分析的工作,但是也顺便把语法分析的任务完成了,过程中对报错信息的检测也插入到了词法分析的文件中,对于错误类型的判断也是根据语法分析的类型得出,错误的行号和文本信息则是借助 yylineno 和 yytext 获取的,为了保证检测到错误时不直接退出程序还需要做错误恢复操作

四、 测试结果

测试样例

```
int main(){
int a=10;
int b=0x12;
int c=017;
int d=0x1z;
int e=08;
~
return 0;
int a[]={1};
```

```
}
```

测试代码:

```
root@2842d3b9be58:~/experiment/exp3# ./parsecpp ./main.c
open file ./main.c successfully
int:<INT,291>
main:<Ident,290>
(:<L,278>
):<R,279>
{:<OB,274>
int:<INT,291>
a:<Ident,290>
=:<ASSIGN,258>
10:<INTNUM,298>
GET INTNUM: 10
get number value : 10
;:<SEMI,271>
int:<INT,291>
b:<Ident,290>
=:<ASSIGN,258>
0x12:<ILLEGAL_OCTAL_CONST,295>
Error type [Number] at line [3] : Illegal octal number 0x12
;:<SEMI,271>
int:<INT,291>
c:<Ident,290>
=:<ASSIGN,258>
017:<INTNUM,298>
GET INTNUM: 017
get number value : 15
;:<SEMI,271>
int:<INT,291>
d:<Ident,290>
=:<ASSIGN,258>
0x1z:<ILLEGAL_HEX_CONST,294>
Error type [Number] at line [5] : Illegal hex number 0x1z
;:<SEMI,271>
int:<INT,291>
e:<Ident,290>
=:<ASSIGN,258>
08:<ILLEGAL_OCTAL_CONST,295>
Error type [Number] at line [6] : Illegal octal number 08
;:<SEMI,271>
Error type [character] at line [7] :Invalid character ~
return:<RETURN,286>
0:<INTNUM,298>
GET INTNUM: 0
get number value : 0
;:<SEMI,271>
int:<INT,291>
a:<Ident,290>
[:<LB,276>
]:<RB,277>
Error type [ArrayDefine] at line [9] : invalid Arraydef: need arguments in []
=:<ASSIGN,258>
{:<OB,274>
1:<INTNUM,298>
GET INTNUM: 1
get number value : 1
}:<CB,275>
;:<SEMI,271>
}:<CB,275>
Funcdef successful!
```

对于错误的八进制和 16 进制分别打印错误类型, 行号以及错误输入

对于正确的则返回对应的 10 进制结果

按照 SySy 文法定义数组时 [] 内必须有参数

将匹配到的每一个 token 的类型和属性值打印

对 SySy 文法未定义的字符打印错误提示

五、 源代码

Sysy.l

%{

引入头文件,定义宏

%}

定义正则表达式别名

声明命名空间

%s COMMENT

%s LINECOMMENT

%%

匹配注释

"/*" {BEGIN(COMMENT);}

"//" {BEGIN(LINECOMMENT);}

//字段匹配

"<" RETURN_TOKEN(LT);

.

printf("Error type [character] at line

[%d] :Invalid character %c \n",yylineno,yytext[0]);

%%

Sysy.y

%{

C 语言定义代码

%}

%union{

//类型声明

}

%token

...

%type

...

%start CompUnit

CompUnit: {}

;

%%

模式匹配模块代码略去

%%

//可以读取文件,也可以手动输入代码

//main 入口函数略去