lab1实验报告

3210105488 刘扬

编译及测试

```
cd path_to_sp24-starter
make
python3 test.py ./compiler lab1
```

实现功能

- flex实现词法分析
- bison实现语法分析

关键代码

flex

- 十进制/十六进制/八进制的处理

```
0[xX][0-9a-fA-F]+  { printf("INTCONST\n"); return INTCONST; }  // 匹配
十六进制整数
0[0-7]+            { printf("INTCONST\n"); return INTCONST; }  // 匹配
八进制整数
[0-9]+             { printf("INTCONST\n"); return INTCONST; }  // 匹配
十进制整数
```

- 注释的处理

  定义了old_state用来存储YY_START状态，重新定义了一个新状态COMMET来处理多行注释

```
int old_status;

%x COMMENT
```

```
"//".*             ;
"/*"               { old_status = YY_START; BEGIN COMMENT; }
<COMMENT>"*/"      { BEGIN old_status; }
```

- 其他词法分析

正常枚举其他词法即可，比较简单

bison

- 运算符优先级定义

  给了ELSE最高优先级，这样子在后续IF+ELSE和IF的匹配中，会先匹配IF+ELSE

  ```
  %left AND_OP OR_OP
  %left EQ_OP NE_OP
  %left LT_OP GT_OP LE_OP GE_OP
  %left ADD_OP SUB_OP
  %left MUL_OP DIV_OP MOD_OP
  %left ELSE
  ```

  ```
  Stmt: ...
      | IF LPAREN Cond RPAREN Stmt %prec ADD_OP  // AND_OP是最低优先级
      | IF LPAREN Cond RPAREN Stmt ELSE Stmt
      ...
  ```

- 处理附录中不合理的部分

  附录中给出的int有多种解释方法：

  ```
  BType         ::= "int";
  FuncDef       ::= FuncType IDENT "(" [FuncFParams] ")" Block;
  FuncType      ::= "void" | "int";
  ```

  在bison中修改如下：

  ```
  BType: INT_TYPE;
  FuncDef: BType IDENT LPAREN RPAREN Block
      | BType IDENT LPAREN FuncFParams RPAREN Block
      | VOID_TYPE IDENT LPAREN RPAREN Block
      | VOID_TYPE IDENT LPAREN FuncFParams RPAREN Block
      ;
  ```

- 其他语法分析

  按照附录完成bison即可

  将附录中的[]内的内容翻译成0次或1次，即枚举该情况

  将附录中的{}内的内容展开为右递归