

暨南大学本科实验报告专用纸

课程名称 编译原理 成绩评定
实验项目名称 Tiny C 语言编译程序实验一 Scanner 指导教师 余芳
实验项目编号 02 实验地点 N116
学生姓名 温钊迪 学号 2016051487
学院 信息科学技术学院 系 计算机科学系 专业 计算机科学与技术
实验时间 2019 年 04 月 26 日

一、实验目的

填写 getToken() 函数，完成词法分析器 scan.c。实验任务

二、实验要求

- 1) 仅允许整数类型，不允许实数类型
- 2) 标识符由大小写英文字母组成，最多 52 个。其识别按最长匹配原则
- 3) 整数后紧跟非数字，或标识符后紧跟非字母认为是一个新 Token 开始
- 4) 由 {} 括起来符号串都认为是注释部分，该部分在词法分析时被过滤掉
- 5) 识别出的 Token 由两个变量：currentToken, tokenString 识别，其中 currentToken 代表 Token 的类属，为一个名为 TokenType 的枚举类型，在文件 globals.h 中定义；tokenString 代表 Token 在程序中出现的形式，即其本来面目。例如整数 10 的 currentToken 值为 NUM，而 tokenString 值为 '10'；标识符 i 的 currentToken 值为 ID，而 tokenString 值为 'i'

三、源代码

(1) scan.c 文件

```
TokenType getToken(void)
{
    /* index for storing into tokenString */
    int tokenStringIndex = 0;

    /* holds current token to be returned */
    TokenType currentToken;

    /* current state - always begins at START */
    StateType state = START;
```

```

/* flag to indicate save to tokenString */
int save;

while (state != DONE)
{ int c = getNextChar();

  save = TRUE;

  switch (state)
  { case START:

    if (isdigit(c)){

      state = INNUM;

      break;

    }else if(c==''){ //转去注释

      state = INCOMMENT;

      save=FALSE;

      break;

    }else if(c==' ' || c=='\t' || c=='\n' ){ //state 不变

      save = FALSE;

      state = START;

      break;

    }else if(c==':'){ //转赋值

      state = INASSIGN;

      break;

    }else if(c>='a' && c<='z' || c>='A' && c<='Z'){ //转字符

      state=INID;

      break;

    }else if(c=='='){

      state = DONE;

      currentToken = EQ;

      break;

    }else if(c=='<'){

      state = DONE;

```

```
        currentToken = LT;

        break;
    } else if(c=='+'){

        state = DONE;

        currentToken = PLUS;

        break;
    } else if(c=='-'){

        state = DONE;

        currentToken = MINUS;

        break;
    } else if(c=='*'){

        state = DONE;

        currentToken = TIMES;

        break;
    } else if(c=='/'){

        state = DONE;

        currentToken = OVER;

        break;
    } else if(c=='('){

        state = DONE;

        currentToken = LPAREN;

        break;
    } else if(c==')'){

        state = DONE;

        currentToken = RPAREN;

        break;
    } else if(c==';'){

        state = DONE;

        currentToken = SEMI;

        break;
```

```

    }else if(c==EOF){

        state = DONE;

        currentToken = ENDFILE;

        break;

    }else{

        state = DONE;

        currentToken = ERROR;

        break;

    }          /* 此处请填写完整 //此处请自己填写(字符、:、空格/tab/换行、
{、算符及界符等) */

    break;

case INCOMMENT: //注释

    if(c==''){ //注释结束

        state= START;

        save=FALSE;

        //currentToken = INCOMMENT;

        break;

    }else{ //其他字符 state 不变

        state = INCOMMENT;

        save= FALSE;

        break;

    }          /* 此处请填写完整 此处请自己填写, 仅出现 ‘}’ 或 EOF (注释未
完结束程序) 时才改变状态。*/

    break;

case INASSIGN: //赋值

    if(c=='='){

        state= DONE;

        currentToken=ASSIGN;

        break;

    }else{

```

```

        state = DONE;

        currentToken = ERROR;

        break;

        /* 此处请填写完整 此处请自己填写，‘=’ 或其它（出现错误）*/
    }

    break;

case INNUM: //数字

    if (!isdigit(c)) { /* backup in the input */

        ungetNextChar();

        save = FALSE;

        state = DONE;

        currentToken = NUM;

        break;

    } else {

        state = INNUM;

        break;

    }

    break;

case INID: //字符

    if (c >= 'a' && c <= 'z' || c >= 'A' && c <= 'Z') {

        state = INID;

    } else { //不是字符 结束 回退一个

        ungetNextChar();

        save = FALSE;

        state = DONE;

        currentToken = ID;

        break;

    }

    /* 此处请填写完整 /此处请自己填写，不是字符则回吐，并
    进入 DONE，且识别出一个 ID*/

```

```

        break;

    case DONE:
        default: /* should never happen */

            fprintf(listing,"Scanner Bug: state= %d\n",state);

            state = DONE;

            currentToken = ERROR;

            break;

    }

    if ((save) && (tokenStringIndex <= MAXTOKENLEN))

        tokenString[tokenStringIndex++] = (char) c;

    if (state == DONE)

    { tokenString[tokenStringIndex] = '\0';

        if (currentToken == ID)

            currentToken = reservedLookup(tokenString);

        }

    }

    if (TraceScan) {

        fprintf(listing,"\t%d: ",lineno);

        printToken(currentToken,tokenString);

    }

    return currentToken;

} /* end getToken */

```

(2)

(2)在安装完 tcc 后

①在 windows 的 shell 下运行 tcc main.c scan.c util.c

```
PS C:\Users\lenovo\Desktop\untitled1> tcc main.c scan.c util.c
```

运行得到 main.exe

② 然 后 运 行 .\main.exe .\SAMPLE.TNY

