

## 实验部分

### 作业题：一个简单的编译器 25 分

编译器演示程序，可将 C 语言测试程序编译为目标代码——汇编程序，用汇编器转换为二进制程序后运行无误。

#### 1. 基本要求：

数据类型：int

语句：注释，声明，赋值，循环（while 和 for），判断（if），输入输出

算术运算：+，-，\*，/，%

关系运算：==，>，<，>=，<=，!=

逻辑运算：&&（与），||（或），！（非）

#### 2. 完成功能：

词法分析	语法分析	中间代码	类型检查	代码优化	错误分析	汇编程序
6	8	6	1	1	1	2

#### 3. 提交内容：

编译器源程序文本，包括 Lex、Yacc 等程序和 C/C++ 程序。

实验报告

#### 加分项：

1. 支持过程或函数调用 +2
2. 支持数组运算 一维数组 +1 多维数组 +2
3. 支持指针运算 一维指针 +1 多重指针 +2
4. 支持结构体 +1

...

## 上机检查截止时间：

### 11.10 词法分析

### 12.01 语法分析

### 12.22 汇编器（或中间代码）

## 词法分析

```
c:\users\zying\documents\lex yacc project\word_2
void main() {
VOID      void
MAIN      main
LP        (
RP        )
LBRACE    {
    int a = 0;
INT       int
ID        a      11212040
ASSIGN    =
NUMBER    0      0
SEMICOLON ;
    int i;
INT       int
ID        i      11212096
SEMICOLON ;
    if ( !a ) {
IF        if
LP        (
LNOT      !
ID        a      11212040
RP        )
LBRACE    {
    i = 1;
ID        i      11212096
ASSIGN    =
NUMBER    1      1
SEMICOLON ;
    }
RBRACE    }
}
RBRACE    }

/* fortest */
COMMENT   /* fortest */
// fortest //
```

## 语法分析

```
int main()
{
    int i=1;
    int a=i+3;
    while(a<=10)
    {
        i++;
    }
    return 0;
}
```

```
ID      main      12911792
ID      i         12916768
ID      a         12958728
start printing syntax tree:
-*Program (1)
  -*ExtDefList (1)
    -*ExtDef (1)
      -*Specifire (1)
        |-*TYPE :int
      -*VarDec (1)
        |-*ID :main
      -*Compst (3)
        -*DefList (3)
          -*Def (3)
            -*Specifire (3)
              |-*TYPE :int
            -*DecList (3)
              -*Dec (3)
                -*VarDec (3)
                  |-*ID :i
                -*ASSIGNOP (3)
                -*Exp (3)
                  |-*INTEGER :1
          -*DefList (4)
            -*Def (4)
              -*Specifire (4)
                |-*TYPE :int
              -*DecList (4)
                -*Dec (4)
                  -*VarDec (4)
                    |-*ID :a
                  -*ASSIGNOP (4)
                  -*Exp (4)
                    -*Exp (4)
                      |-*ID :i
                    -*PLUS (4)
                    -*Exp (4)
                      |-*INTEGER :3
        -*StmtList (5)
          -*Stmt (5)
            -*WHILE (5)
            -*Exp (5)
            -*Exp (5)
```

## 中间代码生成

```
(FUNCTION, main, {}, _)  
(=, #10, _, temp0)  
(=, #4, _, temp2)  
(*, temp2, temp0, temp1)  
(DEC, temp1, _, array0)  
(=, #2, _, temp3)  
(=, #4, _, temp6)  
(*, temp3, temp6, temp5)  
(+, &array0, temp5, temp4)  
(=, #5, _, temp7)  
(=, temp7, _, *temp4)  
(=, #2, _, temp8)  
(=, #4, _, temp11)  
(*, temp8, temp11, temp10)  
(+, &array0, temp10, temp9)  
(=, #3, _, temp12)  
(+, *temp9, temp12, temp13)  
(=, temp13, _, var0)  
(CALL, print, {var0}, _)  
(=, #0, _, temp14)  
(RETURN, _, _, temp14)
```