第四章

2018302070001沈思源

4.2

4.2　在汇编语言中下面的符号名哪些是正确的？哪些是不正确的？

ASMB,@ PROG,C1995,INPUT,3MON,x * y,JOW,OuT - BuF,IN/OUT,HIGHT,LOW,LOOP,MOV,IN,MOVE,AX,LEA,DISPLAY,OUTBUF,CR_LF,M_DWORD,PUSHM,INPUT,OUTPUT,DATA_AREA,[BX]

解

汇编语言中语句行名字项表示的规定如下:

字母：A-Z,a-z

数字：0-9

专用字符：**?**,**·**，**@**,**_**,**$**

除去数字外,所有的字符都可以放在源语句的第一个位置，且不可以使用保留字。如果用了·,则必须是第一个字符，且长度<=31个字符.

在宏汇编语言中所有变量名、标号名、记录名、指令助记符和寄存器名等统称符号.

正确
ASMB,@PROG,C1995,INPUT,JOW,HIGHT,LOW,LOOP,MOV,IN,MOVE,
AX,LEA,DISPLAY,OUTUF,CR_LF,M_DWORD,PUSHM,INPUT,OUTPUT,DATA_AREA
不正确
x*y, Out-Buf, IN/OUT
[BX]

4.3

4.3　下述每条语句在高级汇编程序汇编后,分别占有或留下多少个字节的存储空间?

```
STRING   DB      'Pentium.ASM'
COUNTS   =       $ - STRING
         DB      'AB','A','B'
         DW      '89','AB'
LEN      DW      80  DUP(80H)
COUNTL   EQU     LENGTH  LEN
```

```
            DB      10   DUP(5 DUP(?),' T ',6)
M_DWOR DD       ' AB '
            ORG     $ +10
            MOV     AL,9
```

解

首先，根据缩进和变量的定义，所有定义的内存单元都是在唯一的变量**STRING**指向的数据区中存储，除了**MOV**指令外，这里其余的都是伪指令，不产生机器码。

而指令的长度判断可参考:https://blog.csdn.net/Apelpoo/article/details/51870154

STRING DB 'Pentium.ASM'

采用DB定义，一个单位一字节(Byte)

因此占有 11个字节

COUNTS= $-STRING 该行变量COUNTS等价于EQU不占有内存空间

DB 'AB','A','B' 采用字节定义,占据4个字节 AB=4142H(ASCII码一个字符表示一个8位数字)

DW '89','AB' 采用字定义，一个字2字节可以存储两个字符,因此占据2字4字节

LEN DW 80 DUP(80H) 定义80个字，每个字数据项内容是80H，占据160字节

COUNTL EQU LENGTH LEN LENGTH用于计算元素个数，因此LENGTH LEN为80，COUNTL不占据内存

DB 10 DUP(5DUP(?),'T',6), 10*(5+1+1)=70,占据70个字节

M_DWOR DD 'AB' DD双字定义，占据4字节

ORG $+10 跳过10个字节

MOV AL,9 本指令涉及立即数，AL又是一字节，因此占据2字节

共占据16个字节

4.11

4.11 语句“MOV BX,M”和“LEA BX,M”有何区别？“LEA EBX,ARRAYD[EBX + ESI]”是否可以用“MOV EBX,OFFSET ARRAYD[EBX + ESI]”来代替？为什么？

解

MOV BX,M

表示将操作数M的数据值传送给BX寄存器，执行后(BX)=(M)

LEA BX,M

表示将M的EA传送给BX，EA是段内偏移地址或者叫做有效地址，只有16位

LEA EBX ARRAYD[EBX+ESI]表示取二维数组的对应对某行某列的一个数据项的EA传送给EBX

MOV EBX OFFSET ARRAYED[EBX+ESI]使用了OFFSET(取偏移地址EA的运算符)，**但是**，使用了复杂的地址表达式，不能够用OFFSET来取EA，因为汇编阶段不能够计算出寄存器内容的值形成EA，因此错误

4.13

4.13 阅读下列宏语句,并将引用宏指令的语句进行宏扩展。

```
SHIFT   MACRO   X,Y             SHIFT1  MACRO   X,Y,Z
        MOV     CL,X                    MOV     CL,X
        SAL     Y,CL                    S&Z     Y,CL
        ENDM                            ENDM
SHIFT   4,BX                    SHIFT1  8,SI,HR
```

```
MOV CL,4
SAL BX,CL

MOV CL,8
SHR SI,CL
```

4.14

4.14 阅读下面源程序段,将宏指令的引用进行宏展开。

```
.386
INOUTN  MACRO   X,Y,Z,W
        PUSH    AX
        X       Y,Z
        MOV     AH,W
        INT     21H
        POP     AX
        ENDM
PROMPT  DB      'Please input :','$'
N       EQU     20
ARRAY   DB      N+1,N+2 DUP(0)
        INOUTN  LEA,DX,PROMPT,9
        INOUTN  LEA,DX,ARRAY,10
        INOUTN  MOV,DL,'$',2
```

```
PUSH AX
LEA DX,PROMPT
MOV AH,9
INT 21H
POP AX

PUSH AX
LEA DX,ARRAY
MOV AH,10
INT 21H
POP AX

PUSH AX
```

```
MOV DL,'$'
MOV AH,2
INT 21H
POP AX
```

4.17

```
;--------------
;IO LIB
;--------------
INOUT MACRO X,Y
    LEA DX,X
    MOV AH,Y
    INT 21H
    ENDM

;-------------
;INPUT.ASM
;-------------
.586
INCLUDE IOLIB.LIB
EXTRN BUF:BYTE
PUBLIC GETINPUT
CODES SEGMENT USE16 PARA PUBLIC 'CODE'
      ASSUME  CS:CODES,FS:SEG BUF
GETINPUT PROC FAR
    MOV AX,SEG BUF
    MOV FS,AX
    INOUT  FS:BUF,10
    RET
GETINPUT ENDP
CODES ENDS
      END

;-------------
;OUTPUT.ASM
;-------------
.586
INCLUDE IOLIB.LIB
EXTRN BUF:BYTE
PUBLIC DISPLAY
CODES SEGMENT USE16 PARA PUBLIC 'CODE'
      ASSUME  CS:CODES,ES:SEG BUF
DISPLAY PROC FAR
    MOV AX,SEG BUF
    MOV ES,AX
    INOUT   ES:BUF+2,9
    RET
DISPLAY ENDP
CODES ENDS
      END

;-------------
;MAIN.ASM
;-------------
.586
INCLUDE IOLIB.LIB
```

```
51
52  STACKS SEGMENT USE16 PARA STACK 'STACK'
53       DW 128 DUP(0)
54  TOPS LABEL WORD
55  STACKS ENDS
56
57      EXTERN DISPLAY:FAR
58      EXTERN GETINPUT:FAR
59      PUBLIC BUF
60
61  DATAS SEGMENT USE16 PARA PUBLIC 'DATA'
62  BUF DB 21,22 DUP(' ')
63  CR  DB 13,10,'$'
64  DATAS ENDS
65
66  CODES SEGMENT USE16 PARA PUBLIC 'CODE'
67        ASSUME CS:CODES,DS:DATAS
68  START:MOV AX,DATAS
69        MOV DS,AX
70        LEA SP,TOPS
71        CALL FAR PTR GETINPUT
72        ;INOUT BUF,10
73        MOV BH,0
74        MOV BL,BUF+1
75        MOV BUF[2+BX],20H
76        INOUT CR,9
77        CALL FAR PTR DISPLAY
78        MOV AH,4CH
79        INT 21H
80  CODES ENDS
81      END START
82
83
```