

2013-2014 学年第 2 学期《汇编语言程序设计》考试试卷

A 卷 闭卷 考试时间： 2014 年 5 月 15 日

专业_____ 班级_____ 学号_____ 学生姓名_____

题号	一	二	三	四	五	六	七	八	总分	核对人
题分	10	10	10	20	10	10	10	20	100	
得分										

得分	评卷人

一、填空题（共 10 分，每空 1 分）

- 1、访问存放在内存中的数据时，需要知道该数据的物理地址。在实方式下，设 (DS)=1000H, 则指令 MOV AX, DS:[200H] 中源操作数的物理地址是 10200H。
- 2、堆栈也是内存中的一片数据存储区域，要访问栈顶元素，同样要知道其物理地址。在实方式下，设 (SS) = 2000H, (SP)= 100 H，则栈顶元素的物理地址是 20100H。
- 3、一个内存单元的物理地址是唯一的，但是在写指令时，可以由多种方式表达出同一个单元的地址。例如，设在数据段 (DS) 中定义了变量 LEN DW 10H。代码段中有：
LEA BX, LEN
MOV SI, 0
在这两条指令之后，可使用如下几种方式将 LEN 中的内容送入 CX 中。
用直接寻址方式： MOV CX, LEN
用寄存器间接寻址方式： MOV CX, [BX]
用变址寻址方式： MOV CX, LEN[SI] or 0[BX]
- 4、在访问一个内存单元时，除了知道其物理地址外，还应明确其类型，即明确以该地址取一个字节、一个字，还是一个双字。设 BUF DW 1122H, 3344H
则执行 MOV CL, BYTE PTR BUF 后， (CL) = 22H
执行 MOV CX, BUF 后， (CX) = 1122H

执行 MOV ECX, DWORD PTR BUF 后, (ECX) = 33441122H

5、外部设备寄存器只能用 I/O 指令来访问。从端口地址 60H 输入一个字节到 AL 中的指令为: IN AL, 60H。而 “OUT 70H, AL” 的功能是 将 AL 的内容输出到端口地址为 70H 的外设寄存器。

得分	评卷人

二、选择题 (共 10 分, 每空 1 分)

- 指令 MOV [DI], WORD PTR [SI] 的错误原因是 C。
(A) DI 不能用于寄存器间接寻址方式 (B) WORD PTR 的用法错误
(C) 两个操作数不能同时为存储器操作数 (D) 源操作数的类型不明确
- 指令 ADD [CX], AL 的错误原因是 D。
(A) 两个操作数不能同时为存储器操作数 (B) 两个操作数的类型均不明确
(C) 两个操作数的类型不匹配 (D) CX 不能用于寄存器间接寻址方式
- 指令 ADD [BX], 1000H 的错误原因是 B。
(A) 两个操作数不能同时为存储器操作数 (B) 两个操作数的类型均不明确
(C) BX 不能用于寄存器间接寻址方式 (D) 源操作数不能用立即寻址方式
- 指令 ADD AH, BX 的错误原因是 B。
(A) BX, AH 的位置写反了 (B) 两个操作数的类型不匹配
(C) 两个操作数不能同时为存储器操作数 (D) 两个操作数的类型均不明确
- 下面语句中, 源操作数在堆栈段中且语法正确的指令是 A。
(A) MOV AX, SS:[SI] (B) MOV BYTE PTR [BP], '\$'
(C) MOV SS:[SI], AX (D) MOV BYTE PTR [SI], [BP]
- 设 LEN 为符号常量, BUFW 为字变量, 下面语句中有语法错误的语句是 D。
(A) MOV BX, BUFW (B) MOV BUFW, LEN
(C) LEA BX, BUFW (D) MOV LEN, BUFW
- 不能用来访问外部设备寄存器的方法是 A。
(A) MOV 指令 (B) IN / OUT 指令
(C) DOS 系统功能调用 (D) BIOS 功能调用
- 设 (BX) = 7000H, (CX) = 2014H, 则执行 ADD BX, CX 后, 有 D。
(A) SF=0, OF=0 (B) SF=0, OF=1
(C) SF=1, OF=0 (D) SF=1, OF=1
- 设有 BUFB DB 20, 21 DUP(0), 现采用 10 号功能调用输入一个串到 BUFB 缓冲区中。现要将实

实际输入串的长度送入 BX 中，正确的语句是 D。

- (A) MOV BL, BUFB+1 (B) MOV BL, 20
(C) MOV BX, BUFB+1 (D) MOV BL, BUFB+1
MOV BH, 0

10、设 (BX)=2014H, CF=0 则执行循环左移指令 ROL BX, 4 后，有 A。

- (A) (BX) = 0142H, CF=0 (B) (BX) = 0140H, CF=0
(C) (BX) = 0142H, CF=1 (D) (BX) = 0201H, CF=1

得分	评卷人

三、简答题（共 10 分）

1. 描述实方式下，CPU 执行指令 INT m 时，会进行哪些操作（包括从何处得到 m 号中断处理程序的入口段地址及偏移地址）。（5 分）

- (1) 状态寄存器压栈
- (2) 断点的(CS)和(IP)先后压栈
- (3) 从物理地址为 4*m 和 4*m+2 的地方分别取 2 个字节送到 IP 和 CS

2. 简述基于窗口的 Windows 程序中窗口主程序 **WinMain()** 的核心工作流程。（3 分）

- (1) 定义窗口变量并进行初始化，注册窗口类
- (2) 创建窗口类
- (3) 装载菜单等各种资源
- (4) 进入消息循环：从操作系统获取与本窗口程序有关的消息，若是退出消息则返回；否则对消息进行分析并发送到本程序的窗口消息处理函数，然后转(4)

3. 设 (AX) = 9014H, (BX) = 2014H, 在执行指令 **CMP AX, BX** 后，**“JG L1”** 的转移条件是否成立？如果将 **“JG L1”** 换成 **“JA L1”**，转移条件又是否成立？（2 分）

JA L1 成立
JG L1 不成立

得分	评卷人

四、问答题（共 10 分）

1. 一个数据段定义如下：

```
DATA    SEGMENT USE16
X1      DW  5678H
X2      DB  '5678'
LEN      EQU  $ - X1
```

		偏移地址
		00H
X1	78H	01H
	56H	02H
X2	35H	03H
	36H	04H
	37H	05H
	38H	06H
X3	02H	07H
	00H	08H
X4	05H	09H
	05H	

```

X3    DW  X2
X4    DB  2 DUP (5)
DATA  ENDS

```

请在右表格中以字节为单位填写该数据在存储器的存放形式，并标明 X1、X2、X3、X4 所处的位置及偏移地址。（7 分）

X1、X2、X3、X4 的偏移地址分别是：00H、02H、06H、08H

2、执行如下指令后，寄存器的值是什么？（3 分）

```

MOV BX, LEN    (BX)= 06H
MOV SI, X3     (SI)= 0002H
MOV AL, [SI]   (AL)= 35H

```

3、实方式下，设有如下程序：（10 分）

```

        BUF DB 20 DUP (0)
        .....

        INVOKE ITOA, 1234H, 10, OFFSET BUF
        MOV  DX, OFFSET BUF
        .....
ITOA PROC NEAR STDCALL NUM:WORD, RADIX:WORD, RESULT:WORD
        MOV AX, NUM
        MOV BX, RADIX
        .....
        RET
ITOA ENDP
        .....

```

生成的机器代码，反汇编的结果如下：

```

CS:0005  PUSH 001F
CS:0008  PUSH 000A
CS:000A  PUSH 1234
CS:000D  CALL 001B
CS:0010  MOV DX,001F
        .....
CS:001B  PUSH BP
CS:001C  MOV BP, SP

```

CS:001E MOV AX, [BP + 04]

CS:0021 MOV BX, [BP+ 06]

① 试以字节为单位画出刚进入子程序 ITOA 时的堆栈示意图，并在图中标出 NUM, RADIX, RESULT 的位置。(5 分)

② 伪指令 INVOKE 对应的机器指令是什么？参数是按什么顺序压入堆栈的？(3 分)

③ 在生成的目标代码中，读取参数 NUM 的值的语句是什么？(2 分)

(BP-L)	(BP)=(SP), 低地址	
(BP-H)		
1BH	+2	(2) CALL 001B, 从右到左
00H		
34H	+4 NUM	(3) MOV AX, [BP+04]
12H		
0AH	+6 RADIX	
00H		
1FH	+8 RESULT	
00H	高地址	

得分	评卷人

五、程序填空题（共 10 分，每空 1 分）

1. 子程序 F2T16 的功能是将 (BX) 中的内容以 **4 位** 十六进制串的形式显示到屏幕上，请在程序空白处填入空缺的代码。

TAB DB '0123456789ABCDEF'

.....

F2T16 PROC

PUSHA ; 将 8 个 16 位通用寄存器压栈

MOV CX, 4

F2T16_LP: ROL BX, 4; 将 BX 循环左移 4 位

MOV SI, BX

AND SI, 0FH

MOV DL, TAB[SI]

MOV AH, 02H

INT 21H

DEC CX

JNZ F2T16_LP

POPA ; 将 8 个字从堆栈弹出，送相应的寄存器

RET

F2T16 ENDP

2、下面程序段的功能是统计 MSG 中所含字符 'A' 的个数,并送入变量 COUNT 中,请在程序空白处填入空缺的代码。

:

```

MSG DB 'ASSEMBLY LANGUAGE PROGRAMMING'
LEN EQU $ - MSG
COUNT DB 0
    :
    MOV CX,LEN
    MOV BL,0
    MOV SI,OFFSET MSG
LP1: CMP [SI], 'A'
    JNE LP2
    INC BL
LP2: INC SI
    DEC CX
    JNZ LP1
    MOV COUNT,BL

```

得分	评卷人

六、编写程序段（共 10 分）

- 1、设计宏指令 D_ADD，完成 $(X)+(Y) \rightarrow Z$ ，其中 X,Y,Z 都是字类型的变量。要求调用宏指令前、后，所有通用寄存器中的内容保持不变。（5 分）

```

D_ADD MACRO X,Y,Z
    PUSH    AX
    MOV     AX,X
    ADD     AX,Y
    MOV     Z,AX
    POP     AX
ENDM

```

- 2、编写程序段(只要核心代码，有关段定义、假定伪指令均不需要)，统计 BX 中“1”的个数，统计结果放在 CX 中。要求程序段中不出现移位操作指令（5 分）

```

        MOV     CX,0
LP1:    CMP     BX,0
        JZ      LP3
        TEST    BX,8000H
        JZ      LP2
        INC     CX
LP2:    ADD     BX,BX
        JMP     LP1
LP3:    ...

```

得分	评卷人

七、分析程序（共 10 分，每题 5 分）

- 1 阅读程序，回答问题。

.386

```

DATA    SEGMENT USE16
STR     DB      ' This is a test program $'
LEN     DB      0
DATA    ENDS
CODE    SEGMENT USE16
        ASSUME   CS:CODE, DS:DATA
BEGIN:  MOV     AX, DATA

```

```

                MOV    DS, AX
                MOV    SI, OFFSET STR
                MOV    CL, 0
LOP:            MOV    AL, [SI]
                CMP    AL, '$'
                JE     EXIT
                INC    SI
                INC    CL
                JMP    LOP
EXIT:           MOV    LEN, CL
                MOV    AH, 4CH
                INT    21H
CODE            ENDS
                END    BEGIN

```

(1) 该程序完成什么功能? (3 分)

统计字符串 STR 中第一个 ‘\$’ 字符前面的字符个数并存放到变量 LEN 中。

(2) 如果将 INC SI 漏写, 结果如何? (2 分)

若 STR 中的第一个字符为 ‘\$’, 则程序正常执行 (结果为 0) ; 否则死循环。

2、阅读程序, 解答问题。

```

CODE    SEGMENT USE16
        ASSUME CS:CODE
BEGIN:  CALL OUTPUT

        LP1: DB  'VERY GOOD $', 0DH, 0AH, 0

        LP2: CALL OUTPUT
        LP3: DB  'ASSEMBLY LANGUAGE', 0
LP4:    MOV AH, 4CH
        INT 21H

OUTPUT PROC
        POP SI

```



```

LOOPA:  MOV DL, CS:[SI]
        INC SI
        CMP DL, 0
        JZ EXIT
        MOV AH, 2
        INT 21H
        JMP LOOPA
EXIT:   PUSH SI
        RET
OUTPUT ENDP
CODE   ENDS
        END BEGIN

```

(1) 运行该程序后，屏幕上会显示什么？（2分）

VERY GOOD
ASSEMBLY LANGUAGE

(2) 第一次调用子程序，并执行子程序的 RET 指令后，程序会转移到何处？为什么？（3分）

程序会转移到 LP2。因为子程序在 EXIT 处 SI 的内容为 LP2 的偏移地址，它被压入堆栈后紧接着执行 RET 指令，RET 指令将从堆栈中弹出一个字到 IP 寄存器中，这时 (IP) 为 LP2 的偏移地址。

得分	评卷人

八、程序设计（20 分）

设以 BUF1、BUF2 为首地址的字节存储区中，均存放了以数字 0 为结束标志的字符串。试编写一个完整的程序，比较两个存储区中的字符串是否相同。若相同，则显示 SAME, 否则显示 NOT SAME。

要求：

- (1) 比较两个串是否相同，要用子程序 STRCMP 来实现。两个串的首地址要通过堆栈方式传递，也即子程序从堆栈中获得两个要比较串的首地址。若两个串相同，子程序返回(AX)=1; 否则返回(AX)=0。
- (2) 在主程序中显示比较结果信息。
- (3) 写出子程序中寄存器的使用分配情况，并给出必要的注释。

- (4) 程序完整（包括数据段定义、堆栈段定义、代码段定义等）。
- (5) BUF1, BUF2 中字符串的内容自己设定。

```

STACK      SEGMENT USE16 STACK
            DB 200 DUP(0)
STACK      ENDS
DATA       SEGMENT USE16
BUF1       DB '1234abcd', 0
BUF2       DB '1234abc', 0
MSGOK      DB 'SAME', '$'
MSGERR     DB 'NOT SAME', '$'
DATA       ENDS
CODE       SEGMENT USE16
            ASSUME CS:CODE, DS:DATA, SS:STACK
START:     MOV     AX, DATA
            MOV     DS, AX
            MOV     AX, OFFSET BUF2
            PUSH    AX
            MOV     AX, OFFSET BUF1
            PUSH    AX
            CALL    STRCMP
            MOV     DX, OFFSET MSGERR
            CMP     AX, 0
            JZ      NEXT
            MOV     DX, OFFSET MSGOK
NEXT:      MOV     AH, 9
            INT     21H
            MOV     AX, 4C00H
            INT     21H

```

;;比较 2 个字符串，将使用 3 个寄存器：AX、SI、DI

```

STRCMPPROC NEAR
            PUSH    BP
            MOV     BP, SP
            MOV     SI, [BP+4]      ;字符串 1 的偏移地址
            MOV     DI, [BP+6]      ;字符串 2 的偏移地址
LP:         MOV     AL, [SI]
            MOV     AH, [DI]
            CMP     AX, 0
            JZ      LPEQ            ;2 个字符串具有相同的长度，并且全部相同
            CMP     AL, 0
            JZ      LPNEQ          ;字符串 1 结束但字符串 2 没有结束
            CMP     AH, 0
            JZ      LPNEQ          ;字符串 2 结束但字符串 1 没有结束
            CMP     AH, AL
            JNZ     LPNEQ          ;2 个字符串相同位置的字符不相同
            INC     SI

```

```

        INC     DI
        JMP     LP
LPEQ:   MOV     AX, 1           ;2 个字符串相等
        JMP     LPEXIT
LPNEQ:  MOV     AX, 0           ;2 个字符串不相等
LPEXIT: POP     BP
        RET
STRCMP  ENDP
CODE    ENDS
        END    START

```