

## 《汇编语言程序设计》课程目标（考核点）

支撑的毕业要求二级指标点	对应本课程目标	考核点示例	占比
1.2 能针对计算机复杂工程问题的具体对象进行建模和求解	<b>目标 1:</b> 掌握汇编语言的概念与特点, 理解 80X86 计算机的基本工作原理; 掌握阅读和分析汇编语言程序的方法; 加深对高级语言内部实现机制以及计算机软、硬件系统的整体化理解; 能从汇编语言的角度描述和求解计算机的工程问题。	<p><b>【是否掌握建模基础】</b></p> <p>1) 概念 (如: 什么是汇编、中断矢量、保护方式?); 2) 原理 (如: 物理地址形成、指令读取、子程序调用返回、转移指令、IO 与中断机制等); 3) 方法 (如: 变量与宏如何定义、子程序如何编写、模块化和 WIN32 程序如何编写等); 4) 表示法 (汇编语句的表达, 流程图等)。</p> <p><b>【能否给出求解方案】</b></p> <p>1) 利用建模基础给出方案描述 (如: 设计思路描述、寄存器分配、内存单元分配、流程图等); 2) 分析和完善已有求解方案 (如: 分析程序段或完整源程序的功能, 修改、完善程序等)。3) 给出满足要求的源程序 (程序段或完整源程序)。</p>	70%
3.2 能为计算机复杂工程问题解决方案设计满足特定需求的软/硬件模块	<b>目标 2:</b> 掌握编写、调试汇编语言程序的方法; 能设计出较充分利用了汇编语言优势的软件功能模块。	<p>1) 汇编语言本身的特点 (如: INT 等指令特点、段定义、地址表等); 2) 编写、调试中的特点 (如: 工具的特点、中断的调试方法等); 3) 编写、调试具有汇编语言优势的模块 (如: 程序优化、地址直接操纵、中断处理、反跟踪等)。</p>	30%

2020 年 3 月

# 计算机科学与技术学院 2019-2020 学年第 2 学期考试试卷

## 汇编语言程序设计 样卷 闭卷

姓名\_\_\_\_\_班级\_\_\_\_\_学号\_\_\_\_\_考试日期 2020-08-XX

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

得分	评卷人

### 一、阅读程序，改正与完善程序（共 20 分）。

1. 下列程序的功能是：用户输入一个字符，将输入字符的 ASCII 码按照十六进制的形式显示在屏幕上，当用户输入回车时，程序退出，否则继续循环等待输入。显示格式如下所示：

THE ASCII CODE OF A IS 41H

THE ASCII CODE OF Z IS 37H

THE ASCII CODE OF     （当用户输入回车时，程序退出）

请将程序中的语法错误和逻辑错误圈出来并在其右侧写出正确的形式（每改正一处 1 分，共 10 分）。

.386

DATA SEGMENT USE16

BUF DB 'THE ASCII CODE OF '

BUF1 DB ' IS \$'

BUF2 DB ' ', 0, 0, 0A, 0DH, '\$'

TAB DB '0123456789ABCDEF'

DATA ENDS

STACK SEGMENT USE16 STACK

100 DUP(0)

STACK ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:STACK

BEGIN: MOV DS, DATA

MOV DX, OFFSET BUF

MOV AH, 9

INT 21H

NEXT: MOV AH, 1

INT 21H

CMP AL, 0DH

JE EXIT

PUSH AX

LEA DX, BUF1

MOV AH, 9

INT 21H

POP CL

MOV AL, CL

AND AL, 0FH

MOV BX, AL

```

MOV AL, TAB[BX]
MOV BUF2+2, AL
MOV AL, CL
AND AL, F0H
SHR AL, 4
MOVZX BX, AL
MOV AL, TAB[BX]
MOV BUF2+1, AL
MOV DX, OFFSET BUF2
MOV AH, 9
INT 21H
JMP NEXT
EXIT: MOV AX, 4CH
      INT 21H
CODE ENDS
END BEGIN

```

2. 下面程序的功能是：从键盘输入一个字符串到 STR 中，测试字符串中是否存在 '+' 或者 '-' 字符。如存在，则把 DL 的最高位为 1，否则置为 0；DL 其他位内容保持不变（请完善程序，每处 1 分，共 10 分）。

```

.386
STAB SEGMENT USE16 STACK
      DB 200 DUP(0)

_____
DATA SEGMENT USE16
STR1 DB 20
      DB ?

_____
DATA ENDS
CODE SEGMENT USE16 'CODE'
      ASSUME CS:CODE, SS: STAB, DS: DATA
KAISHI: MOV AX, DATA
        MOV DS, AX
        MOV AH, 10

_____
        INT 21H

_____

NEXT:  MOV AL, [SI]
        CMP AL, '+'

_____
        CMP AL, '-'

```

```

L1:
    JMP  EXIT
L2:
    INC  SI
    LOOP NEXT

EXIT: MOV  AH,4CH
      INT  21H
CODE  ENDS

```

得分	评卷人

## 二、根据计算机工作的基本原理，完善以下描述（共 10 分，每空 1 分）。

存储程序和程序控制是计算机工作的基本原理，要执行的程序预先存放在内存中。实方式下，CPU 当前要解释执行的指令的物理地址由段首址和偏移地址计算得到。在取出指令后，需要对该指令进行译码，此时指令指示器(IP)会自动加上该指令的字节数，使其指向该指令的下一条指令。在执行该指令时，若指令不是转移类（包括条件转移、无条件转移、子程序调用和返回、软中断调用和返回等）的指令时，就不会影响 CS 和 IP。这样，取出的下一条指令就在该指令之下的存储单元，程序将会顺序执行。若该指令是段内子程序调用语句，CPU 将首先将（IP）压栈，然后将子程序的偏移地址送入到 IP。这样取下一条指令时，取出的就是子程序中的第 1 条要执行的指令。若该指令是 JMP WORD PTR[BX]，则该指令被称为无条件间接转移指令，转移到的目标指令的偏移地址存放在代码段中，通过寄存器间接寻址方式取到相应单元的值送给 IP。但在 WIN32 下，指令指示器就需要使用 EIP 了，因为内存的偏移地址范围从 16 位上升到了32位；而且代码段地址空间以及 IO 地址空间都可以受到保护机制的保护。

得分	评卷人

### 三、分析与优化程序（共 10 分）。

如下的 C 语言程序段实现了一个 long 型数组中 10 个数据求和功能，其编译后调试版本的汇编语言代码如下(注：为便于阅读，将有些具体的地址用符号代替了，斜体部分为 C 语句，x,i 均是 long 类型的变量)。(共 10 分)

```

x=0;
    MOV     x, 0
for (i=0;i<10;i++)
    MOV     i, 0
    JMP     L1
L0:    MOV     EAX, i
        ADD     EAX, 1
        MOV     i, EAX
L1:    CMP     i, 0Ah
        JGE     L2
x=x+a[i];
        MOV     ECX, i
        MOV     EDX, x
        ADD     EDX, a[ECX*4]
        MOV     x, EDX
        JMP     L0
L2:

```

(1) 指出该段程序执行效率不高的原因 （2 分）。

(2) 改编相应的汇编程序，以提高程序的执行效率(5 分)。  
 要求写出变量与寄存器对应关系，尽可能与调试版本一致。

(3) 实方式下，假设指令 JGE 的操作码是一个字节，则该程序中的 JGE L2 的机器码共有几个字节，为什么？（3 分）

得分	评卷人

四、简答题（共 20 分）。

在下列段定义中，设 DATA 与 STACK 段相邻存放，DATA 的段首址为 2000H，BUFA 的偏移地址为 0，请回答下面的问题。

DATA SEGMENT USE16

BUFA DB 1, 3, 5  
BUFB DW -2H, 1234H  
BUFC DW BUFB  
DB 2 DUP('3', 'C')  
LEN EQU \$ - BUFC  
BUFD DB 200 DUP(0)

DATA ENDS

STACK SEGMENT USE16 STACK

DB 16 DUP(0)

STACK ENDS

- (1) 以字节为单位，画出数据段的数据存储示意图（不用画 BUFD 的内容）。（6 分）
- (2) 变量 BUFC 的物理地址是\_\_\_\_\_（1 分）。
- (3) 执行下列各指令语句后,写出各寄存器的内容。（3 分）

MOV SI, DS:[1]  
MOV EBX, LEN  
MOV ECX, DWORD PTR BUFC

低地址端

高地址端


(SI) = \_\_\_\_\_  
(EBX) = \_\_\_\_\_  
(ECX) = \_\_\_\_\_

- (4)程序在执行过程中发现堆栈段空间需要临时让出来并需要增加 128 个字节，想通过临时挪用部分 DATA 段空间的方法（比如借用 BUFD 所在的空间）实现。请给出实现该功能的源程序段。（10 分）

得分	评卷人

## 五、程序分析（共 20 分）。

### 1. 阅读程序，回答问题。（共 10 分）

```

.386
DATA SEGMENT USE16
STRING DB '12ABCKJHaaabjufdsalb47'
LEN EQU $ - STRING
DATA ENDS
STACK SEGMENT USE16 STACK
DB 200 DUP(0)
STACK ENDS
CODE SEGMENT USE16
ASSUME CS:CODE,DS:DATA,SS:STACK
BEGIN:
    MOV AX,DATA
    MOV DS,AX
    LEA SI, STRING
    MOV CX, LEN ; ----- ①
NEXT: MOV AL, [SI] ; ----- ②
    CMP AL, 'A'
    JB L1
    CMP AL, 'Z'
    JA L1
    SUB AL, 'A' - 'a'
    MOV [SI], AL
L1: INC SI ; ----- ③
    LOOP NEXT
    MOV AH, 4CH
    INT 21H
CODE ENDS
END BEGIN

```

(1) 上述程序的功能是什么？（5 分）

(2) 若将语句②处的标号 NEXT 上移一行，误写到语句①处，则程序执行结果会怎样？（2 分）

(3) 若漏写了语句③，程序功能会发生什么变化？（3 分）

2. 阅读程序，回答问题。（共 10 分）

```
OUTB MACRO A
    MOV DL,A
    MOV AH,2
    INT 21H
ENDM

.386
STACK SEGMENT USE16 STACK
    DB 200 DUP(0)
STACK ENDS
DATA SEGMENT USE16
BUF DW 11, 22, 33, 44
N EQU ($-BUF)/2
DATA ENDS
CODE SEGMENT USE16 'CODE'
    ASSUME CS:CODE, SS: STACK, DS: DATA
BEGIN: MOV AX, DATA
    MOV DS, AX
    MOV CX, N
    LEA SI, BUF
NEXT1: MOV BX, [SI]
    CALL RADIX
    OUTB ''
    ADD SI, 2
    LOOP NEXT1
EXIT: MOV AH,4CH
    INT 21H

RADIX PROC
    PUSH CX ; ----- ①
    PUSH DX ; ----- ②

    ROL BX, 1 ; ----- ⑤
    MOV DX, BX
    AND DX, 1
    ADD DL, '0'
    OUTB DL ; ----- ⑥

    MOV CX, 5
NEXT2: ROL BX, 3
    MOV DX, BX
    AND DX, 7
    ADD DL, '0'
    OUTB DL
```



```

        LOOP NEXT2

        POP DX      ; ----- ③
        POP CX      ; ----- ④
        RET
RADIX ENDP
CODE ENDS
        END BEGIN

```

(1) 上述程序中，子程序 RADIX 的功能、入口参数是什么？（5 分）

(2) 语句①、②、③、④的作用是什么？如果省略，则程序执行结果会怎样？（3 分）

(3) 如果从语句⑤到语句⑥都省略，则程序执行结果会怎样？（2 分）

得分	评卷人

**六、完整程序设计（20 分）。**

在 BUF 字缓冲区中存放了 N 个字类型有符号数。编写完整程序，计算这 N 个数的平均值 AVG(保留整数部分)。将 BUF 数组按大于等于 AVG 和小于 AVG 进行划分，分别保存在数组 BUFA 和 BUFB 中，并以十进制形式输出 BUFA 和 BUFB 的元素个数（假设元素个数都小于 10）。

- 要求：**(1) 简要描述设计思想，给出寄存器分配方案。
- (2) 画出主程序和子程序的流程图。
- (3) 用子程序 DISP 实现小于 10 的无符号数的十进制显示输出功能，描述其入口参数、出口参数。
- (4) 程序完整（包括堆栈段、数据段、代码段定义等），至少给出 4 条必要的注释。