



华中科技大学计算机科学与技术学院 2019~2020 第二学期
“汇编语言程序设计”考试试卷 (A 卷)
答案与评分标准

考试方式 闭卷 考试日期 2020-08-31 考试时长 150 分钟

专业班级 学 号 姓 名

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

得分	评卷人

一、填空题 (共 20 分, 每空 1 分)

1、假设在实方式下的数据段中定义有如下变量 (DS 中的内容已为数据段的首址):

X DW 1122H, 3344H

PX DW X

现需要将 X 中的第 2 个字单元中的内容 (即 3344H) 送入 AX 中。请写出使用以下 5 种不同的方式访问该单元的指令语句或语句组合 (一条横线上只能写一条指令语句): ①直接寻址

方式: MOV AX, X+2。 ②寄存器间接寻址方式: LEA SI, X + 2、

MOV AX, [SI]。 ③变址寻址方式: LEA SI, X、MOV AX, [SI+2]。

④基址加变址寻址方式: LEA SI, X、MOV BX, 2、MOV AX, [BX + SI]。

⑤ 通过变量 PX 间接访问: MOV SI, PX、MOV AX, [SI+2]。

2、实方式下, CPU 在执行完一条指令后, 会检查是否有中断发生。若检测到了中断信号 (设为 m 号中断), 并且要响应该中断时, CPU 会将标志寄存器中的内容压栈, 然后, 将 (CS) 压栈, 再将 (IP) 压栈, 最后, 从内存的最低端 (段地址为 0)、偏移地址为 m*4 处取一个字单元内容送入 IP 中; 取该单元的下一个字单元的内容送给 CS。在此之后, CPU 根据 CS 和 IP 取指令, 就可以得到中断处理程序中第一条待执行的指令。在中断处理程序执行结束时, 会有 IRET 指令, 该指令完成的操作是 出栈到 IP、出栈到 CS、弹出一个字送给标志寄存器。

3、Windows 窗口应用程序一般由主程序、窗口主程序、窗口消息处理程序和 用户子程序组成, 窗口主程序的主要功能是在窗口参数初始化后 注册窗口类、创建窗口和加载窗口资源, 最后进入消息循环。

评分标准: 每空 1 分。

题 1: (1) 将 MOV AX, X+2 写成 MOV AX, X+1, 得 0.5 分;

(2) 将 3344H 的偏移地址看成 2, 得 0.5 分, 如第 2 和 3 个空写为: MOV BX, 2、MOV AX, [BX], 得 1.5 分;

题 2: (1) 将 CS、IP 写成 IP、CS, 得 0.5 分, 是否加括号 () 不扣分;

(2) 将 出栈到 IP、出栈到 CS 写反了, 得 0.5 分;

题 3: (1) 将 创建窗口 写成 显示窗口, 得 0.5 分。

得分	评卷人

二、简答题（共 20 分）

1. 设一个程序中有如下段定义（STACK 段紧接在 DATA 段之后）：

```
DATA SEGMENT USE16
STR1 DB '158'
LEN = $ - STR1
STR2 DB LEN DUP(0)
SX DW 2019H, 3344H
BY DB 2 DUP(6, '7')
PX DW SX
DATA ENDS
STACK SEGMENT USE16 STACK
DB 200 DUP(0)
STACK ENDS
```

请在右表格中以字节为单位填写 DATA 段中各数据在存储器的存放形式，并标明各变量所处的位置及偏移地址。（10 分）

'1'	0000H	STR1
'5'	0001H	
'8'	0002H	
0	0003H	STR2
0	0004H	
0	0005H	
19H	0006H	SX
20H	0007H	
44H	0008H	
33H	0009H	
6	000AH	BY
'7'	000BH	
6	000CH	
'7'	000DH	
6	000EH	PX
0	000FH	

评分标准：
STR1 顺序与 ASCII 表示各 1 分；
LEN 不出现在表中 1 分；
STR2 数量与值 1 分；
SX 高低位与顺序各 1 分；
BY 1 分；
PX 1 分；
变量名位置表示 1 分，
每错 1 个扣 0.5 分；偏移地址 1 分，每错一个扣 0.5 分。

2、写出一个宏定义，宏名为 SSTODS，所带的参数为段首址 DATA。其功能是将堆栈段与数据段合并成一个段。（10 分）提示：逻辑地址由“段址：段内偏移”组成。同一个物理单元，可以有多个逻辑地址，实方式下，物理地址=段址*10H+段内偏移。合并两个段，即（DS）保持不变，改变（SS）的值，使得（SS）与（DS）相同，但由于当前栈顶的物理位置并没要求改变，所以当（SS）被修改后，（SP）也必须修改。

```
SSTODS MACRO DATA
CLI
MOV AX, SS
SUB AX, DATA
SHL AX, 4
ADD SP, AX
MOV AX, DATA
MOV SS, AX
STI
ENDM
```

评分标准：
宏定义共 4 分，每个单词 1 分；
开关中断 1 分；
修改 SS 2 分；
修改 SP 3 分（计算差值、左移、求和各 1 分）。

得分	评卷人

三、程序填空与改错题（共 20 分，每空 1 分）

1. 下面的子程序 STRCMP 的功能类似于 STRCMP (STR1, STR2)，即比较两个以 0 作为结束符的字符串 STR1 与 STR2 的大小关系。若 STR1>STR2，返回 AX 为 1；若 STR1=STR2，返回 AX 为 0；若 STR1<STR2，返回 AX 为-1。请根据要求补充有关语句（每处 1 分，共 10 分）。

实方式下，在数据段中定义了 STR1 和 STR2，其中，以 STR1 为首地址的串中内容为 hello，其定义语句为：

STR1 DB 'hello', 0

在主程序中，给出子程序的调用语句：

PUSH OFFSET STR2

PUSH OFFSET STR1

CALL STRCMP

.....

； 入口参数，两个字符串的首地址 ； 出口参数，AX 为 1，第 1 个串>第 2 个串

STRCMP PROC NEAR

PUSH BP

MOV BP, SP

PUSH SI

PUSH DI

MOV SI, [BP + 4] ；（SI） 为后入栈的串的首地址

MOV DI, [BP + 6]

LP: MOV AL, [SI]

CMP AL, [DI]

JA BIGGER

JB SMALLER

CMP AL, 0

JZ STR_EQU

INC SI

INC DI

JMP LP

BIGGER:

MOV AX, 1

JMP EXIT

SMALLER:

MOV AX, -1

JMP EXIT

STR_EQU:

MOV AX, 0

EXIT:

POP DI

POP SI

POP BP

RET 4 ； 4 不一定需要

STRCMP ENDP

2. 下列程序的功能是：用户输入一个无符号的数字字符串，然后将该串转换为一个字类型的数据（不考虑数值溢出）。请将程序中的语法错误和逻辑错误圈出来（圈出具体的错误位置）并在其右侧写出正确的形式（请重点关注带*的行，每改正一行中的错误得 1 分，共 10 分）。

```
.386
DATA SEGMENT USE16
BUF DB 6
    DB 7 DUP(0)
DATA ENDS
STACK SEGMENT USE16 STACK
    200 DUP(0)          ; * DB 200 DUP(0)
STACK ENDS
CODE SEGMENT           ; * CODE SEGMENT USE16
    ASSUME CS:CODE,DS:DATA,SS:STACK
BEGIN: MOV AX,DATA
    MOV DS,AX
    MOV DX,BUF          ; * MOV DX, OFFSET BUF
    MOV AH,9            ; * MOV AH,10
    INT 21H
    MOV AX,0
    MOV BX,10
    MOV SI,0
LP:
    MOV CX,BUF[SI+2]     ; * MOV CL,BUF[SI+2]/MOVZX CX,BUF[SI+2]
    CMP CL,0D           ; * CMP CL,0DH
    JZ EXIT
    IMUL BX              ; * MUL BX
    SUB CL,'0'
    ADD AX,CL            ; * MOV CH,0 、 ADD AX,CX / ADD AX,CX
    INC SI
    JMP LP
EXIT: MOV AH,4C00H      ; * MOV AH,4CH/MOV AX,4C00H
    INT 21H
CODE ENDS
    END START           ; * END BEGIN
```

评分标准：

每空/行 1 分。

第 1 题：(1) 字符串未写结尾 0，得 0.5 分；

(4)、(5)，使用 BP 间接寻址，位移量不对，得 0.5 分；

(6)、(7)，使用有符号跳转 JG、JL，得 0.5 分

第 2 题：(8) 未考虑高位清 0，只写了 MOV AX,CX，得 0.5 分；

得分	评卷人

四、程序阅读理解（10 分）

阅读下面的程序，回答问题。

```
.386
DATA SEGMENT USE16
BUF DB 'I Like Assembly Language'
LEN EQU $ - BUF
COUNT DB 26 DUP(0)
DATA ENDS
CODE SEGMENT USE16
    ASSUME CS:CODE, DS:DATA
BEGIN: MOV AX, DATA
        MOV DS, AX
        LEA DI, BUF
        MOV CX, LEN          ; ----- ①
NEXT: MOV BL, [DI]           ; ----- ②
        CMP BL, 'a'
        JB L1
        CMP BL, 'z'          ; z 是小写字母
        JA L1
        SUB BL, 'a'
        MOVZX BX, BL
        INC COUNT[BX]
L1: INC DI                    ; ----- ③
        DEC CX
        JNZ NEXT
        MOV AH, 4CH
        INT 21H
CODE ENDS
END BEGIN
```

(1) 上述程序的功能是什么？（3 分）

统计 BUF 字符串中，'a' — 'z' 26 个小写字母出现的次数，并依字母顺序次保存在字节数组 COUNT 中对应位置。

评分标准：统计 BUF 字符串中 26 个小写字母出现的次数 2 分，以字母顺序保存在 COUNT 的对应位置 1 分。

(2) 如果将①处的语句，写成了“MOV CX, 0”，程序执行结果会怎样？（2 分）

重复执行循环 65536 次（10000H 次），统计以 BUF 为首址长度是 10000H 的串中小写字母出现的次数，可能出现溢出，因而程序执行结果不确定。

评分标准：重复循环 65536 次（10000H 次）得 1 分，统计 10000H 的字符串中小写字母出现次数得 1 分。

(3) 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）

若 LEN < 1，则死循环；若 LEN = 1，执行 1 次退出。

评分标准：仅回答死循环不扣分。

(4) 若漏写了语句③，程序功能会发生什么变化？（3 分）

始终检查 BUF 第一个字符，若第一个字符为小写字母，则对应次数为所有小写字母出现次数和 LEN 的值；若第一个字符不为小写字母，则对应次数为 0。其他所有字母对应的次数为 0。

评分标准：统计 BUF 第一个字符 1 分，对应次数为 LEN 1 分，小写字母 1 分。若答所有字母对应次数（或 COUNT）均为 0 则给 3 分。

得分	评卷人

五、分析与优化程序（共 10 分）

如下的 C 语言程序段实现了找一个整型数组(int a[5];)中的最小数并放入 x 中的功能，其编译后调试版本的汇编语言代码如下(注：斜体部分为 C 语句，x, i 均是 int 类型的变量)。(10 分)

```

    x = a[0];
009213F1  mov     eax,4
009213F6  imul    ecx,eax,0
009213F9  mov     edx,dword ptr a[ecx]
009213FD  mov     dword ptr [x],edx
    for (i = 1; i <= 4; i++){
00921400  mov     dword ptr [i],1
00921407  jmp     wmain+62h (0921412h)
00921409  mov     eax,dword ptr [i]
0092140C  add     eax,1
0092140F  mov     dword ptr [i],eax
00921412  cmp     dword ptr [i],4
00921416  jg      wmain+80h (0921430h)
    if (x > a[i])
00921418  mov     eax,dword ptr [i]
0092141B  mov     ecx,dword ptr [x]
0092141E  cmp     ecx,dword ptr a[eax*4]
00921422  jle     wmain+7Eh (092142Eh)    ; 机器码 7E 0A
    x = a[i];
00921424  mov     eax,dword ptr [i]
00921427  mov     ecx,dword ptr a[eax*4]
0092142B  mov     dword ptr [x],ecx
    }
0092142E  jmp     wmain+59h (0921409h)
00921430  .....

```

(1) 指出该段程序执行效率不高的原因（2 分）。

存在重复多余语句，特别是循环体内存在多余语句，对于内存中的 x,i,a[] 变量多次重复读写。

中间结果未使用寄存器进行缓存。

(2) 改编相应的汇编程序，以提高程序的执行效率(6 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。

变量与寄存器对应关系：i => ebx; x => eax

```

mov ebx, 0
mov eax, a[ebx*4]
e1:  inc ebx
    cmp ebx, 4
    jg exit
    cmp eax, a[ebx*4]
    jle e1
    mov eax, a[ebx*4]
    jmp e1
exit: mov x, eax

```

评分标准：

变量与寄存器对应关系 1 分；

程序段共 5 分。其中，没有将最小值从寄存器送回 x，

扣 0.5 分。ebx*4 中*4 写掉，扣 1 分。

(3) 解释 jle 指令语句的机器码（7EH, 0AH）中 0AH 代表的含义（2 分，卓越班仅 1 分）

0AH 为 jle 转移到的目标地址与 jle 下一条指令地址之间的差值。

评分标准：只回答偏移地址，给 1 分。回答跳转到 IP+0AH，未回答是下一条指令，或者写成相对于本指令的差

值，给 1.5 分。

(4) （仅卓越班做）某编译器将一段 C 语言编译成如下的汇编代码，请分析 cmovge 指令的功能。

(1 分)

if (x >= y)	mov	ecx, y	大于等于时传送。回答标志位的组合 SF=OF 也算正确。
i = x;	cmp	x, ecx	
else	cmovge	ecx, x	
i = y;	mov	i, ecx	

得分	评卷人

六、完整程序设计（20 分）

设以 BUFA、BUFB 为首地址的存储区中，分别存储有 N 个和 M 个非零的字数据。现编一个程序，将在两个存储区中都出现的数据拷贝到 BUFC 中（假设 BUFA 中的 N 个数互不相同，N 和 M 自定）。（ACM 班要求：BUFA 中的 N 个数可能相互相同，要避免重复拷贝）。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 FIND 判断一个数是否在指定的某个存储区中出现，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。

算法思想：

采用循环的方法，对 BUFA 中的数，逐一都调用 FIND 子程序，判断是否在指定的缓冲区中出现，若出现，则存放到缓冲区 BUFC 中。

ESI: BUFA 中变址寄存器
EDI: BUFB 中变址寄存器
EBP: BUFC 中变址寄存器
AX: BUFA 中查找数
ECX: 计数器

FIND 子程序入口参数：

AX: 查找数
EDI: 存储区首址
EDX: 存储区长度
出口参数：
BX: 0 表示未找到，1 表示找到

```
.386
STACK  SEGMENT USE16 STACK
        DB 200 DUP(0)
STACK  ENDS
DATA    SEGMENT USE16
        BUFA  DW 1, 2, 3, 4, 5
        N     EQU ($-BUFA)/2
        BUFB  DW 3, 5, 7
        M     EQU ($-BUFB)/2
        BUFC  DW N DUP(0)
DATA    ENDS
CODE    SEGMENT USE16
        ASSUME CS:CODE, SS:STACK, DS:DATA, ES:DATA
START:  MOV  AX, DATA
        MOV  DS, AX
        MOV  ESI, OFFSET BUFA    ; 初始化
        MOV  EBP, OFFSET BUFC
        MOV  ECX, N
L2: MOV  AX, [ESI]
        MOV  EDI, OFFSET BUFB    ; 设置子程序入口参数
```

```

MOV    EDX, M
CALL   FIND      ; 调用子程序
CMP    BX, 1      ; 检查查找结果
JNE    L1
MOV     DS:[EBP], AX    ; 若找到, 将数据复制到 BUFC 中
ADD     EBP, 2
L1: ADD  ESI, 2
LOOP   L2
MOV    AH, 4CH
INT    21H
; FIND : 在以 (EDI) 为首址的字符串缓冲区中, 找 (AX) 是否出现; 若出现, 置 (BX) =1
; 入口参数 AX: 查找数    EDI: 存储区首址  EDX: 存储区数据个数
; 出口参数: BX: 0 表示未找到, 1 表示找到
FIND  PROC
    PUSH EDI      ; 保护现场
    PUSH EDX
    MOV  BX, 1
F1: CMP  AX, [EDI] ; 在存储区中查找
    JE   F2
    ADD  EDI, 2
    DEC  EDX
    JNZ  F1
    MOV  BX, 0      ; 未找到, 设置 BX 为 0
F2: POP  EDX        ; 恢复现场
    POP  EDI
    RET
FIND  ENDP
CODE  ENDS
      END  START

```

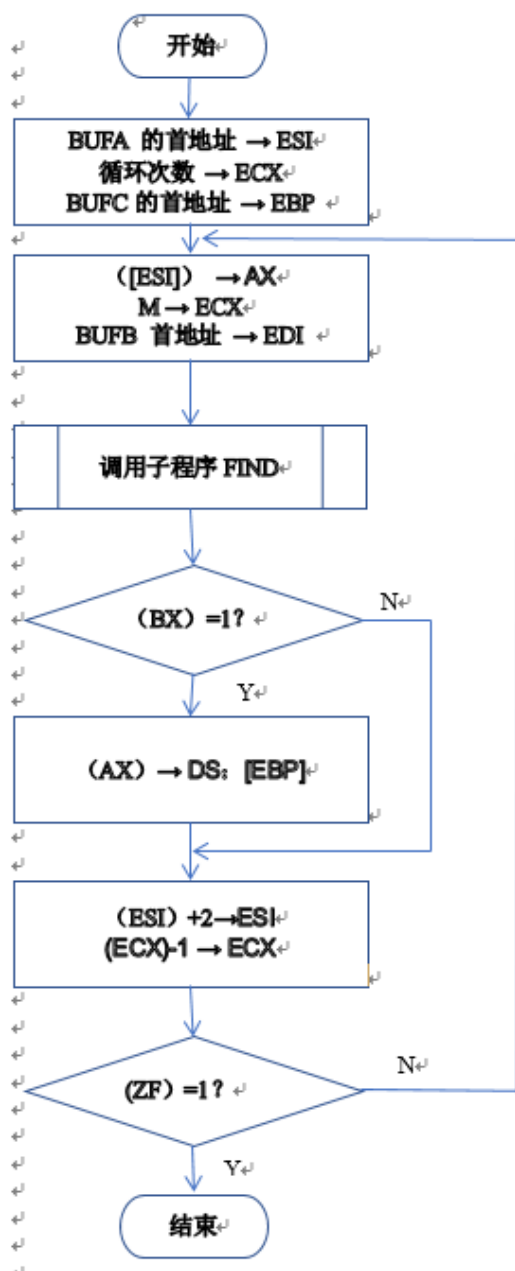
评分标准:

- (1) 文档完整 3 分
 - 画流程图 1 分、算法思想描述 1 分、寄存器分配及注释 1 分
- (2) 程序基本结构 5 分
 - 数据段定义 2 分、堆栈段定义 1 分、代码段定义 (含 ASSUME, END) 1 分
 - 给 DS 赋值及返回 DOS 1 分
- (3) 主程序(5 分)
 - 循环控制 1 分、取数据 1 分、传参数 1 分、调用子程序 1 分、相同数保存 1 分
- (4) 子程序 7 分
 - 子程序结构 及位置 1 分
 - 寄存器保护与恢复 1 分
 - 取参数 1 分
 - 循环控制、比较 2 分
 - 返回参数 1 分
 - 子程序返回 1 分

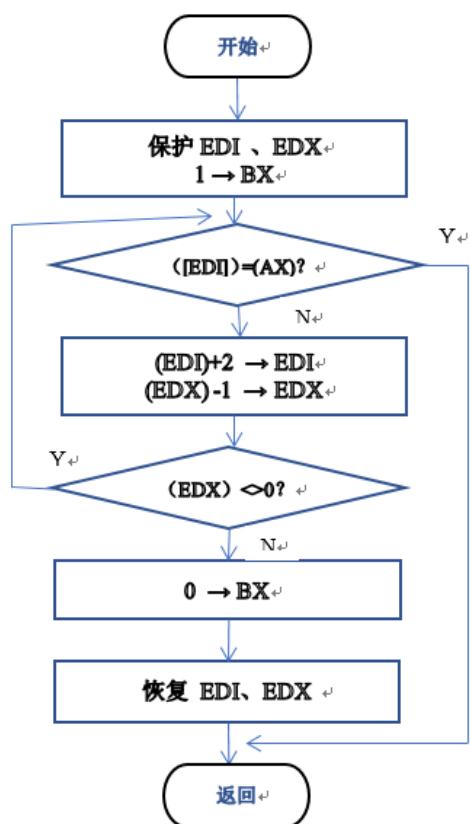
说明:

即使程序运行结果正确, 但未按要求设计子程序的功能, 将不给相应的分数。

(ACM 班若答案没有避免重复则扣 2 分)



主程序流程图



子程序 FIND 流程图

