

汇编语言 期中考试

类型：独立完成开卷考试，不得参考他人（包括往届）答案

截止日期：本学期最后一周周日（含）

上交方式：打印，手写，扫描为pdf后交给助教，文件格式“期中-姓名-学号.pdf”

姓名：陈静雯

学号：202200130048

诚信声明：我承诺本开卷考试由本人独立手写扫描完成，没有参考他人的答案。
(在下面抄写)

我承诺本开卷考试由本人独立手写扫描完成，没有参考他人的答案。

一、单项选择题 (10x2 = 20)

1. 将机器语言目标文件组装成可执行文件的实用程序是 (C)。
A. 解释程序 B. 编译程序
C. 链接程序 .obj → .exe 或 .com D. 汇编程序 汇编器, 生成初步的机器码目标文件 .asm → .obj
2. 设 SP 初值为 1000H, 执行指令 CALL FAR PTR PROC 后, SP 为 (B)。 C. 压栈 . SP = SP - 2.
A. 0FFFH B. 1002H
C. 0FFE H D. 0FFCH
3. 能保存当前执行指令的下一条指令的地址的寄存器称为 (A)。
A. 指令指针寄存器 B. 数据寄存器
C. 段寄存器 D. 地址寄存器
4. 完成将 AL 和 CF 清零不能使用的指令是 (D)。
A. SUB AL, AL B. XOR AX, AX OF. 对 CF 清零.
C. AND AL, 0D. MOV AL, 0 无法影响标志位.
5. 下列指令中错误的一项是 (D)。
A. MOV WORD PTR SS:[BX], 2008H
B. ADD DS:[BP+DI], 0800H 源, 目的操作数长度一致.
C. MOV DX, 2008H
D. MOV WORD PTR [SP], 08H
6. MOV AL, 85H
SUB AL, 0AFH 85 - AF.
上述指令执行后, 标志位 CF 和 OF 的值是 (A)。
CF 运算有进/借位, 置 1.
OF 有符号数运算溢出, 置 1.
ZF 运算结果为 0, 置 1.
SF 结果为负, 置 1.

A. CF=1,OF=0

B. CF=0,OF=0

C. CF=1,OF=1

D. CF=0,OF=1

00001000

7. 如果 AL 中仅有存放的数据的第 3 位 (最低为第 0 位) 为 1, 则转到标号 LAB1 去执行。不能实现上述转移的程序段是 ()

A. MOV CL,4

B. MOV CL,4

SHL AL,CL

逻辑左移

SHR AL,CL

逻辑右移

最后一个移出的位为 1, CF 置 1.

JC LAB1

JC LAB1

C=1 则跳转.

C. AND AL,08H

D. TEST AL,08H

按位与, 置位标志位, 只有一个临时结果.

JNZ LAB1

JNZ LAB1

Z=0 AL & 08H = 08H.

高位在低地址.

8. 对大端字节序的计算机而言, 设物理地址 (10FF0H)=10H, (10FF1H)=20H, (10FF2H)=30H, 如从地址 10FF1H 中取出一个字的内容是 ()

A. 2030H

B. 2010H

C. 1020H

D. 3020H

段寄存器 DS:

~~BP~~ BX, SI, DI

段寄存器 SS:

BP.

9. 已知 DS=1000H, SS=2000H, BP=2000H, DI=1234H, 则指令 MOV AX, [BP+DI+2] 的源操作数据在 () 中.

A. 物理地址 23234H

B. 物理地址 13234H

有 BP, 段寄存器一定是 SS.

C. 物理地址 23236H

D. 物理地址 13236H

(相对).

(可跨越).

(线性地址) 段 $\times 16 + 偏移$.

基址变址寻址.

一定是 BP/BX + SI/DI.

10. 下面指令语句中, 语法正确的是 ()

A. MOV BYTE PTR[BX], 5000H

B. JMP SHORT PTR OPR

C. CMP WORD PTR[BX], 60

D. INC [BX] \times PTR.

二、填空题 (10x3 = 30)

1. 对于 8086 而言, 假设某中断的中断号为 12H, 则其中断向量的 CS 位于物理地址 4AH。

IF=0, TF=0, IP=0000: [源 $\times 4$], CS=0000: [源 $\times 4$]

2. 如果当前 (DS)=2100H, (BP)=0158H, (DI)=10A5H, 则执行指令 LEA BX, [BP+DI] 后, (BX) = 221FDH。

$2100 + 0158 + 10A5$
 $\underline{2158}$
 $221FD$

3. MOV AX, SS:[BX+SI+3] 的寻址方式是 基址变址寻址

$5 + 30 \times 36 + 6 \times 36^2 + 15 \times 36^3$

4. 将 36 进制数 F6U5 转换为 10 进制是 708701

5. (+25) 的 8 位 16 进制补码表示是 19H, (-38) 的 8 位 16 进制补码表示是 DAH。

6. 外设和主机交换 (传送) 数据的四种基本方式是 直接交换、查询 (轮询)、中断、成组传送。

三、名词解释 (5x4 = 20)

1. 汇编语言

由助记符和指令中数字为组成的汇编码为基本元素、直接描述计算机硬件基本操作序列(指令流)的编程语言。

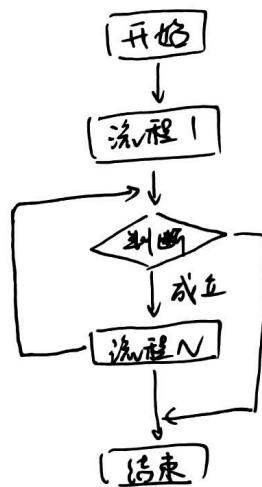
2. 8086 的指令分类

- ① 数据传送 ② 算术运算 ③ 位操作
④ 串操作 ⑤ 控制转移 ⑥ 处理器控制。

3. 微机系统 (PPT没找到)

以微处理器为核心, 结合存储器, 输入输出设备以及其他硬件和软件资源组成的计算机系统。

4. 画流程图说明“当...成立”型循环的循环结构



四、程序阅读 (10)

阅读下列程序, 回答问题:

(1) 该程序(很可能)实现什么功能? 冒泡排序, 从小到大排序。

(2) 已知该程序中有两个(类)错误, 它们分别是什么错误?

(3) 如果(2)中的错误都被修正, 程序执行完毕后, ARRAY 的值是多少?

	TITLE	WHATSTHIS
LEN	EQU	10

(2) ARRAY 是以字为单位,

① 但是 DI 在比较相邻两个元素时只加了 1。

② 内循环判断 DI 应该是小于则跳转。
JNZ 只判断是否相等。

(3) 14, 44, 111, 423, 451, 524,
741, 891, 987, 8831。

DATA	SEGMENT	
ARRAY	DW	524,14,44,111,8831,451,741,891,981,423
DATA	ENDS	

CODE	SEGMENT	
	ASSUME	CS:CODE, DS:DATA

START:	MOV	AX, DATA
	MOV	DS, AX
	XOR	SI, SI

OUTER:	XOR	DI, DI	DI = 0
--------	-----	--------	--------

INNER:	MOV	AX, ARRAY[DI]	
	MOV	BX, ARRAY[DI+1]	DI + 2
	CMP	AX, BX	
	JB	SKIP	
	MOV	ARRAY[DI], BX	
	MOV	ARRAY[DI+1], AX	

SKIP:			DI + 2
-------	--	--	--------

JB 无符号数
JL 有符号数

JB
或 JL

ADD	DI, 2
CMP	DI, LEN*2
JNZ	INNER
ADD	SI, 1
CMP	SI, LEN
JNZ	OUTER
MOV	AH, 4CH
INT	21H

JB 小于则跳转
> AX > BX 交换

检查范围

CODE	ENDS
------	------

END	START
-----	-------

五、程序编写 (20) - 一定要把平时的实验过一遍.

设有 113 个学生的汇编语言课程成绩存放在 SCORE 数组中。试编写程序统计 60~69 分, 70~79 分, 80~89 分, 90~100 分段 (均含) 的整数平均分并分别存放在 A6, A7, A8, A9 和 A10 存储单元中, 其中负责统计单个分数段的部分要求使用过程或宏实现, 并在主程序中针对各个区间分别调用该过程或宏。退出程序可使用 4CH 号调用。

参考程序开头:

5 个分段?

100 单独在一段.

```

                TITLE      AVGSCORE

STUNUM         EQU        113

DATA           SEGMENT
GRADE          DB         76,69,84,90,73,88,99,63,100,80, ....
A6             DB         0
A7             DB         0
A8             DB         0
A9             DB         0
A10            DB         0
...
DATA           ENDS
                SUM        DB         5    DUP(0)
                NUM        DB         5    DUP(0)
    
```

```

CODE           SEGMENT
                ASSUME     CS:CODE, DS:DATA.

MAIN           PROC        FAR.
                MOV        AX, DATA
                MOV        DS, AX            ; 初始化.
                MOV        CX, 0
                XOR         CX, CX          ; CX=0, 开始计数.
START:         CALL        SCORE_CLASS    ; 统计每个分段
                INC        CX              ; CX++.
                CMP        CX, 4
                JLE        START           ; CX ≤ 4, 继续循环. 总共 5 个分段.
                CALL        SCORE_AVG     ; 计算平均分.
                MOV        AH, 4CH        ; 结束程序.
                INT        21H
                RET
    
```

MAIN ENDP

SCORE_CLASS PROC NEAR ; CX为对应分段下标.

MOV AL, 10

XOR AH, AH ; AX=10.

MUL CX ; AX = 0, 10, 20, 30, 40.

ADD AX, 60

MOV BX, AX

ADD BX, 9 ; 当前分段 [AX, BX]

~~PUSH CX~~

MOV DL, 0

XOR DH, DH ; DX=0, 计数

LEA SI, ~~ARRAY~~ GRADE ; H grade[] 开始.

CHECK: ~~CMP~~ CMP [SI], AX

JL ~~THELOOP~~

CMP [SI], BX

JG THELOOP

LEA DI, SUM

~~ADD DI, CX~~

ADD DI, CX

PUSH AX

MOV AX, [SI]

ADD [DI], AX ; SUM[CX] += GRADE[DX].

POP AX

LEA DI, NUM

ADD DI, CX

INC BYTE PTR [DI] ; NUM[CX]++.

THELOOP: INC DX < INC SI

CMP DX, 113

JL CHECK

RET

SCORE_CLASS ENDP.

SCORE - AVG PROC NEAR

; 可以用个循环设一下.

MOV AL, SUM[0]

XOR AH, AH

MOV BL, NUM[0]

DIV BL

LEA SI, A6

MOV [SI], AL ; SUM[0]/NUM[0] → A6, 以下同理.

MOV AL, SUM[1]

XOR AH, AH

MOV BL, NUM[1]

DIV BL

LEA SI, A7

MOV [SI], AL

MOV AL, SUM[2]

XOR AH, AH

MOV BL, NUM[2]

DIV BL

LEA SI, A8

MOV [SI], AL

MOV AL, SUM[3]

XOR AH, AH

MOV BL, NUM[3]

DIV BL

LEA SI, A9

MOV [SI], AL

MOV AL, SUM[3]

XOR AH, AH

MOV BL, NUM[3]

DIV BL

LEA SI, A10

MOV [SI], AL

RET

SCORE - AVG ENDP

CODE ENDS

END MAIN.